



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN



DISSERTATION

Interactive Natural Language Processing for Discourse Analysis

Tim Fischer

Language Technology

Department of Informatics

Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg

Hamburg, Germany

A thesis submitted for the degree of
Doctor rerum naturalium (Dr. rer. nat.)

2026

Interactive Natural Language Processing for Discourse Analysis

Dissertation submitted by: Tim Fischer

Date of Submission: 10.03.2026

Date of Disputation: 27.04.2026

Supervisor:

Prof. Dr. Chris Biemann, Universität Hamburg

Committee:

1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg

2nd Examiner: Prof. Dr. Simone Paolo Ponzetto, Universität Mannheim

Chair: Prof. Dr. Mathias Fischer, Universität Hamburg

Vice Chair: Prof. Dr. Anne Lauscher, Universität Hamburg

Universität Hamburg, Hamburg, Germany

Faculty of Mathematics, Informatics and Natural Sciences

Department of Informatics

Language Technology

Affidavit

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

I hereby declare and affirm that this doctoral dissertation is my own work and that I have not used any aids and sources other than those indicated. If electronic resources based on generative artificial intelligence (gAI) were used in the course of writing this dissertation, I confirm that my own work was the main and value-adding contribution and that complete documentation of all resources used is available in accordance with good scientific practice. I am responsible for any erroneous or distorted content, incorrect references, violations of data protection and copyright law or plagiarism that may have been generated by the gAI.

Date

Signature

(Tim Fischer)

Use of Third-Party Software

This dissertation was completed with the assistance of several third-party tools, which are acknowledged here.

The thesis itself was written and managed on the Overleaf¹ platform, utilizing the University of Hamburg LT thesis template².

For editorial support, Grammarly³ was employed to check punctuation, grammar, spelling, and to refine the overall style and tone of the text. Furthermore, I utilized generative AI tools in a supplementary way. Gemini⁴ assisted in rephrasing paragraphs to enhance fluency, coherence, or conciseness. In addition, I used Google Translate⁵ to translate my initial German drafts of several sections, including the introduction, background on discourse analysis, and motivations, which were then manually proofread and corrected. Importantly, these generative AI tools were used exclusively to generate suggestions, which I subsequently corrected and incorporated myself. They were not used to generate any core content or substantive text.

Miro⁶ served as a platform for initial brainstorming, note-taking, and early draft development, as well as for creating several figures. Additionally, other figures were drawn using draw.io⁷ and PowerPoint⁸. The code for all data tables was generated using TablesGenerator⁹ from CSV exports of my experimental results. Finally, all screenshots featured throughout the dissertation were captured using the Snipping Tool integrated into Windows and subsequently edited in Paint.NET¹⁰ for clarity and emphasis.

1. <https://www.overleaf.com/>

2. <https://github.com/uhh-lt/thesis-template-uhh-lt-latex>

3. <https://www.grammarly.com/>

4. <https://gemini.google.com/app>

5. <https://translate.google.com/>

6. <https://miro.com/>

7. <https://app.diagrams.net/>

8. <https://www.microsoft.com/powerpoint>

9. <https://www.tablesgenerator.com/>

10. <https://www.getpaint.net/>

Abstract

Qualitative research methodologies, particularly the Sociology of Knowledge Approach to Discourse (SKAD), rooted in Grounded Theory and Hermeneutics, face limitations when confronted with the massive scale of digital data. This dissertation addresses this gap by developing and extending the Discourse Analysis Tool Suite (DATS), an open-source web application that integrates natural language processing (NLP) technologies with the iterative demands of discourse analysis. This work develops an interactive, NLP-based working environment to advance the digitalization of qualitative discourse analysis. It is structured around a multi-step iterative workflow (import, exploration, annotation, analysis, interpretation) derived from the SKAD methodology, with contributions designed to augment each phase. We first define a set of requirements, emphasizing scalability and adaptability, which serve as the guiding principles for DATS and its subsequent extensions. DATS is the foundation of this work by providing both a preprocessing pipeline that prepares multimodal data for analysis and basic functionalities such as search, manual annotation, count-based statistics, and memos for interpretation. The core of this thesis lies in developing several AI-augmented extensions to DATS that follow the human-in-the-loop paradigm. To facilitate exploration, we introduced *Perspectives*, an interactive clustering interface that leverages contextual embeddings. To align the visualization with researchers' emergent analytical perspective, we implement an iterative refinement process utilizing few-shot fine-tuning of the embedding model. This document clustering pipeline was evaluated on several datasets, confirming its effectiveness in steering analytical outcomes. Complementing this, the *LLM Assistant* was developed to streamline document organization, metadata extraction, and span annotation by employing zero-shot inference from large language models (LLMs). This approach was validated through a comprehensive benchmarking of open-weight LLMs across tasks in qualitative data analysis. Building on the *LLM Assistant*, we developed a three-stage *Annotation Assistant* for sequential sentence classification. This adaptive approach dynamically transitions between zero-shot prompting, few-shot prompting, and model fine-tuning to deliver optimal suggestions regardless of data scarcity. The strategy was evaluated on five datasets, confirming that performance scales with the number of examples. Furthermore, we integrated search and annotation into an analytical feature, the *Concept Over Time Analysis* (COTA). COTA uses an iterative feedback loop that leverages few-shot fine-tuning to rapidly adapt semantic search, enabling semi-automatic classification and visualization of concepts over time. This method was evaluated across three classification datasets to confirm its adaptability and concept-discrimination capabilities. Finally, we enhanced DATS' interpretation support by integrating *Whiteboards* for visual analysis. *Whiteboards* create an interactive working environment for interpretation by providing a 2D canvas for organizing and linking research artifacts (memos, codes, documents,

and annotations). The utility of this feature was validated through multiple qualitative use cases developed in collaboration with social scientists.

Zusammenfassung

Qualitative Forschungsmethoden, insbesondere die auf der Grounded Theory und Hermeneutik basierende Wissenssoziologische Diskursanalyse (WDA), stoßen an ihre Grenzen, wenn sie mit großen Datenmengen konfrontiert werden. Diese Dissertation adressiert dieses Problem durch die Entwicklung und Erweiterung der Discourse Analysis Tool Suite (DATS), einer Open-Source-Webanwendung, die Natural Language Processing (NLP) Technologien mit den iterativen Anforderungen der Diskursanalyse vereint. Diese Arbeit entwickelt eine interaktive, NLP-basierte Arbeitsumgebung, um die Digitalisierung der qualitativen Diskursanalyse voranzutreiben. Sie ist um einen mehrstufigen iterativen Workflow (Import, Exploration, Annotation, Analyse, Interpretation) strukturiert, der auf der WDA-Methodik basiert. Wir definieren zunächst zentrale Anforderungen, insbesondere Skalierbarkeit und Anpassungsfähigkeit, die als Leitprinzipien für DATS und dessen Erweiterungen dienen. DATS ist die Grundlage dieser Arbeit und stellt sowohl die Vorverarbeitungspipeline, die multimodale Daten für Analysen aufbereitet, als auch Basisfunktionalitäten wie Suche, Annotation, Statistiken und Memos bereit. Der Kern dieser Arbeit liegt in der Entwicklung mehrerer KI-gestützter Erweiterungen für DATS im Sinne des Human-in-the-Loop-Paradigmas. Zur Erleichterung der Exploration entwickelten wir *Perspectives*, eine interaktive Clustering Anwendung, die kontextualisierte Embeddings nutzt. Um die Visualisierung an die analytischen Perspektiven der Forschenden anzupassen, wurde ein iterativer Verfeinerungsprozess mittels few-shot fine-tuning entwickelt. Diese Clustering-Pipeline wurde an mehreren Datensätzen evaluiert und die Wirksamkeit bei der Steuerung analytischer Ergebnisse bestätigt. Ergänzend dazu wurde der *LLM Assistant* entwickelt, um die Dokumentenorganisation, die Metadatenextraktion und die Annotation durch den Einsatz von Large Language Models (LLMs) zu erleichtern. Dieser zero-shot Ansatz wurde durch umfassendes Benchmarking offener LLMs für Aufgaben der qualitativen Datenanalyse validiert. Darauf aufbauend entwickelten wir einen 3-stufigen *Annotation Assistant* zur Satzklassifikation. Dessen adaptiver Ansatz wechselt dynamisch zwischen zero-shot prompting, few-shot prompting und Model fine-tuning, um abhängig von der verfügbaren Datenmenge optimale Vorschläge zu liefern. Die Strategie wurde an fünf Datensätzen evaluiert, wodurch ihre Leistungsskalierbarkeit bestätigt wurde. Darüber hinaus integrierten wir Suche und Annotation in eine analytische Funktion: Die *Concept Over Time Analysis* nutzt iteratives Feedback und few-shot fine-tuning, um semantische Suchen schnell anzupassen und so eine semi-automatische Klassifikation und Visualisierung von Konzepten über Zeit zu ermöglichen. Diese Methode wurde anhand von drei Datensätzen evaluiert, um ihre Anpassungsfähigkeit zu bestätigen. Schließlich erweiterten wir DATS um *Whiteboards* zur visuellen Analyse. *Whiteboards* schaffen eine interaktive 2D-Umgebung für Interpretationen, in der Forschungsartefakte (Memos, Codes, Dokumente

und Annotationen) organisiert und verknüpft werden. Die Nützlichkeit wurde durch mehrere qualitative Anwendungsfälle validiert, die in Zusammenarbeit mit Sozialwissenschaftlern entwickelt wurden.

Contents

List of Publications	vii
List of Figures	xi
List of Tables	xiii
List of Abbreviations	xv
1 Introduction	1
1.1 Motivation	1
1.2 SKAD Overview	2
1.3 Problem Statement & the D-WISE project	4
1.4 An Abstract Methodology for Discourse Analysis	6
1.5 Requirements for a Discourse Analysis Tool Suite	7
1.6 Research Questions	9
1.7 Research Methodology	12
1.8 Contributions & Publications	13
1.8.1 Core Publications	13
1.8.2 Related Publications	15
1.9 Dissertation Overview	16
2 Background – NLP	19
2.1 Embeddings	20
2.1.1 Lexical Semantics	20
2.1.2 Vector Semantics	21
2.1.3 Sparse, Count-Based Embeddings	21
2.1.4 Dense Embeddings: word2vec	21
2.1.5 Cosine Similarity	22
2.1.6 Limitations	22
2.1.7 Visualizing Embeddings	22
2.1.8 Text Clustering	23
2.2 Language Models	24
2.3 Large Language Models	26
2.3.1 Prompting	26
2.3.2 Autoregressive Generation	26
2.3.3 Training Pipeline	27
2.3.4 Evaluation	29
2.3.5 Limitations	29
2.4 Transformer	30
2.4.1 Attention	31

2.4.2	The Transformer Block	32
2.4.3	Input Encoding	33
2.4.4	Language Modeling Head	34
2.4.5	Training	34
2.4.6	Computational Efficiency	35
2.5	Masked Language Models	36
2.5.1	Architecture	37
2.5.2	Pretraining	37
2.5.3	Contextual Embeddings	38
2.5.4	Fine-Tuning for Downstream Tasks	39
2.6	LLMs used in the Dissertation	41
2.6.1	Llama 3.1	41
2.6.2	Mistral NeMo	42
2.6.3	Gemma 2	42
2.6.4	Gemma 3	42
3	Background – Discourse Analysis	45
3.1	Empirical Social Sciences	46
3.1.1	Definition	46
3.1.2	Research Approaches	47
3.1.3	Research Process	48
3.1.4	Quality Criteria	49
3.2	Grounded Theory	50
3.2.1	Definition	50
3.2.2	Methods	51
3.3	Hermeneutics	54
3.3.1	Definition	54
3.3.2	The Hermeneutic Attitude	55
3.3.3	The Interpretation Process	56
3.4	Discourse Analysis	57
3.4.1	The Discourse	57
3.4.2	Discourse Analysis	59
3.4.3	The Meaning of Power	61
3.4.4	Variants of Discourse Analysis	62
3.5	Procedure of a Discourse Analysis	62
3.5.1	Phase 1: Preliminary Work	62
3.5.2	Phase 2: Corpus Formation	62
3.5.3	Phase 3: Reconstructing the Discursive Formation	63
3.5.4	Phase 4: Fine-grained Analysis of Function & Power	64
3.5.5	Phase 5: Social Analysis	65
3.6	Sociology of Knowledge Approach to Discourse	65
3.6.1	Hermeneutics & Reflexivity	65
3.6.2	Grounded Theory as an Analytical Tool	66
3.6.3	Analytical Focus	66

4	Discourse Analysis Tool Suite	69
4.1	Motivation	70
4.2	Related Tools	71
4.3	Feature Overview	80
4.4	Terminology	82
4.5	Technical Architecture	82
4.5.1	Backend Implementation	83
4.5.2	Client Server Communication	85
4.5.3	Frontend Implementation	86
4.5.4	Deployment & Orchestration	86
4.6	Import & Preprocessing	87
4.6.1	Motivation	87
4.6.2	Challenges	88
4.6.3	Data Import & Preprocessing in DATS	88
4.7	Introducing Alice	91
4.8	Conclusion	92
5	Exploration, Search & Organization	95
5.1	Motivation	96
5.1.1	Exploration	96
5.1.2	Search & Filtering	97
5.1.3	Data Organization	98
5.2	Challenges	99
5.3	Exploration, Search & Organization in DATS	101
5.3.1	Workflow with Alice	101
5.3.2	Addressed Challenges	102
5.3.3	Excursus: Configurable Similarity Search	103
5.4	Perspectives: Interactive Document Tagging in DATS	105
5.4.1	Introduction	105
5.4.2	Related Work	106
5.4.3	Workflow with Alice	108
5.4.4	Interactive Clustering	110
5.4.5	Evaluation	113
5.4.6	Integrating Perspectives in DATS	119
5.4.7	Conclusion	120
5.5	Exploring Large Language Models for Document Organization	121
5.5.1	Introduction	121
5.5.2	Related Work	122
5.5.3	Workflow with Alice	123
5.5.4	Benchmark Setup	125
5.5.5	Document Classification Evaluation	125
5.5.6	Metadata Extraction Evaluation	128
5.5.7	Integrating the LLM Assistant in DATS	131
5.5.8	Conclusion	132
5.6	Conclusion & Discussion	132

6	Annotation	137
6.1	Motivation	138
6.2	Challenges	139
6.3	Coding vs. Annotation	140
6.4	Computational Grounded Theory	141
6.5	Annotation in DATS	144
6.5.1	Workflow with Alice	144
6.5.2	Addressed Challenges	145
6.5.3	Excursus: Annotation Scaling	145
6.6	Exploring Large Language Models for Annotation	147
6.6.1	Introduction	147
6.6.2	Workflow with Alice	148
6.6.3	Sequence Labeling Benchmark	149
6.6.4	Integrating Annotation in DATS' LLM Assistant	153
6.6.5	Conclusion	153
6.7	Semi-automatic Sequential Sentence Classification	154
6.7.1	Introduction	154
6.7.2	Workflow with Alice	156
6.7.3	Three-phase Annotation Assistance	157
6.7.4	Benchmarking Sequential Sentence Classification	159
6.7.5	Transitioning Phase 2 $\rightarrow$ 3	163
6.7.6	Integrating Sequential Sentence Annotation in DATS' LLM Assistant	164
6.7.7	Conclusion	164
6.8	Excursus: AnnoPlot - Visual Annotation Organization	165
6.9	Conclusion & Discussion	167
7	Analysis	171
7.1	Motivation	172
7.2	Challenges & Risks	173
7.3	Analysis in DATS	174
7.3.1	Workflow with Alice	175
7.3.2	Addressed Challenges	177
7.4	Concept Over Time Analysis	177
7.4.1	Introduction	177
7.4.2	Related Work	179
7.4.3	Workflow with Alice	180
7.4.4	Experiments	182
7.4.5	Implementation	185
7.5	Conclusion & Discussion	187
8	Reflection & Interpretation	191
8.1	Motivation	192
8.1.1	Interpretation	192
8.1.2	Reflection	193
8.2	Challenges	194
8.3	Interpretation & Reflection in DATS	195
8.3.1	Workflow with Alice	195

8.3.2	Addressed Challenges	197
8.4	Whiteboards for Visual Qualitative Analysis	197
8.4.1	Introduction	197
8.4.2	Related Work	198
8.4.3	Implementation	199
8.4.4	Exemplary Use Cases	201
8.4.5	Workflow with Alice	205
8.5	Conclusion & Discussion	206
9	Conclusion	209
9.1	Summary	209
9.2	Limitations	213
9.3	Future Work	215
Appendices		
A	Additional Material	221
A.1	Chapter 4 - Exploration, Search & Organization	221
A.1.1	Perspectives: Interactive Document Tagging in DATS . . .	221
A.2	Chapter 5 - Annotation	222
A.2.1	Semi-automatic Sequential Sentence Classification	222
	References	227

List of Publications

Dissertation Core Publications

Tim Fischer and Chris Biemann. 2024. Exploring Large Language Models for Qualitative Data Analysis. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, edited by Mika Härmäläinen, Emily Öhman, So Miyagawa, Khalid Alnajjar, and Yuri Bizzoni, 423–437. Miami, FL, USA: Association for Computational Linguistics.

Contributions: I conceptualized and implemented the LLM Assistant extension for DATS, including modifications to the system architecture and the integration of large language models. I also managed the research design, including the composition of the multilingual benchmark and the subsequent comparative performance analysis of open-weight models. Chris Biemann provided recommendations that improved the final manuscript.

———. 2025. Semi-automatic Sequential Sentence Classification in the Discourse Analysis Tool Suite. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, edited by Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao, 151–162. Albuquerque, NM, USA: Association for Computational Linguistics.

Contributions: I evolved the LLM Assistant into a three-staged Annotation Assistant by conceptualizing the adaptive strategy and implementing the software components in the DATS backend and frontend. I also integrated the automated model training pipeline and designed the experimental framework to evaluate performance across five datasets. Chris Biemann provided suggestions and recommendations for the final manuscript.

———. 2026. Perspectives - Interactive Document Clustering in the Discourse Analysis Tool Suite. Note: under review. arXiv: 2602.15540 [cs.CL].

Contributions: I conceptualized and implemented the *Perspectives* extension, managing the development of database extensions, backend services, and frontend interfaces. I also integrated the underlying language models and independently designed and executed the empirical evaluations. Chris Biemann provided conceptual guidance and editorial suggestions.

Tim Fischer, Florian Schneider, Robert Geislinger, Florian Helfer, Gertraud Koch, and Chris Biemann. 2024. Concept Over Time Analysis: Unveiling Temporal Patterns for Qualitative Data Analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, edited by Kai-Wei Chang, Annie Lee, and Nazneen Rajani, 148–157. Mexico City, Mexico: Association for Computational Linguistics.

Contributions: Florian Schneider and I developed the initial concept for this temporal analysis tool. While Robert Geislinger implemented the first prototype, I expanded and integrated it into DATS and developed the entire user interface and APIs. Florian Schneider and I conducted the experiments jointly. Florian Helfer and Gertraud Koch identified research interests and initial use cases for analyzing concept evolution, thereby initiating this project. Chris Biemann contributed through editorial recommendations.

Tim Fischer, Florian Schneider, Fynn Petersen-Frey, Anja Silvia Mollah Haque, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2024. Extending the Discourse Analysis Tool Suite with Whiteboards for Visual Qualitative Analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, edited by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, 7017–7022. Torino, Italia: ELRA and ICCL.

Contributions: I designed and implemented the Whiteboards extension as a full-stack addition to the DATS platform. Anja Silvia Mollah Haque, Isabel Eiser, and Gertraud Koch initiated the project by testing Miro and identifying the need for an integrated visualization tool for discourse analysis that ensures interpretations are traceable back to the source materials. Following the technical implementation, Anja Silvia Mollah Haque provided the exemplary use cases as part of her analysis of the German e-health debate. Chris Biemann, Florian Schneider, and Fynn Petersen-Frey provided suggestions for improving the manuscript.

Florian Schneider, Tim Fischer, Fynn Petersen-Frey, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2023. The D-WISE Tool Suite: Multi-Modal Machine-Learning-Powered Tools Supporting and Enhancing Digital Discourse Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, edited by Danushka Bollegala, Ruihong Huang, and Alan Ritter, 328–335. Toronto, Canada: Association for Computational Linguistics.

Contributions: Florian Schneider and I jointly conceptualized the system architecture, database structure, and preprocessing pipelines. I held primary technical responsibility for the end-to-end implementation of the user interfaces and frontend logic. I worked in an iterative feedback loop with Isabel Eiser and Gertraud Koch, who provided the foundational use cases and domain insights necessary to align the platform with the requirements of digital discourse analysis. While Florian Schneider implemented the REST API and backend services, Fynn Petersen-Frey contributed towards the flexible span annotation components. Chris Biemann contributed to the initial conceptual phase and provided recommendations for improving the paper.

Dissertation Related Publications

Isabel Eiser, Tim Fischer, Florian Schneider, Gertraud Koch, Chris Biemann, and Fynn Petersen Frey. 2023. Open Science Prinzipien und interdisziplinäre Kollaboration in D-WISE: Zwischen Hermeneutik und Digitaler Methode in der Diskursanalyse. In *9. Tagung des Verbands Digital Humanities im deutschsprachigen Raum*, 252–257. Trier, Luxemburg: Zenodo.

Contributions: Isabel Eiser and I conceptualized this work on interdisciplinary collaboration. Isabel Eiser developed the idea of a co-creation approach to software development and authored the majority of this publication. I authored sections on related work and the technical section on the D-WISE Tool Suite, which were reviewed by Florian Schneider and Fynn Petersen-Frey. Gertraud Koch and Chris Biemann provided editorial improvements.

Elisabeth Fittschen, Tim Fischer, Daniel Brühl, Julia Spahr, Yuliia Lysa, and Phuoc Thang Le. 2024. AnnoPlot: Interactive Visualizations of Text Annotations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Nikolaos Aletras and Orphee De Clercq, 106–114. St. Julians, Malta: Association for Computational Linguistics.

Contributions: I conceptualized a system for visualizing annotations and code hierarchies, and for automatically detecting errors. I then supervised the student team (Elisabeth Fittschen, Daniel Brühl, Julia Spahr, Zuliia Lysa, and Phuoc Thang Le) that conducted the implementation. I provided technical support during development and performed the editorial refinements of the paper. Elisabeth Fittschen was the lead developer of the dynamic UMAP model and conducted the associated error-detection experiments.

Hans Ole Hatzel, Fynn Petersen-Frey, Tim Fischer, and Chris Biemann. 2023. Dimensions of Similarity: Towards Interpretable Dimension-Based Text Similarity. In *Frontiers in Artificial Intelligence and Applications*, edited by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Rădulescu, 1004–1011. Kraków, Poland: IOS Press.

Contributions: Hans Ole Hatzel, Fynn Petersen Frey, and I jointly conceptualized the interpretable similarity search system. I contributed to fine-tuning experiments on the multi-task learning approach and designed UI mockups for the user interaction model. While Hans Ole Hatzel led the experimental effort (conducting preliminary experiments and experiments on entity focus and German poetry), Fynn Petersen Frey conducted comparisons between human and machine judgments and all downstream experiments. I managed the final consolidation of results and authored the manuscript. Chris Biemann provided editorial suggestions.

Gertraud Koch, Chris Biemann, Isabel Eiser, Tim Fischer, Florian Schneider, Teresa Stumpf, and Alejandra Tijerina García. 2022. D-WISE Tool Suite for the Sociology of Knowledge Approach to Discourse. In *Culture and Computing*, edited by Matthias Rauterberg, 68–83. Cham, Germany: Springer International Publishing.

Contributions: This paper outlines the development of the D-WISE Tool Suite, which all authors conceptualized. Gertraud Koch invented the idea of AI-in-the-loop, as opposed to human-in-the-loop. Isabel Eiser, Teresa Stumpf, and Tijerina García authored the chapters on project overview and human-computer interaction. Together with Florian Schneider, I authored the technical chapter of this work. Chris Biemann provided editorial suggestions.

Fynn Petersen-Frey, Tim Fischer, Florian Schneider, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2023. From Qualitative to Quantitative Research: Semi-Automatic Annotation Scaling in the Digital Humanities. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, edited by Munir Georges, Aaricia Herygers, Annemarie Friedrich, and Benjamin Roth, 52–62. Ingolstadt, Germany: Association for Computational Linguistics.

Contributions: Fynn Petersen Frey, Florian Schneider, and I jointly conceptualized this system. I defined the operational workflow and designed the user interface mockups for both annotation and analysis. Fynn Petersen Frey led the overall project, outlined the technical methodology for scaling, and conducted the corresponding annotation scaling experiments. I collaborated with Florian Schneider on the design and execution of the experiments involving few-shot transformer adapter classifiers. Chris Biemann provided editorial suggestions.

Other Publications

Selenia Anastasi, Tim Fischer, Florian Schneider, and Chris Biemann. 2025. IDA – Incel Data Archive: A multimodal comparable corpus for exploring extremist dynamics in online interaction. In *Methods, analyses, and corpus construction*, edited by Louis Cotgrove, Laura Herzberg, and Harald Lungen, 275–304. De Gruyter.

Selenia Anastasi, Florian Schneider, Chris Biemann, and Tim Fischer. 2024. VIDA: The Visual Incel Data Archive. A Theory-oriented Annotated Dataset To Enhance Hate Detection Through Visual Culture. In *Proceedings of the 8th Workshop on Online Abuse and Harms*, edited by Yi-Ling Chung, Zeerak Talat, Debora Nozza, Flor Miriam Plaza-del-Arco, Paul Röttger, Aida Mostafazadeh Davani, and Agostina Calabrese, 59–67. Mexico City, Mexico: Association for Computational Linguistics.

Thennal D. K., Tim Fischer, and Chris Biemann. 2025. Large Language Models Are Overparameterized Text Encoders. In *Proceedings of the 10th Workshop on Representation Learning for NLP*, edited by Vaibhav Adlakha, Alexandra Chronopoulou, Xiang Lorraine Li, Bodhisattwa Prasad Majumder, Freda Shi, and Giorgos Vernikos, 170–184. Albuquerque, NM, USA: Association for Computational Linguistics.

Naquee Rizwan, Seid Muhie Yimam, Daryna Dementieva, Dr. Florian Skupin, Tim Fischer, Daniil Moskovskiy, Aarushi Ajay Borkar, Robert Geislinger, Punyajoy Saha, Sarthak Roy, Martin Semmann, Alexander Panchenko, Chris Biemann, and Animesh Mukherjee. 2025. HatePRISM: Policies, Platforms, and Research Integration. Advancing NLP for Hate Speech Proactive Mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, edited by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, 16008–16022. Vienna, Austria: Association for Computational Linguistics.

List of Figures

1.1	Research process of the Sociology of Knowledge Approach to Discourse	2
1.2	D-WISE interpretation of a Grounded Theory-based, hermeneutic research process	6
2.1	Language model architectures: decoders, encoders, and encoder-decoders	25
2.2	The (unidirectional/causal) transformer architecture	30
2.3	The architecture of a transformer block	32
2.4	LoRA adapter	35
3.1	Main focus of discourse analysis and discourse theory	57
4.1	Architecture of the Discourse Analysis Tool Suite	83
4.2	DATS multimodal preprocessing pipeline	89
4.3	DATS import user interface	92
4.4	DATS health check user interface	93
5.1	DATS search & filtering interface	101
5.2	DATS multimodal semantic search interface	102
5.3	Configurable similarity search mockup	104
5.4	Document similarity analysis mockup	104
5.5	<i>Perspectives</i> dashboard	108
5.6	<i>Perspectives</i> cluster detail dialog	109
5.7	<i>Perspectives</i> interactive 2D document map	110
5.8	<i>Perspectives</i> interactive clustering pipeline	111
5.9	<i>Perspectives</i> experiments – distribution of stars and product categories in the Amazon test set	113
5.10	<i>Perspectives</i> experiments – distribution of emotions and genres in the Spotify test set	114
5.11	Detailed <i>Perspectives</i> evaluation with increasing number of labeled examples per class	118
5.12	<i>LLM Assistant</i> 5-step user interface for document tagging	124
6.1	Three-step Computational Grounded Theory framework: From dataframe to conclusion	143
6.2	DATS annotation interface	144
6.3	<i>Annotation Scaling</i> mockup	146
6.4	<i>LLM Assistant</i> 5-step user interface for span annotation	149

6.5	Three-stage <i>Annotation Assistant</i> 6-step user interface for sentence annotation	157
6.6	DATS comparative sentence annotator	158
6.7	Evaluation of NV-embedd-based sequence tagger with increasing number of training examples	163
6.8	<i>AnnoPlot</i> Plot View	165
6.9	<i>AnnoPlot</i> Category View	165
6.10	<i>AnnoPlot</i> Error Identification	166
7.1	DATS word frequency analysis interface	175
7.2	DATS code frequency analysis interface	176
7.3	DATS search interface with contextual statistics	176
7.4	<i>Concept Over Time Analysis</i> annotation interface	181
7.5	<i>Concept Over Time Analysis</i> timeline interface	182
7.6	COTA evaluation – accuracies deviations over 10 fine-tuning runs	184
7.7	COTA evaluation – average runtimes over 10 fine-tuning runs . .	185
7.8	COTA evaluation–evolution of sentence embeddings	190
8.1	DATS logbook interface	196
8.2	<i>Whiteboards</i> use case – sampling map	199
8.3	<i>Whiteboards</i> use case – taxonomy management	202
8.4	<i>Whiteboards</i> use case – actor network	203
8.5	<i>Whiteboards</i> use case – image map	204

List of Tables

1.1	Contributions of the core papers to this dissertation’s research questions, organized per chapter	16
5.1	<i>Perspectives</i> evaluation – excerpt of prompts to rewrite original documents	116
5.2	<i>Perspectives</i> evaluation – excerpt of document embedding instructions	116
5.3	<i>Perspectives</i> evaluation – document rewriting, instruction-finetuned embeddings, and few-shot fine-tuning	117
5.4	LLM evaluation – prompts used for the document classification task	126
5.5	LLM evaluation – taxonomies of the datasets used in the document classification task	126
5.6	LLM evaluation – results of the document classification task . . .	127
5.7	LLM evaluation – prompts for the information extraction task . .	128
5.8	LLM evaluation – taxonomy of the MUC-4 dataset used in the template filling information extraction task	129
5.9	LLM evaluation – results of the information extraction task	130
6.1	LLM evaluation – prompts used for the sequence labeling task . .	150
6.2	LLM evaluation – taxonomies of the datasets used in the sequence labeling task	151
6.3	LLM evaluation – results of the sequence labeling task	152
6.4	Three-stage <i>Annotation Assistant</i> evaluation – prompt template for sequential sentence classification	160
6.5	Three-stage <i>Annotation Assistant</i> evaluation – results of prominent LLMs in zero-shot, few-shot, and fully fine-tuned settings on five sequential sentence classification datasets	162
7.1	<i>COTA</i> evaluation – number of tokens (white-space-separated words) per dataset	184
A.1	<i>Perspectives</i> evaluation – document embedding instructions	221
A.2	<i>Perspectives</i> evaluation – document rewriting prompts	222
A.3	Three-stage <i>Annotation Assistant</i> evaluation – category system of the CSAbstract dataset	223
A.4	Three-stage <i>Annotation Assistant</i> evaluation – category system of the Pubmed200k dataset	223
A.5	Three-stage <i>Annotation Assistant</i> evaluation – category system of EmotionLines and DailyDialog dataset	224

A.6	Three-stage <i>Annotation Assistant</i> evaluation – category system of the Coarse Discourse dataset	225
-----	--	-----

List of Abbreviations

AI	Artificial Intelligence
CAQDAS	Computer-Assisted Qualitative Data Analysis Software
COTA	Concept Over Time Analysis
CRF	Conditional Random Fields
DA	Discourse Analysis
DATS	Discourse Analysis Tool Suite
DH	Digital Humanities
DTO	Data Transfer Object
D-WISE	Digitale wissenssoziologische Diskursanalyse
ESR	Empirical Social Research
FP16	floating-point number format that uses 16 bits of memory
GPU	Graphics Processing Unit
GT	Grounded Theory
HITL	Human In the Loop
IE	Information Extraction
ICL	In-Context Learning
KNN	K-Nearest Neighbors
LLM	Large Language Model
LM	Language Model
LoRA	Low-Rank Adaptation
LSTM	Long short-term memory
MLM	Masked Language Modeling
MMLU	Massive Multitask Language Understanding
MTEB	Massive Text Embedding Benchmark
NER	Named Entity Recognition
NLP	Natural Language Processing
PCA	Principal Component Analysis
PEFT	Parameter-Efficient Fine-Tuning
QA	Question Answering
QDA	Qualitative Data Analysis

RAG		Retrieval-Augmented Generation
REST		Representational State Transfer
RQ		Research Question
SetFit		Sentence Transformer Fine-Tuning
SGD		Stochastic Gradient Descent
SKAD		Sociology of Knowledge Approach to Discourse
t-SNE		t-distributed Stochastic Neighbor Embedding
UMAP		Uniform Manifold Approximation and Projection

1

Introduction

Contents

1.1	Motivation	1
1.2	SKAD Overview	2
1.3	Problem Statement & the D-WISE project	4
1.4	An Abstract Methodology for Discourse Analysis	6
1.5	Requirements for a Discourse Analysis Tool Suite	7
1.6	Research Questions	9
1.7	Research Methodology	12
1.8	Contributions & Publications	13
1.8.1	Core Publications	13
1.8.2	Related Publications	15
1.9	Dissertation Overview	16

1.1 Motivation

The pervasive digitalization of public life has led to an unprecedented growth in the amount of data available online. This massive volume of data, spanning social media, forums, news, archived documents, and more, constitutes the landscape of contemporary societal discourse and opinion formation, making computer-assisted methods increasingly relevant for empirical analysis (Trübner and Mühlichen 2022).

Digital humanities investigates these discourses to understand their societal impact by answering key questions: *How do discourses subconsciously steer our cognition and actions? Which knowledge and opinions have become normalized and are now accepted as self-evident?* The analysis of underlying narratives, argumentation patterns, and power structures is therefore essential for deciphering complex social and political dynamics (Keller 2011b, pp. 43, 57–58; Traue et al. 2022, pp. 638–640).

Qualitative discourse analysis (DA) provides the multidisciplinary research framework for this task. It is deeply rooted in sociology, linguistics, philosophy, communication sciences, computer science, and statistics. Adopted widely since the 1970s, discourse analysis typically examines phenomena across extended temporal spans or contrasts distinct knowledge systems to establish the genesis and specific structure of social realities (Traue et al. 2022, p. 628).

The methodology of discourse analysis is broadly applicable. It provides a versatile framework for analyzing the dynamic interplay between language and social life and can be employed to investigate diverse communicative contexts, such as analyzing political speeches to understand mobilization strategies (Wodak et al. 1990), examining media texts to uncover underlying biases and structural exclusions (Jäger 1992), or exploring everyday interactions to reveal the construction of social norms (Truschkat 2008). Discourse analysis has become an indispensable analytical tool for researchers. Its methodologies have been adopted and adapted in a wide range of fields, including political communication, media studies, sociology, linguistics, anthropology, and education (Traue et al. 2022, p. 628).

This extensive application scope highlights the critical need for advanced tools to manage and process the scale of digital discourse data.

1.2 SKAD Overview

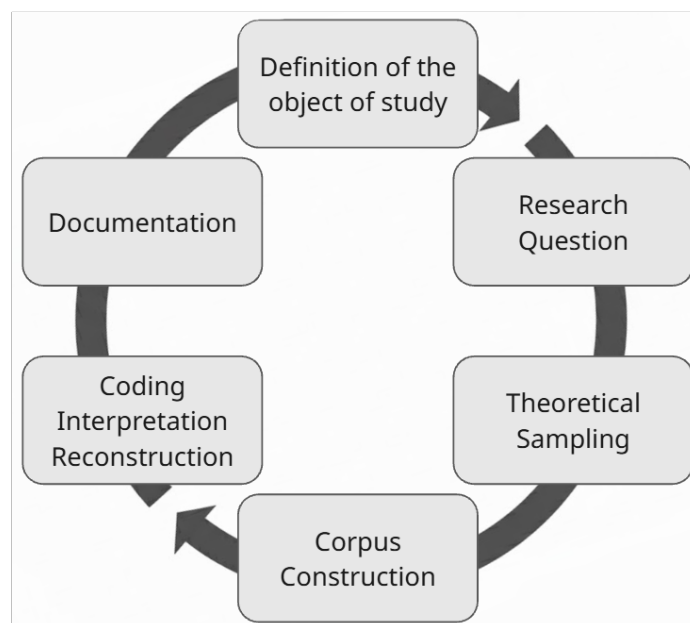


Figure 1.1: Research process of the Sociology of Knowledge Approach to Discourse (Keller 2011a, 2011b)

Discourse analysis encompasses a diverse range of theoretical and methodological perspectives, among which the Sociology of Knowledge Approach to Discourse (SKAD), developed by Keller (2011b), is prominent in sociological research in the German-speaking area. SKAD is central to the D-WISE project (Section 1.3) and, consequently, to this dissertation. Its particular contribution lies in the systematic integration and adaptation of approaches from Hermeneutics (Section 3.3) and

Grounded Theory (Section 3.2) for concrete empirical research practice (Keller 2011b, pp. 61–63). The SKAD framework (Section 3.6) is decisive for the conception of the Discourse Analysis Tool Suite (Chapter 4) and thus also for the technical further developments elaborated in this dissertation (Chapters 5 – 8).

The research program of SKAD is based on the principles of Grounded Theory and conceived as an iterative research process (see Figure 1.1). Instead of following a rigid, linear sequence, discourse analysis is understood as an iterative succession of interpretive, methodologically guided work steps that largely run in parallel. The process of a discourse analysis is explained in Section 3.5. Following Keller (2011a), we give a short overview of this workflow:

1. **Definition of the Object of Study and Research Question:** Discourse analysis begins with the definition of the field of knowledge or discourse to be examined. This can be broadly defined by topic, institutional setting, or actors. Exploration of the field occurs concurrently (e.g., through a literature review) and the provisional formulation of the research questions, which may be modified throughout the research project.
2. **Corpus Construction (theoretical sampling):** Instead of a complete corpus, a relevant section of the discourse is obtained from the potentially unmanageable amounts of data using theoretical sampling (Section 3.2.2). This methodologically guided selection serves to compile relevant research data, whereby generally only so-called "*natural data*" (documents found in the field, not created by researchers) are included in the corpus.
3. **Coding:** Across the various points in time of the investigation, the elements of the discourse are elaborated. This is done through the qualitative coding processes by Strauss et al. (1996) (open, axial, and selective coding, Section 3.2.2), in which objects, concepts, and strategies of the discourse are identified and organized into categories.
4. **Interpretation and Reconstruction:** Building on the coded content, the interpretative-analytical reconstruction takes place. This includes working out the discursive formation, the interpretation and narrative patterns, as well as the analysis of power, functional, and social relationships. The purely linguistic analysis is transcended by demonstrating connections between the linguistic forms and non-linguistic objectivations (such as social structures, institutions, and legal relationships).
5. **Documentation and Reflection:** Each research cycle concludes with the documentation and reflection of the achieved state of knowledge. This serves as the starting point for the following iterative cycle.

The iterative nature of qualitative data analysis (in particular SKAD and Grounded Theory) states that the material in the next cycle is circularly expanded to differentiate, consolidate, and verify the findings from the current iteration. Here, the constant comparative method is used (Section 3.2.2): Categories are consolidated through minimal comparison (by referencing similar data) and subsequently confronted with maximally different material (maximal comparison)

to account for variations and changes. This process continues until theoretical saturation is reached, at which point no essential new findings are added. Through this principle, the completeness of the text basis is measured by the representation of the statements represented in the discourse, not by the total quantity of all texts.

Studies using the SKAD framework focus, e.g., on environmental politics (Keller 1998), the symbolic production of space and cityscapes (Christmann 2004), health care policy (Bechmann 2007), the acknowledgment of competency in employment strategies (Truschkat 2008), public discourse on Satanism (Schmied-Knittel 2008), identity building in left wing social movements in Germany and Great Britain (Ullrich 2008) and Chinese migrant communities in Romania (Wundrak 2010), criminology (Singelstein 2009), same-sex marriage TV controversies in the US (Zimmermann 2010) or political sciences' mapping of suicide terrorism (Brunner 2011).

1.3 Problem Statement & the D-WISE project

The Challenges of Discourse Analysis

Discourse Analysis reaches its limits when confronted with the exponentially growing volumes of data available today. The coding and interpretation by human researchers, which constitute the essential steps of discourse analysis, have always been time-consuming and resource-intensive. However, this becomes particularly problematic given the enormous amounts of thematically relevant data available today. Comparative analyses across large data corpora are becoming increasingly complex.

The Interdisciplinary D-WISE Project

This is where natural language processing (NLP) methods (Chapter 2) come into play, offering promising solutions for the challenges of discourse analysis. Although NLP has made immense progress in recent years, applying these methods to the specific requirements of qualitative social science research remains a complex challenge.

This dissertation was conducted under the umbrella of the D-WISE¹ project (Digitale Wissenssoziologische Diskursanalyse), which aims to build the necessary methodological and technical bridges between computer science and the digital humanities. The D-WISE project, active from 2019 to 2023, represented an interdisciplinary partnership between Language Technology² and Anthropological Studies in Culture and History³. Its overarching goal was to develop digital methods for multimodal meaning analyses in research processes guided by Grounded Theory, specifically the Sociology of Knowledge Approach to Discourse. Concretely, two main goals were defined:

1. <https://www.dwise.uni-hamburg.de/>

2. <https://www.inf.uni-hamburg.de/en/inst/ab/lt>

3. <https://www.kulturwissenschaften.uni-hamburg.de/ekw.html>

1. The theoretical, methodological, and technical further development of the digital humanities, especially of the Sociology of Knowledge Approach to Discourse.
2. The development of a web-based, prototypical, open-source work environment as digital support for SKAD, intended to make relevant procedures available in a uniform, collaborative web environment and integrate the developed technical advancements.

The D-WISE project focused on the following two challenges that were to be addressed methodologically and technically:

- **Multimodal Data:** The materials to be analyzed in a discourse analysis potentially cover the entire spectrum of linguistic, visual, and auditory data formats. This multimodality is a challenge due to the lack of suitable procedures in the digital humanities.
- **Large Data Volumes:** The efficiency and quality of hermeneutic analyses should be improved by processing larger text quantities.

Contextualized embeddings (Section 2.1) were identified as a promising NLP solution for overcoming these challenges, especially when working with large corpora. Embeddings are vector representations for the meaning of words, sentences, or documents, arranged such that semantically similar terms are placed close to each other in vector space, depending on their context. Their use was planned particularly for search and annotation functions:

- **Search:** The use of embeddings could enable more powerful retrieval systems. Instead of directly linking meaning to keywords, such systems could support methods of discourse analysis by identifying similar statements or evidence for interesting text passages (analogously for image and audio content).
- **Annotation:** Following the human-in-the-loop paradigm, such semantic searches could be closely coupled with functions for manual annotation. This would allow similar statements or evidence to be semi-automatically found, compared, and coded.

Starting Point & Scope

A key outcome of the project is the development of the D-WISE Tool Suite (Schneider et al. 2023), a digital, collaborative, and open-source platform for discourse analysis that addresses the project challenges. During the course of the project, the name was changed to Discourse Analysis Tool Suite (DATS)⁴ (Chapter 4) is part of and a conceptual starting point for the further developments described in this dissertation.

Still, several challenges and opportunities were not explored and are consequently addressed in this work. While multimodality played an important role in the original D-WISE project's research questions, this dissertation focuses exclusively on text data.

4. <https://github.com/uhh-lt/dats>

1.4 An Abstract Methodology for Discourse Analysis

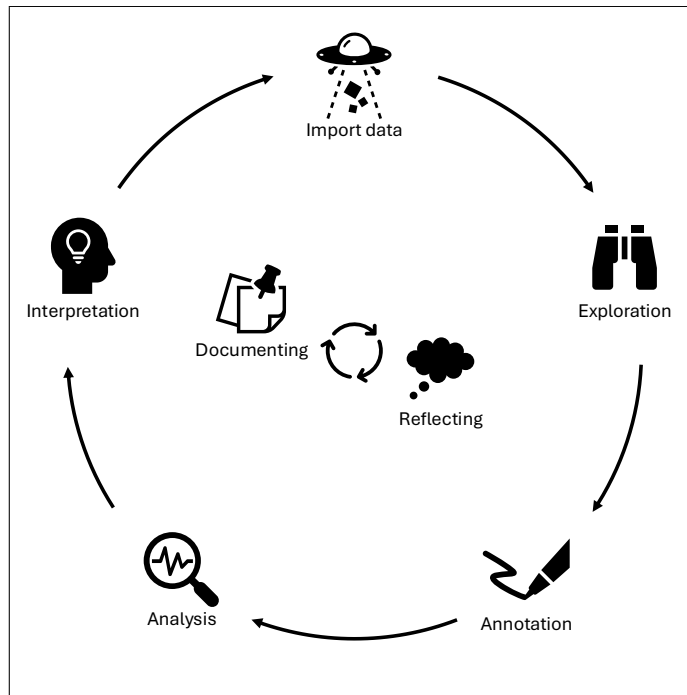


Figure 1.2: D-WISE interpretation of a Grounded Theory-based, hermeneutic research process. It was developed in the D-WISE project to be more abstract and more generally applicable across various research projects.

The Sociology of Knowledge Approach to Discourse offers a valuable framework for qualitative research projects. Keller (2011b, p. 48) understands SKAD as a research program for analyzing societal knowledge relationships and knowledge policies, providing a general framework for knowledge-analytical research beyond the narrower disciplinary boundaries of sociology.

Building on this SKAD framework, a slightly more general interpretation of a Grounded Theory-based, hermeneutic work process for discourse analysis was abstracted in the D-WISE project. This approach is oriented toward the methodology of SKAD, but distances itself slightly to allow connection to further discourse-analytical research perspectives and methods of empirical social research. This developed five-step workflow consequently forms the basis for the design and further development of the Discourse Analysis Tool Suite and is reflected in the structure of this thesis.

Viewed from a practical computer science perspective, discourse analysis can be modeled as an iterative process of knowledge discovery and exploration, in which researchers interact with data (through annotation), apply analytical techniques, and generate insights to answer their research questions (see Figure 1.2). The following iterative phases describe this process:

1. **Data Import:** Researchers first import and manage their data. This step involves organizing and preparing the data for analysis, supporting potentially different formats such as text, images, audio, and video. Importantly, in contrast to typical qualitative research endeavors, we assume the import of large-scale datasets (e.g., news archives).

2. **Data Exploration:** Researchers explore and search their large-scale corpora. This step often involves using computer-aided tools and algorithms (distant reading, Moretti (2016)) to both identify data relevant to their research question and reduce it to a manageable volume for qualitative analysis.
3. **Annotation:** Researchers code their data by categorizing relevant information, which requires close reading and interpretation of the data. Researchers apply their expertise here to identify and classify relevant discourse phenomena. The differences between the interpretive coding process in the humanities and the functional annotation practice in computer science, as well as the opportunities for combining these two paradigms, are discussed in Section 6.3.
4. **Analysis:** Researchers apply quantitative analysis techniques to recognize and understand patterns and relationships in their data. This step includes creating visualizations and enables a scalable workflow, allowing researchers to switch between quantitative statistics and qualitative insights. The application of purely quantitative methods is not widespread in primarily qualitative discourse research.
5. **Interpretation:** Finally, researchers interpret their (intermediate) results, draw conclusions, and generate insights that answer their research questions. This interpretive step constitutes the main effort of discourse analysis and can be digitized only to a limited extent.

Throughout this workflow, reflection and interpretation are continuous processes that occur in every phase of the analysis. The steps are not necessarily sequential but rather iterative, reflecting the dynamic and exploratory nature of discourse analysis, especially in Grounded Theory-based approaches, where the theory emerges inductively from the data itself. The dissertation's respective Chapters 5–8 explore the challenges and opportunities associated with the individual steps and highlight my contributions to the further development of DATS in addressing these aspects.

1.5 Requirements for a Discourse Analysis Tool Suite

Central requirements for a supportive work environment can be derived from the goals and challenges of the D-WISE project (Section 1.3), the methodology of the Sociology of Knowledge Approach to Discourse (Section 1.2), and the specific needs of the digital humanities user group. These criteria are decisive for the design and evaluation of the methods developed in this dissertation:

Supporting Discourse Analysis: The most fundamental requirement for a discourse analysis tool suite is the complete support for the iterative and hermeneutic research process, as described in Section 1.4. The work environment must provide researchers with a coherent, flexible digital environment to map the phases from corpus formation to interpretation without gaps.

Scalability: Given the large data volumes, *scalability* is the central technical challenge. The developed algorithms and software components must be capable of efficiently handling large and growing amounts of text.

User Control & Transparency: Since manual analysis is time-consuming, applying NLP methods to handle large data volumes is an interesting approach. Contextualized embeddings are a promising approach for exploring large datasets. This necessitates cooperation between humans and machines. The human-in-the-loop (HITL) principle (Holzinger 2016) is often mentioned in this context, meaning that the human is actively integrated into the machine-driven process. In the D-WISE project, we coined the term AI-in-the-loop (Koch et al. 2022) to emphasize that AI actively supports researchers in the iterative, hermeneutic circle of discourse analysis. Thereby, the researcher is placed at the center. Two central requirements for a discourse analysis tool suite arise from this cooperation. *User control* must ensure that researchers retain full authority over all analytical decisions. The AI is thus always to be seen as an augmentative tool, and never as a substitute for human interpretation. *Transparency* demands insight into the AI's decision-making so that researchers can understand the conclusions, thereby ensuring scientific validity.

Traceability: Closely linked to the requirement for *transparency* is the principle of *traceability*. Particularly in Grounded-Theory-guided interpretation processes and their presentation, it is important to be able to access the underlying data at all times to ground the theories with concrete evidence. *Traceability* thereby ensures scientific validity and reliability by guaranteeing that interpretations are grounded in the data and are not merely products of researchers. Consequently, *traceability* is another requirement: A tool suite for discourse analysis must enable all derived results, suggestions, or aggregations to be traceable to the source documents and text passages, thereby maintaining the connection between abstract categories and textual evidence.

Adaptability, Interactivity & Responsiveness: A chain of further requirements arises from the hermeneutic and Grounded Theory-based (iterative, data-driven) research methodology (such as the Sociology of Knowledge Approach to Discourse). Firstly, this is *adaptability*: the algorithms used must adapt continuously, incrementally, and iteratively to the (dynamic) concepts, categories, and research objects that evolve throughout the research process. The software must actively support this incremental interplay between technical adaptation and human knowledge gain. This leads to the requirement for *interactivity*: especially interactive methods and visualizations are necessary to enable researchers to engage in a flexible interplay among algorithm adaptation, knowledge gain, and further development of understanding. However, the development of such interactive systems directly entails a third, central requirement for the software: *responsiveness*. A smooth, user-friendly research process is only guaranteed if iterative and interactive systems function without delay.

Flexibility & Accessibility: Finally, the last requirements are derived from the target group of digital humanities researchers. Since digital humanities is an umbrella term for various disciplines, the research questions, objects of study, and materials of the individual projects are also highly diverse. Consequently, *flexibility* is a requirement for a discourse analysis tool suite: the developed methods must be either generally applicable or readily adaptable across different use cases and domains. This requires using general-purpose approaches and adapting them, rather than relying on highly specialized models optimized for a narrowly defined task. Another equally central requirement arises from the fact that researchers are often not experts in machine learning, NLP, or computer science. Therefore, *accessibility* is an essential requirement for the interfaces developed in such a tool suite: intuitive user interfaces should abstract complex processes to such an extent that a high level of usability is guaranteed for researchers without machine learning expertise.

These requirements are discussed throughout this dissertation and guide the methodological and technical solutions presented here.

1.6 Research Questions

This dissertation addresses the central challenges of the D-WISE project (Section 1.3) and specifies them in the context of the iterative research process (Section 1.4) and the methodical and technical requirements (Section 1.5). This work, in contrast to the multimodal project goal, focuses exclusively on text data. From these considerations results the leading question of this dissertation:

Lead Question

How to effectively apply NLP methods to advance discourse analysis methodologies to handle large datasets?

To answer this leading question, the contributions elaborated in this dissertation are closely aligned with and structured along the phases of the iterative, Grounded Theory-based research process. Therefore, the following six specific research questions (RQs) address challenges in the respective phases of discourse analysis and define the concrete scientific contribution of this work.

Exploration, Search & Organization: Efficient corpus formation and the exploration of large data volumes represent significant challenges of discourse analysis. Contextualized embeddings (e.g., vector representations of words, sentences, or documents, where semantically similar terms are placed close to each other in vector space, more details in Section 2.1) enable the visualization of large data volumes and thus offer a promising solution for bringing order to large-scale text corpora. Such a visual overview, which maps semantic similarities, pushes the boundaries of distant reading further. It enables the recognition of patterns, structures, and hidden features in extensive text collections that would otherwise remain concealed during close reading. This creates opportunities to explore,

organize, and structure documents. Therefore, the following research question is considered:

Research Question 1

How can vector-based document representations be leveraged to provide researchers with a flexible, interactive visualization of large-scale text corpora that enables initial explorative structuring and classification?

The recent success of large language models (LLMs) opens new opportunities for organizing text data, such as zero-shot document classification and information extraction. Their utility and applicability for discourse analysis are explored with this research question:

Research Question 2

How can large language models be utilized to support researchers with the organization of large datasets?

Annotation: Qualitative coding, the interpretive analysis of research material through close reading, has become a significant methodological bottleneck due to the vast volumes of digital discourse data now available. To maintain the hermeneutic depth of discourse analysis while processing larger corpora, new approaches for both researcher assistance and the scaling of qualitative insights are required.

In this context, AI assistance can serve a dual purpose. First, it can support researchers directly during the coding phase by utilizing the zero- and few-shot inference capabilities of large language models to suggest relevant concepts and patterns within the text. Second, it enables *Annotation Scaling*—the process of operationalizing these inductively developed codes into a formal schema and training supervised models to classify the remainder of a large-scale corpus. Exploring these methodologies represents a promising research direction to bridge the gap between human interpretation and computational scale.

Research Question 3

How can language models be used to assist researchers during the qualitative coding process and enable the scaling of these insights to large datasets?

Analysis: The scalable reading (Krautter 2024) approach to data analysis describes the interplay and integration of close reading and distant reading to merge interpretive qualitative results and computer-aided quantitative results. In this work, we aim to closely couple semantic search functionality (as a feature for distant reading) and manual annotation (as a feature for close reading) by exploring timeline analyses, which are particularly relevant for discourse researchers: a chronological overview of concepts such as events, topics, or frames enables researchers to identify patterns, trends, and shifts in discourse over time. By

visualizing such concepts along a timeline, developments can be traced and various elements within the corpus contextualized. Consequently, this research question is explored:

Research Question 4

How can semantic search and manual annotation be integrated into an interactive process to facilitate the semi-automated classification and analysis of concepts over time?

Interpretation: In the course of a discourse analysis, numerous research objects and results are compiled (documents, annotations, aggregations, notes). In large-scale projects, this can quickly become confusing. Especially when interpreting results, it is advantageous to connect the partial results and gain an overview. Furthermore, it is essential to trace findings back to the original data. This leads to the requirement for *traceability*. The following research question addresses the problem of a lack of overview and the necessity of linking research artifacts:

Research Question 5

How can research artifacts be meaningfully collected, linked to one another, and traced back to the original sources to create a working environment for hermeneutic interpretation?

Adaptability: The requirement for *adaptability* poses an overarching challenge, as it demands methods that adapt incrementally to constantly evolving research interests. This is relevant for all AI-augmented components, including visualization (RQ 1), annotation (RQ 2), and semantic search (RQ 4). In discourse analysis, this adaptation must specifically handle the scarcity of labeled data. Researchers in discourse analysis projects focus on interpretative coding. Their research never yields labeled data quantities typical of NLP datasets. They are not to be treated as data annotators. This means that adaptation must be achieved with few labeled examples, while high performance and rapid adaptation to new domains are simultaneously expected.

Research Question 6

How can AI-augmented methods—including visualization, annotation, and semantic search—be effectively designed within a human-in-the-loop framework to incrementally and rapidly adapt to evolving concepts and domain-specific inquiries in qualitative discourse analysis projects despite the scarcity of labeled data?

1.7 Research Methodology

While traditional NLP research often focuses on optimizing performance metrics—such as F1-score or Accuracy—on static, standardized benchmarks, the research conducted in this dissertation follows a more applied, iterative, and human-centered methodology. The primary objective is to bridge the gap between state-of-the-art NLP technology and the practical needs of digital humanities researchers, particularly in the field of digital discourse analysis. This research process can be organized into four idealized stages:

1. Conceptualization: My research process often begins with identifying state-of-the-art NLP technology and recent breakthroughs in the field, such as large language models, instruction-based embeddings, or parameter-efficient fine-tuning. Instead of testing these technological advancements on existing NLP leaderboards, I conceptualize a feature that could leverage them to address a known bottleneck in qualitative research workflows (e.g., scaling annotations or exploring massive document collections).

2. Simulation-Driven Development: Before a feature is integrated into the DATS platform, it undergoes a usage simulation phase. In this stage, I evaluate the model's performance on datasets that mimic the real-world complexity of discourse analysis. In particular, I evaluate how the model would perform under constraints typical of qualitative research, most often limited initial amounts of labeled data. Based on these simulations, I refine the technical approach. This may involve selecting a different model, adjusting hyperparameters, or other steps.

3. Human-Centered Implementation: Once the technical approach is stabilized and optimized through simulation, it is implemented as a functional module within the DATS platform. A core focus during this phase is initial usability. The goal is to transition the "raw" NLP logic into an interface that allows a non-technical researcher to interact with the underlying technology, following the ideal of accessibility through interactivity. This stage often involves developing specialized frontends that translate the NLP technologies into problem-solving tools.

4. Expert Feedback and Collaborative Refinement The final and most critical stage involves handing the tool over to project partners (researchers in the digital humanities) for testing on their actual research data. This collaborative phase triggers a feedback loop focused on three key areas:

1. **UI/UX:** Identifying points of friction where the technical implementation does not match the researcher's mental model or workflow.
2. **Transparency:** Researchers often ask for the "methodology behind the tool". For example, they do not just need a label; they need to understand why the model made a certain suggestion. This leads to further development of transparency features.

3. **Functional Alignment:** Ensuring the implemented feature (e.g., AI assistance) genuinely addresses the core of the discourse analytical task, rather than providing functionality for peripheral or unrelated problems.

This back-and-forth loop ensures that the final contribution is not just a well-performing algorithm, but a research instrument that empowers scholars to handle the growing volume and complexity of digital data relevant to their analysis.

1.8 Contributions & Publications

This dissertation presents my contributions to advancing digital discourse analysis and natural language processing, with a particular focus on developing and enhancing the Discourse Analysis Tool Suite. Throughout my research, I have focused on developing and integrating novel tools that leverage language models' capabilities to support researchers in their qualitative endeavors. These contributions are detailed in the following publications.

1.8.1 Core Publications

The D-WISE Tool Suite: Multi-Modal Machine-Learning-Powered Tools Supporting and Enhancing Digital Discourse Analysis (Schneider et al. 2023). First, in this paper, we provided an overview of the DATS platform. This platform, developed in close collaboration with cultural anthropologists, addresses the limitations of existing tools in handling the growing volume and complexity of multimodal data. The DATS platform integrates machine learning techniques from natural language processing and computer vision to facilitate comprehensive discourse analysis. It supports various functionalities, including multimodal data import, automated annotation pipelines, lexical and semantic search, manual annotation, and interactive visualizations. This work lays the groundwork for subsequent contributions.

Perspectives: Interactive Document Tagging in the Discourse Analysis Tool Suite (Fischer and Biemann 2026). Building on this foundation, I presented the *Perspectives* extension to the DATS platform in this paper, which enables researchers to explore and organize extensive, unstructured document collections on an interactive 2D map. I implemented a flexible, aspect-focused document clustering pipeline with human-in-the-loop (HITL) refinement capabilities. The clustering process is initially guided by LLM-driven document rewriting and instruction-finetuned embedding models to define analytical lenses that focus the analysis on user-defined aspects. The system centers on an interactive 2D document map for exploration. It provides refinement tools — such as splitting, merging, and adding clusters — as well as a mechanism for few-shot fine-tuning of the embedding model to further align results with user intent. I evaluated the effectiveness of document rewriting, instruction-tuned embeddings, and few-shot fine-tuning across multiple datasets and tasks, demonstrating that task-specific instructions consistently yield superior performance and that document rewriting is often beneficial. This work addresses Research Questions 1 and 6 by building a tool

for visualizing extensive document collections using contextualized embeddings, adaptable to user preferences via few-shot fine-tuning of the embedding model.

Exploring Large Language Models for Qualitative Data Analysis (Fischer and Biemann 2024). In this paper, I investigated the potential of LLMs to enhance qualitative data analysis (QDA) workflows within the DATS platform. I identified key opportunities for AI assistance in the QDA process and translated them into corresponding natural language processing tasks. To evaluate the effectiveness of LLMs in this context, I created a benchmark comprising English and German datasets aligned with relevant QDA use cases. I then assessed the performance of three leading open-weight LLMs on this benchmark, demonstrating their potential to streamline and improve QDA processes. Finally, I integrated AI assistance in the form of the *LLM Assistant* into DATS. This work addresses Research Questions 2 and 3, which focus on evaluating LLM capabilities across various QDA tasks: document categorization, information extraction, and text annotation.

Semi-automatic Sequential Sentence Classification in the Discourse Analysis Tool Suite (Fischer and Biemann 2025). In this paper, I introduced a three-phase approach to sequential sentence annotation. This approach leverages LLMs to provide researchers with optimal suggestions during the coding process by dynamically adapting its strategy based on the number of available annotations, using zero-shot prompting, few-shot prompting, or fine-tuned models to optimize its recommendations. I evaluated this approach with a benchmark comprising five diverse datasets, demonstrating its effectiveness in facilitating efficient sequential sentence annotation. Finally, I integrated this three-staged approach into the previously introduced *LLM Assistant*. This contribution addresses Research Question 3 by improving coding efficiency in QDA workflows and Research Question 6 by developing an approach to progressively adapt AI suggestions as the number of available annotations grows.

Concept Over Time Analysis: Unveiling Temporal Patterns for Qualitative Data Analysis (Fischer, Schneider, Geislinger, et al. 2024). In this paper, I presented an extension to the DATS platform that enables researchers to analyze the evolution of concepts over time. This tool allows users to define, refine, and visualize their concepts of interest within an interactive interface. By employing few-shot sentence classification based on sentence transformer embeddings, the system facilitates an iterative process of semantic similarity searches, sentence annotation, and model refinement. The final output is a timeline visualization of sentences classified into concepts, providing researchers with valuable insights into temporal patterns and trends. This contribution addresses Research Question 4 by integrating semantic search and annotation in a human-in-the-loop manner. Also, it addresses Research Question 6 by exploring a few-shot method for handling the limited number of available annotations.

Extending the Discourse Analysis Tool Suite with Whiteboards for Visual Qualitative Analysis (Fischer, Schneider, Petersen-Frey, et al. 2024). Finally, in this paper, I introduced a visualization tool called *Whiteboards* to the DATS

platform. This tool uses interactive graph-based interfaces to empower researchers to organize and manipulate various types of qualitative data. *Whiteboards* offer a customizable view of research materials and support a wide range of actions, opening new possibilities for interaction and analysis. I demonstrated the versatility of this tool by showcasing its utility across various QDA use cases, including reflection on the research process, creation of actor networks, and refinement of code taxonomies. This contribution addresses Research Question 5 by presenting a flexible tool for interpretative processes that fully supports *traceability*.

1.8.2 Related Publications

During my research, I also worked on some related projects, which are cited in this dissertation.

Dimensions of Similarity: Towards Interpretable Dimension-Based Text Similarity (Hatzel et al. 2023). In this paper, we introduced a method for achieving interpretable, configurable semantic similarity search by training sentence embedding models to identify textual similarity along specific aspects or dimensions. Drawing on the SemEval 2022 Task 8 dataset (Chen et al. 2022), which is annotated along seven dimensions of similarity (including entity, tone, and style), we fine-tuned bi-encoder models to specialize in individual dimensions. We proposed a conceptual configurable similarity search system that enables users to find documents that are similar in specific aspects (e.g., high similarity in entities and time) but dissimilar in others (e.g., low similarity in style and tone). The usefulness of this approach was demonstrated through downstream retrieval and classification tasks on various datasets. This work showed that fine-tuning individual models for each dimension (multi-model approach) significantly outperformed a single multi-task learning model.

AnnoPlot: Interactive Visualizations of Text Annotations (Fittschen et al. 2024). In this paper, we presented *AnnoPlot*, a web application for analyzing, managing, and visualizing annotated text data. We developed an approach that uses pre-trained transformer models to compute high-dimensional embeddings of span annotations and then applies dimensionality reduction (UMAP, McInnes et al. (2020)) to offer users an interactive 2D scatter plot view of their datasets. The tool is particularly suited for the digital humanities, enabling researchers to gain an overview of large annotated datasets, reorganize categories, and identify erroneous annotations. The key feature is a dynamic UMAP model that can be iteratively refined through user interactions, such as training on annotated categories or dragging data points. *AnnoPlot* also includes a category view for visualizing category relationships and facilitating taxonomy management tasks, such as merging. Finally, we demonstrated that our HDBSCAN (Campello et al. 2013) based automatic error detection performs significantly better with dynamic UMAP reduction compared to static UMAP.

From Qualitative to Quantitative Research: Semi-Automatic Annotation Scaling in the Digital Humanities (Petersen-Frey et al. 2023). In this paper, we proposed

and evaluated assistive methods for semi-automatically scaling a small number of manual annotations to large corpora. This work targeted qualitative researchers who want to increase their efficiency and leverage large datasets to provide quantitative grounding for their hypotheses. Our approach uses a two-stage method that requires no programming or NLP expertise. The first stage is interactive, semi-automatic annotation scaling. Here, we use contextualized embeddings of user-provided annotations to find similar occurrences in the corpus and provide adaptive suggestions. Users can accept or discard suggestions in batches, making it easier and faster than manual annotation. The core of this stage is an advanced retrieval strategy that uses both positive and negative support sets to generate higher-quality suggestions. The second stage is fully automatic annotation scaling. Once enough examples are collected, the system uses the labeled data to train a classifier that applies the learned categories to the extensive document collection. For the final high-quality classifier, we employ computationally and sample-efficient adapters. Our evaluation across multiple datasets for sentence- and word-level annotations demonstrated strong scalability. We also conceptualized a user interface to help researchers scale their annotation-based research, bridging the gap between qualitative and quantitative results.

1.9 Dissertation Overview

Chapter	RQs	Core Papers
4 DATS	Foundation	D-WISE Tool Suite (Schneider et al. 2023)
5 Search	RQ1, RQ2, RQ6	Perspectives (Fischer and Biemann 2026) LLMs for QDA (Fischer and Biemann 2024)
6 Annotation	RQ3, RQ6	Sentence Classification (Fischer and Biemann 2025) LLMs for QDA (Fischer and Biemann 2024)
7 Analysis	RQ4, RQ6	COTA (Fischer, Schneider, Geislinger, et al. 2024)
8 Interpretation	RQ5	Whiteboards (Fischer, Schneider, Petersen-Frey, et al. 2024)

Table 1.1: Contributions of the core papers to the research questions of this dissertation, organized per chapter.

The structure of this dissertation is closely aligned, both methodologically and thematically, with the iterative five-step workflow of discourse analysis (Section 1.4), with the individual chapters treating the respective steps of the work process in detail. Table 1.1 gives an overview of my core papers, organized by research questions and chapter. The following two chapters establish the necessary knowledge base for this interdisciplinary work.

Chapter 2 NLP introduces the technical foundations of the methods used, including contextualized embeddings and large language models, which are utilized in almost all contributions of this work.

Chapter 3 Empirical Social Sciences explains the most important concepts of the empirical social sciences that form the framework for this dissertation, presenting discourse analysis and the Sociology of Knowledge Approach to Discourse in detail.

Chapter 4: Discourse Analysis Tool Suite introduces DATS, which constitutes a collaborative contribution and the technical framework of this dissertation. This chapter explains the platform’s architecture and addresses the first step in the discourse analysis workflow: data import & preprocessing.

The subsequent Chapters 5 to 8 form the core of the dissertation by presenting the contributions and addressing each of the remaining four steps of the discourse analysis workflow:

Chapter 5: Exploration, Search & Filtering deals with the explorative phase and the efficient reduction of large data volumes. It is dedicated to RQ 1 and RQ 2 by presenting the development of visualizations based on contextualized embeddings and LLM-based organization approaches to support distant reading and corpus structuring. Answers to the overarching RQ 6 are also provided here, as the visualization is implemented to be adaptable through few-shot learning.

Chapter 6: Annotation focuses on the challenge of coding large quantities of text. This chapter addresses RQ 3 by investigating the use of language models to augment coding processes. It also provides answers to the overarching RQ 6 by comparing in-context few-shot learning and fine-tuning approaches.

Chapter 7: Analysis places quantitative analysis in the foreground. It answers RQ 4 by implementing a timeline analysis feature that closely couples semantic search and manual annotation within the human-in-the-loop paradigm. Answers to the overarching RQ 6 are also provided here, as the semantic search functions are adaptable to evolving concepts using a few-shot learning approach.

Chapter 8: Interpretation addresses the concluding step of the research process. Here, RQ 5 is addressed by presenting digital whiteboards for the collection, linking, and seamless traceability of research artifacts, thus creating a workspace for hermeneutic interpretation.

Chapter 9: Conclusion summarizes the results of this work, discusses the limitations of the developed approaches, and provides an outlook on future research directions in the field of NLP-supported discourse analysis.

To address the challenges of scaling qualitative discourse analysis to large-scale data volumes, it is first necessary to understand the computational natural language processing methods that enable this undertaking. The following chapter provides the theoretical and technical foundations. We begin by examining the evolution of natural language processing, from early vector representations of meaning to the transformative architectures of large language models, providing

the technical vocabulary and concepts required to understand the subsequent contributions of this work.

2

Background – NLP

Contents

2.1	Embeddings	20
2.1.1	Lexical Semantics	20
2.1.2	Vector Semantics	21
2.1.3	Sparse, Count-Based Embeddings	21
2.1.4	Dense Embeddings: word2vec	21
2.1.5	Cosine Similarity	22
2.1.6	Limitations	22
2.1.7	Visualizing Embeddings	22
2.1.8	Text Clustering	23
2.2	Language Models	24
2.3	Large Language Models	26
2.3.1	Prompting	26
2.3.2	Autoregressive Generation	26
2.3.3	Training Pipeline	27
2.3.4	Evaluation	29
2.3.5	Limitations	29
2.4	Transformer	30
2.4.1	Attention	31
2.4.2	The Transformer Block	32
2.4.3	Input Encoding	33
2.4.4	Language Modeling Head	34
2.4.5	Training	34
2.4.6	Computational Efficiency	35
2.5	Masked Language Models	36
2.5.1	Architecture	37
2.5.2	Pretraining	37
2.5.3	Contextual Embeddings	38
2.5.4	Fine-Tuning for Downstream Tasks	39
2.6	LLMs used in the Dissertation	41
2.6.1	Llama 3.1	41

2.6.2	Mistral NeMo	42
2.6.3	Gemma 2	42
2.6.4	Gemma 3	42

This chapter establishes the technical foundation for the natural language processing (NLP) methods utilized throughout this dissertation. To effectively augment discourse analysis with machine learning, one must understand how computational systems represent and process human language. We begin with the concept of embeddings (Section 2.1), exploring how lexical semantics can be mapped into a vector space through methods like word2vec (Section 2.1.4) and visualized using dimensionality reduction methods (Section 2.1.7).

The chapter then explains language models (Section 2.2) and details the current era of large language models (Section 2.3). We discuss the transformer architecture (Vaswani et al. 2017) in Section 2.4, which serves as the structural backbone for both the generative decoder models and the masked language models (Section 2.5). By reviewing the parameter-efficient fine-tuning (Section 2.4.5) via LoRA (Hu et al. 2022) and fine-tuning techniques for downstream tasks (Section 2.5.4), this chapter provides the necessary framework for the AI-augmented tools developed in the Discourse Analysis Tool Suite (DATS).

2.1 Embeddings

2.1.1 Lexical Semantics

The modeling of word meaning presents a fundamental challenge in natural language processing. Traditional NLP (like n-gram models) and classical semantics represent a word as a unique symbol. This classical representation fails to capture any inherent meaning, similarity, or relationship between words (e.g., it cannot inherently model that "cat" is similar to "dog").

A domain known as lexical semantics studies the meanings of words and how they relate to one another. Moving beyond simple symbolic representations, a robust model must capture a word's complete semantic profile. In this research area, key aspects of word meaning have been established, all of which a model must address: *word similarity*, *word relatedness*, and *connotation*.

Word similarity defines the degree of semantic interchangeability between two words. This differs from synonymy. While synonymy implies near-identical meaning (e.g., "couch/sofa"), most words share varying degrees of similarity (e.g., "cat" and "dog"). In contrast, *word relatedness* (or association) describes a broader, non-interchangeable connection. Words are related if they co-participate in a common event or belong to the same semantic field (e.g., "coffee" and "cup" are highly related, but are not similar). Finally, *connotation* represents the affective dimension of a word. It defines its sentiment or emotional value (e.g., "excellent" vs. "miserable").

2.1.2 Vector Semantics

To effectively model the complex, multi-faceted aspects of lexical semantics, modern NLP employs *vector semantics*, which represents words as points in a multidimensional space. This approach is based on the *distributional hypothesis*, which posits that words that occur in similar contexts tend to have similar meanings (Harris 1954), famously summarized as "*You shall know a word by the company it keeps*" (Firth 1957). This link between distribution and meaning is the principle that enables *representation learning* (Bengio et al. 2013) – the automated process of generating semantic representations directly from raw text. The resulting numeric vector representations of word meaning are known as embeddings.

Embeddings are learned directly from the distribution of words in texts. This kind of self-supervised learning is an important principle of modern NLP and is also used to train large language models (LLMs), which are discussed in Section 2.3. Embedding models are typically divided into two categories: static and contextualized embeddings. Static embeddings, such as word2vec (Mikolov, Chen, et al. 2013) (Section 2.1.4), GloVe (Pennington et al. 2014), fastText (Bojanowski et al. 2017), assign one fixed, context-independent vector to each word in the vocabulary. For example, this means the vector for "*bank*" is the same regardless of whether it refers to a financial institution or a river edge. Contextualized embeddings (e.g., BERT (Devlin et al. 2019)), by contrast, generate a different vector for a word depending on its specific context. The models discussed in the following sections are static; contextual embeddings are explained in Section 2.5.

2.1.3 Sparse, Count-Based Embeddings

The most straightforward approach to vector semantics is to explicitly count word co-occurrences within a defined context window. Here, each row represents a target word w (from the vocabulary V), each column represents a context word c (from the same vocabulary V), and each cell records the number of times w and c co-occur nearby within a defined context window (e.g., ± 4 words). Such models' embeddings have high dimensionality: the matrix is $|V| \times |V|$, yielding very long vectors. Since most words do not co-occur, these representations are intrinsically sparse. This means most elements of the vector representation are zero. While raw counts are sufficient, performance has been improved by applying weighting schemes such as TF-IDF (Sparck Jones 1972).

2.1.4 Dense Embeddings: word2vec

Sparse vectors are computationally inefficient and may fail to capture nuanced semantic relationships. Consequently, methods for computing dense embeddings (shorter, real-valued vectors) have been developed and have shown better performance on NLP tasks. The popular word2vec framework creates such dense vector representations. Two model architectures have been developed: Continuous bag-of-words (Mikolov, Chen, et al. 2013) and skip-gram with negative sampling (Mikolov, Sutskever, et al. 2013).

Skip-Gram trains a binary classifier to predict: *"Is context word c likely to show up near target word w ?"*. The learned classifier weights are then used as word embeddings.

The model is trained in a self-supervised way, using text as implicit training data: A word c found near w is a positive example; a randomly sampled word c_{neg} is a negative example. The training objective is to adjust the embeddings to maximize the similarity for positive examples ($c_{pos} \cdot w$) and minimize the similarity for negative examples ($c_{neg} \cdot w$). The following section explains that the dot product ($\cdot$) can be used to measure semantic similarity. Stochastic gradient descent (SGD) (Robbins and Monro 1951; Bottou 2010) is used to minimize the training objective. During training, SGD iteratively updates the embeddings, moving w closer to the embeddings of positive context words (c_{pos}) and farther from those of negative context words (c_{neg}).

2.1.5 Cosine Similarity

To quantify the similarity between two word vectors, v and w , the dot product ($\cdot$) can be used. However, it is inherently biased toward longer vectors (e.g., when using count-based embeddings, this metric would favor high-frequency words). Cosine similarity is the de facto standard metric, as it corrects this by normalizing the dot product by the magnitude of the vectors:

$$\text{cosine}(v, w) = \frac{v \cdot w}{\|v\| \|w\|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

Geometrically, this metric measures the cosine of the angle between the two vectors. A higher cosine (up to 1) indicates a smaller angle and greater similarity, while a cosine of 0 indicates orthogonality.

2.1.6 Limitations

Embeddings possess an interesting semantic property: *relational similarity*, where the difference between two vectors encodes a specific relationship. This is the popular vector arithmetic example by Mikolov, Sutskever, et al. (2013), indicating that the vector difference captures the MALE-FEMALE relation: **king** – **man** + **woman** $\approx$ **queen**.

However, embeddings also have limitations. They reproduce and often amplify implicit biases and stereotypes present in the training data. This can lead to allocational harm (e.g., unfair resource distribution) and representational harm (e.g., demeaning social groups). For instance, the analogy **computer_programmer** – **man** + **woman** might favor a homemaker over other occupations.

2.1.7 Visualizing Embeddings

High-dimensional vectors require specialized methods for visualization. A common technique is *dimensionality reduction*, projecting the vectors down to two or three dimensions for graphical display. Common non-linear projection techniques include:

- t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton 2008)
- Principal Component Analysis (PCA) (Pearson 1901; Hotelling 1933)
- Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2020)

UMAP

In this thesis, UMAP is the primary tool for dimensionality reduction. UMAP is a graph-based dimensionality reduction technique that preserves both the local and global topological structure of high-dimensional datasets. The algorithm first constructs a k-nearest-neighbor graph, where edge weights represent the probability that two points are connected given their local density. To account for the *curse of dimensionality* (the phenomenon where distance metrics become increasingly indistinguishable as the number of dimensions grows), UMAP employs a local scaling normalization that ensures each point is connected to at least its closest neighbor. This results in a weighted graph that captures the underlying geometric structure of the data. In the second phase, the algorithm optimizes a low-dimensional embedding—typically using stochastic gradient descent—to minimize the cross-entropy between the high-dimensional graph and its low-dimensional counterpart, ensuring that the probabilistic connectivity of the original data is reflected in the projection. This approach allows UMAP to capture non-linear relationships with lower computational overhead than comparable methods, such as t-SNE.

2.1.8 Text Clustering

While visualization enables exploration of vector spaces, text clustering provides an automated method for discovering latent structures within a corpus. Text clustering is an unsupervised machine learning task. Unsupervised learning means a machine learning algorithm analyzes unlabeled data and learns the data’s inherent structure on its own. Text clustering, consequently, aims to automatically group a set of texts (e.g., documents or sentences) such that items within the same cluster are more semantically similar to each other than to those in other clusters (Aggarwal and Zhai 2012).

In this dissertation, clustering is primarily applied to dense, contextualized embeddings that have been reduced in dimensionality via UMAP. This approach (popularized by Top2Vec (Angelov 2020) and BERTopic (Grootendorst 2022)) is motivated by the observation that many high-dimensional datasets actually lie on a lower-dimensional structure; by reducing dimensionality first, clustering algorithms can operate more efficiently and often achieve higher accuracy by mitigating the effects of noise and the *curse of dimensionality*. This enables the discovery of inherent clusters and patterns in text data solely from the distributional properties of vector representations.

HDBSCAN

The primary clustering algorithm used in this work is Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (Campello et al. 2013). HDBSCAN is a density-based algorithm that extends DBSCAN (Ester et al. 1996) by converting it into a hierarchical clustering algorithm and then using a technique to extract a flat clustering based on cluster stability.

HDBSCAN offers two key advantages that make it particularly suitable for NLP tasks: Unlike traditional centroid-based algorithms such as K-Means (MacQueen 1967), which assume spherical clusters and require the number of clusters (k) to be specified in advance, HDBSCAN automatically infers the number of clusters. This has high practical relevance, as the number of clusters (e.g., topics) is rarely known in advance, and clusters often vary in shape and density. Further, HDBSCAN defines clusters as regions of high density separated by regions of low density, allowing for robust noise handling: it does not force every data point into a cluster but instead identifies low-density points as outliers. This is particularly valuable for text datasets, where many documents or utterances may be semantically isolated. HDBSCAN identifies these points as outliers rather than forcing them into clusters where they do not belong, thereby increasing the semantic purity of the resulting groups.

The algorithm first computes a mutual reachability distance to estimate the data density. It then constructs a minimum spanning tree, which is transformed into a hierarchy of connected components. To simplify this hierarchy, HDBSCAN condenses the tree by discarding small branches that do not meet a minimum cluster size threshold. Finally, it selects the most stable clusters from the condensed tree—those that persist the longest across various density levels—yielding the final clustering.

2.2 Language Models

A language model (LM) is fundamentally a machine learning model designed to predict upcoming words. It achieves this by assigning a probability distribution $P(w_i | w_{<i})$ over all possible next words w_i , conditioned on the preceding word sequence $w_{<i}$.

The advent of large language models represents a significant shift in the field. LLMs are distinguished by their massive scale (with billions of parameters and trained on trillions of tokens), allowing them to capture vast amounts of linguistic and world knowledge that far exceed those of earlier models. At the time of writing, Kimi K2 (K Team 2025) is the largest open-weight LLM with 1 trillion total parameters trained on 15 trillion tokens. This unprecedented capability stems from their pretraining, a self-supervised process in which parameters are adjusted by iteratively performing the language modeling task (here: next-token prediction) across a massive corpus, with the correct label (the next word) automatically derived from the data sequence itself. As this process requires no human labeling, large amounts of training data can be (easily) acquired.

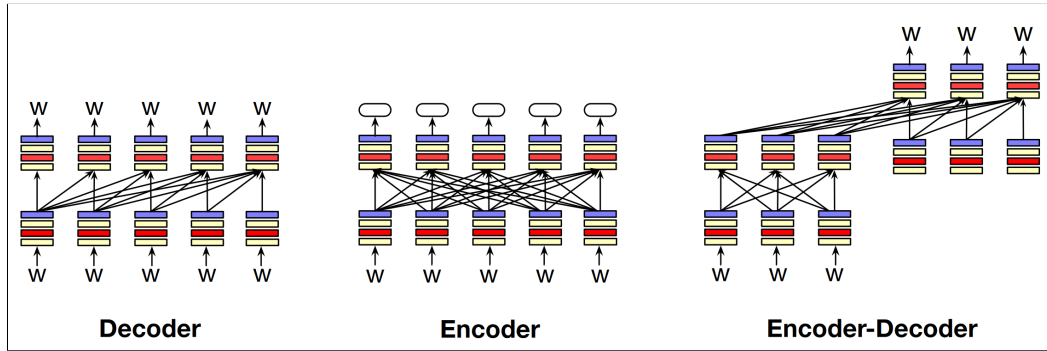


Figure 2.1: Language model architectures: decoders, encoders, and encoder-decoders. The arrows indicate the information flow. All architectures take tokens as input, but decoders generate a token as output, encoders produce an encoding of each token, and encoder-decoders generate a series of tokens as output. Image from Jurafsky and Martin (2025).

Model Architectures

Modern NLP uses three key architectural variants for training language models, all derived from the transformer architecture: the decoder, the encoder, and the encoder-decoder (see Figure 2.1).

- **Decoder (Autoregressive Models):** This architecture is the basis for most modern LLMs, such as GPT (Brown et al. 2020; OpenAI 2023), Llama (Touvron et al. 2023; Grattafiori et al. 2024; MetaAI 2025), and Kimi K2 (K Team 2025). It generates output text one token at a time, moving from left to right. Since it is trained to predict the next token from prior tokens only, its information flow is causal and unidirectional. This architecture powers the era of generative AI, enabled by the highly versatile task of conditional text generation.
- **Encoder (Masked Language Models):** This architecture consumes an input sequence and produces a dense vector representation (an encoding) for each token. It is used in models such as BERT (Devlin et al. 2019) and RoBERTa (Y Liu et al. 2019). It is trained by masking out a word and predicting it using context from both the left and the right, giving it a non-causal (bi-directional) information flow. Encoders are typically used for non-generative tasks, such as text classification, after supervised fine-tuning.
- **Encoder-Decoder:** This architecture maps an input sequence to an output sequence that may have a loose relationship, such as between different languages or formats. Its primary use is in tasks like machine translation (Vaswani et al. 2017) and speech recognition.

This dissertation utilizes both decoder and encoder models. Both are explained in the following sections.

2.3 Large Language Models

A large language model is a neural network trained to predict the next word given a context or prefix. A model capable of prediction is also inherently capable of generating text by repeatedly sampling from its output probability distribution. When trained in a certain way (instruction tuning and preference alignment (see Section 2.3.3), this even enables LLMs to function as conversational agents.

2.3.1 Prompting

Conditional generation is the task of generating text conditioned on an input text. Having a model capable of this task is extremely useful, as it enables functional versatility: almost any NLP task can be reformulated as a word-prediction task (e.g., classification, question answering, summarization).

For example, sentiment analysis can be framed as a token-prediction problem. Given the prompt, "*The sentiment of the sentence 'I like Jackie Chan' is:*", the model assigns probabilities to the immediate next tokens. The sentiment can then be determined by comparing the conditional probability of the target tokens y :

$$P(y = \text{'positive'} \mid x) \quad \text{vs.} \quad P(y = \text{'negative'} \mid x)$$

where $x = (x_1, x_2, \dots, x_n)$ represents the sequence of tokens in the prompt.

Users interact with LLMs via prompts. The discipline of designing, refining, and optimizing these input strings for a given task is known as *prompt engineering* (P Liu et al. 2023; Schulhoff et al. 2025). However, in the experiments in this thesis, no prompt engineering or prompt optimization is conducted. LLM behavior is often guided by a *system prompt*, a lengthy, initial instruction added at the start of every user interaction that defines the model's overall role, tone, and rules (e.g., "*You are a helpful and knowledgeable assistant...*").

Furthermore, LLMs exhibit in-context learning (ICL) (Brown et al. 2020), the ability to learn a new task and improve performance by observing labeled examples (demonstrations) provided in the prompt itself, without requiring gradient-based updates to the model's parameters. Hence, researchers often distinguish between few-shot prompting and zero-shot prompting. In the zero-shot setting, only the instruction is provided to the model. In the few-shot setting, several labeled examples are included in the prompt to demonstrate the required task format and logic.

2.3.2 Autoregressive Generation

Autoregressive generation is an iterative process in which a large language model (LLM) predicts the next token from the sequence of previously generated tokens. Given a context sequence, the model produces a vector of logits—unnormalized log-probability scores $z \in \mathbb{R}^{|\mathcal{V}|}$ where $|\mathcal{V}|$ represents the vocabulary size. These logits are transformed into a categorical distribution via the softmax function, defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{|\mathcal{V}|} e^{z_j}} \quad \text{for } i = 1, \dots, V$$

While the resulting values are often treated as a probability distribution, it is important to note that they represent model confidence scores rather than true probabilities. The subsequent selection of the next token is the task of *decoding*, and the overall, iterative left-to-right generation process is called *autoregressive generation*. The most straightforward approach, *greedy decoding*, always chooses the token with the highest score (or maximum logit value). However, this deterministic method often produces repetitive and generic text, making it unsuitable for most generative tasks.

An alternative decoding method for tasks like machine translation is *beam search*. Unlike greedy methods, *beam search* (Lowerre 1976) maintains a fixed number (the beam width) of candidate sequences, seeking to maximize the joint probability of the entire sequence rather than just the immediate next token. This results in a globally better output with respect to the sequence-level likelihood.

Sampling methods are another approach to introducing stylistic diversity. Here, the next token is selected stochastically based on softmax-normalized "probability" scores. However, this process risks generating statistically improbable, nonsensical text from the distribution's long tail.

Temperature sampling (Ackley et al. 1988) established itself as a commonly used and effective decoding method. It balances the predictability of *greedy decoding* with the diversity of random sampling by introducing a temperature parameter $\tau \in (0, 1]$. The logits u are divided by τ before the softmax:

$$y = \text{softmax}(u/\tau)$$

Lower values of τ (closer to 0) sharpen the distribution, while higher values (closer to 1 or greater) flatten the distribution, increasing randomness.

Other sampling methods used to mitigate the risk of sampling from the low-probability tail were investigated by Holtzman et al. (2020) and include *top-k sampling* (restricting selection to the k most likely tokens) and *top-p sampling* (selecting the smallest set of tokens whose cumulative probability exceeds a threshold p).

2.3.3 Training Pipeline

Modern LLMs are developed through a comprehensive multi-stage training process, which turns them from a generic language model to an interactive, generative assistant:

Stage 1: Pretraining (Self-Supervised Learning). In this first stage, the model acquires world and linguistic knowledge. The pretraining task is *next-token prediction* on massive sequences $w_{1:T}$: At each step $t \in T$, the model is conditioned on the correct preceding tokens $w_{1:t}$ and predicts w_{t+1} . Importantly, this pretraining is self-supervised. The supervision signal (the correct answer) is taken from the input text. Furthermore, this training process uses *teacher forcing*, ensuring the model always sees the actual prior sequence (rather than its previously predicted sequence) to promote stable convergence. *Error backpropagation* (Rumelhart et al. 1986) and *gradient descent* are used to adjust the model's parameters, including word embeddings, to minimize *cross-entropy loss*.

The *cross-entropy loss* is used for classification and language modeling tasks. It measures the difference between two probability distributions: the correct answer (actual distribution) y and the predicted answer (the predicted distribution of the trained model) $\hat{y}$. For a multi-class problem (like predicting the next word), the *cross-entropy loss* is defined as the sum over all classes C (the vocabulary defines the classes, so $C = V$):

$$\mathcal{L}_{CE} = - \sum_{i=1}^C y_i \log(\hat{y}_i)$$

This formula can be simplified. Since the correct answer in classification tasks (and next-token prediction tasks) is always a particular class (or token from the vocabulary), y_i is 1 for the correct class and 0 for all others:

$$\mathcal{L}_{CE} = -1 \cdot \log(\hat{y}_{\text{correct}}) + \sum_{i \neq \text{correct}} 0 \cdot \log(\hat{y}_i) = -\log(\hat{y}_{\text{correct}})$$

This is called the *negative log-likelihood loss* and is in this context mathematically equivalent to the *cross-entropy loss*.

In short, the language model processes a sequence of tokens $w_{1:T}$, attempts to predict the next token for every token position t in this input sequence, and the loss is the negative log probability the model assigns to the next, correct word w_{t+1} in the sequence:

$$\mathcal{L} = -\log(P(w_{t+1}|w_{1:t}))$$

The final training loss is then the average of the losses computed at each token position across the entire batch.

Primarily, uncurated text scraped from the web (e.g., Common Crawl¹), augmented by curated sources (books, Wikipedia, academic papers), is used as training data. Examples are the Colossal Clean Crawled Corpus (Raffel et al. 2020) and the Pile (Gao et al. 2020). This extensive, varied data is helpful because it contains implicit examples of many downstream NLP tasks (e.g., summaries, question answering). Still, creating these datasets poses many challenges. A recent example is test set leakage (Xu et al. 2024), a risk in which the training data may include public test sets (e.g., verbatim questions and answers from popular benchmarks), leading to overestimation of the model’s actual generalization performance.

Stage 2: Instruction Tuning. Following the pretraining phase, the model undergoes instruction tuning to transition from a general-purpose sequence completer to a task-oriented assistant. The pretrained model is adapted to follow human instructions, answer questions, and behave like an assistant. This stage involves supervised fine-tuning on curated datasets of instruction-response pairs (e.g., question answering pairs, command-response pairs), enabling the model to follow human commands and perform zero-shot tasks (Wei et al. 2022). As established in the InstructGPT training pipeline (Ouyang et al. 2022), this process typically utilizes a standard cross-entropy loss to maximize the likelihood of the desired response tokens given the instruction prompt.

1. <https://commoncrawl.org/>

Stage 3: Preference Alignment. The final stage aims to align the model outputs with human values, making it as helpful, harmless, and honest as possible. This is traditionally achieved through reinforcement learning from human feedback, where a reward model is trained on human preference rankings and the LLM is optimized using the proximal policy optimization (PPO) algorithm (Ouyang et al. 2022). In this actor-critic framework, the LLM acts as the policy, while the reward model provides the signal for optimization.

More recently, the direct preference optimization algorithm (Rafailov et al. 2023) has emerged as a widely adopted alternative. Unlike the actor-critic approach, this algorithm eliminates the need for an explicit reward model and reinforcement-learning loops by directly optimizing the policy to satisfy preference constraints via a classification loss. This results in a more stable and computationally efficient alignment process while maintaining, or often exceeding, the performance of the previous method.

2.3.4 Evaluation

Evaluation of LLMs typically relies on three factors. The primary metric for measuring how well a language model predicts unseen text is *perplexity*. A lower perplexity of the model with respect to the sequence indicates a better model. *Perplexity* is the exponentiated average negative log-likelihood (NLL) of a sequence of tokens. Given a sequence of length T , $W = (w_1, w_2, \dots, w_T)$, the average NLL is:

$$\mathcal{L} = \frac{1}{T} \sum_{i=1}^T -\log P(w_i \mid w_1, \dots, w_{i-1})$$

Then perplexity is calculated by exponentiating the average NLL $\mathcal{L}$:

$$\text{PPL} = e^{\mathcal{L}}$$

Second, downstream task performance (e.g., question answering, reasoning, translation) is measured using benchmarks such as MMLU (Massively Multitask Language Understanding, Hendrycks et al. (2021)) for knowledge and reasoning, HELM (Holistic Evaluation of Language Models, Liang et al. (2023)) for a broad assessment across scenarios and ethical concerns, or BIG-bench (Srivastava et al. 2023) for a comprehensive suite of challenging tasks.

Finally, performance is also assessed based on factors such as energy usage (e.g., kWh or kg of CO₂), fairness (assessing bias and equal performance across demographic groups), and speed.

2.3.5 Limitations

Despite their capabilities, Large Language Models (LLMs) face significant limitations. These challenges often stem from issues with data quality and curation (Longpre, Yauney, et al. 2024), which can lead to hallucinations in which the model prioritizes linguistic fluency over factual grounding. Furthermore, LLMs frequently reflect and amplify societal biases inherent in their training corpora, potentially generating harmful content or reinforcing abusive stereotypes (Hofmann et al. 2024). Safety also remains a primary concern (Bickmore et al. 2018),

particularly because models may inadvertently disclose private information from the training data, a problem compounded by users over-disclosing personal information. Beyond operational risks, the reliance on massive datasets introduces complex legal and ethical dilemmas surrounding copyright (Henderson et al. 2023) and consent (Longpre, Mahari, et al. 2024).

2.4 Transformer

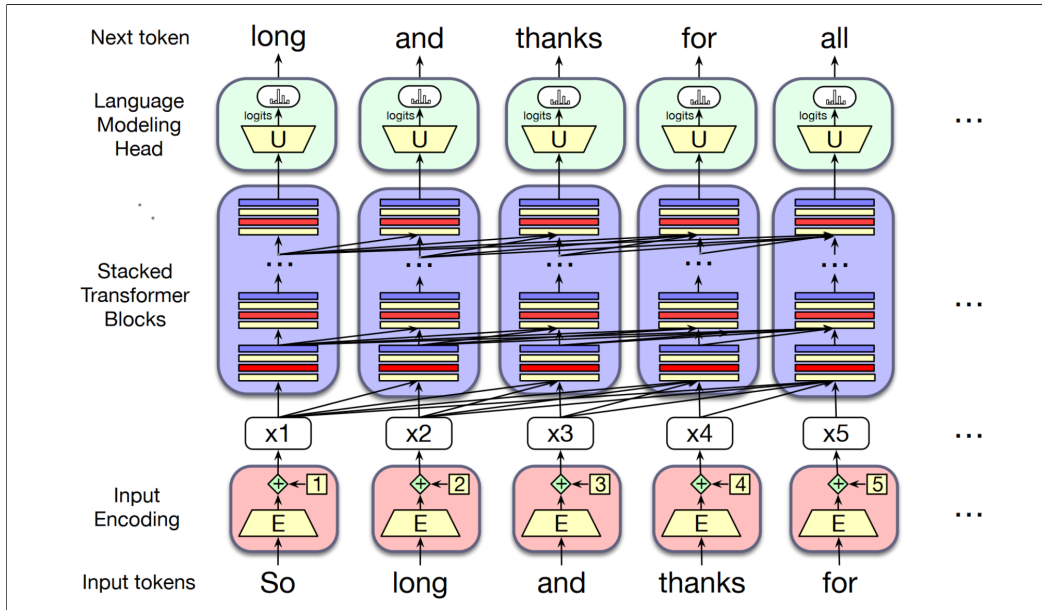


Figure 2.2: The (unidirectional/causal) transformer architecture. It has three components: Each input token gets encoded by the *input encoding*, passed through stacked *transformer blocks*, and a *language modeling head* predicts the next token. Image from Jurafsky and Martin (2025).

The transformer architecture (Vaswani et al. 2017) is the standard architecture for building large language models (Section 2.3) and has revolutionized the field of natural language processing. Its major innovation is self-attention (specifically, multi-head attention, Section 2.4.1). This mechanism builds rich contextual representations of a token by attending to and integrating information from surrounding tokens across vast sequence spans. The architecture is composed of three main components that operate on input sequences (see Figure 2.2):

- **Transformer Blocks:** Transformer blocks (Section 2.4.2) are the core computational units, typically stacked from 12 to 96 or more layers high. Each block processes an input vector using a *multi-head attention* layer and a *feedforward network*, and includes *layer normalization* steps.
- **Input Encoding:** The input encoding (Section 2.4.3) is responsible for translating raw input tokens into numerical, contextual vectors. It combines a token embedding (representing the word) with a positional encoding (representing the word's location in the sequence).

- **Head:** The head (Section 2.4.4) is an additional neural network added to the top of the final transformer block. While various task-specific heads can be added (e.g., for classification), for a causal language model, the *language modeling head* is used. This head takes the final contextualized vector and predicts a probability distribution over the next word in the vocabulary.

2.4.1 Attention

To overcome the inherent limitation of static embeddings (e.g., word2vec, Section 2.1.4), which assign a fixed vector to a word regardless of context, failing to capture context-dependent meaning (polysemy) or long-range dependencies (e.g., pronoun resolution), the transformer dynamically generates contextual embeddings. This contextual embedding is generated by processing input tokens layer by layer, integrating information from surrounding tokens. The operation that enables this is *attention*, a mechanism that selectively weights the influence of other tokens in the context to compute a richer representation of the current token. This helps the model learn how tokens relate to one another across large spans.

Self-attention is executed through three learned projection matrices (W^Q , W^K , W^V), which assign distinct roles to each input vector x_i :

- **Query (q_i):** The current element acts as a *query* (q_i), seeking relevance in the context. It is compared to the prior element. $q_i = x_i W^Q$
- **Key (k_j):** Prior elements act as a *key* (k_j) to signal their content to the *query*. The prior element is compared to the current query to determine similarity. $k_j = x_j W^K$
- **Value (v_j):** The content to be aggregated is contained in the *value* (v_j) vectors. It gets weighted and summed up to compute the output. $v_j = x_j W^V$

First, the similarity score between the current token x_i and all prior tokens x_j ($j \leq i$) is calculated via the scaled dot product of the *query* and *key* vectors:

$$\text{score}(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d}} \quad \forall j \leq i$$

Since a dot product can be an arbitrarily large value, this score is scaled by the dimensionality of the *query* and *key* vectors $\sqrt{d}$ to prevent numerical instability during training. Next, these scores are normalized using a softmax function to produce the attention weights $\alpha_{i,j}$, a pseudo-probability distribution over the context.

$$\alpha_{i,j} = \text{softmax}(\text{score}(x_i, x_j)) \quad \forall j \leq i$$

Finally, the output of a single attention head, head_i , is the weighted sum of all *value* vectors in the context:

$$\text{head}_i = \sum_{j \leq i} \alpha_{i,j} v_j$$

Multi-Head Attention

A single attention head may only focus on one type of relationship. *Multi-head attention* addresses this by using multiple parallel attention heads, each with its own independent set of W^Q , W^K , W^V matrices. This allows the model to capture different linguistic relationships simultaneously (e.g., one head for semantic similarity, another for structural dependency). The outputs of all heads are calculated independently, concatenated, and subsequently projected back down to the model dimensionality d using a final weight matrix W^O

$$a_i = (\text{head}_1 \oplus \text{head}_2 \oplus \dots \oplus \text{head}_A) W^O$$

where $\oplus$ denotes the concatenation operator. Usually, the dimensionality of each head (d_v) is set such that the product of the number of heads (A) and the head dimension equals the model dimension (d). Under this configuration, the concatenated output of the attention heads matches the model dimension d . Consequently, the final output projection matrix W^O becomes a square $d \times d$ matrix.

2.4.2 The Transformer Block

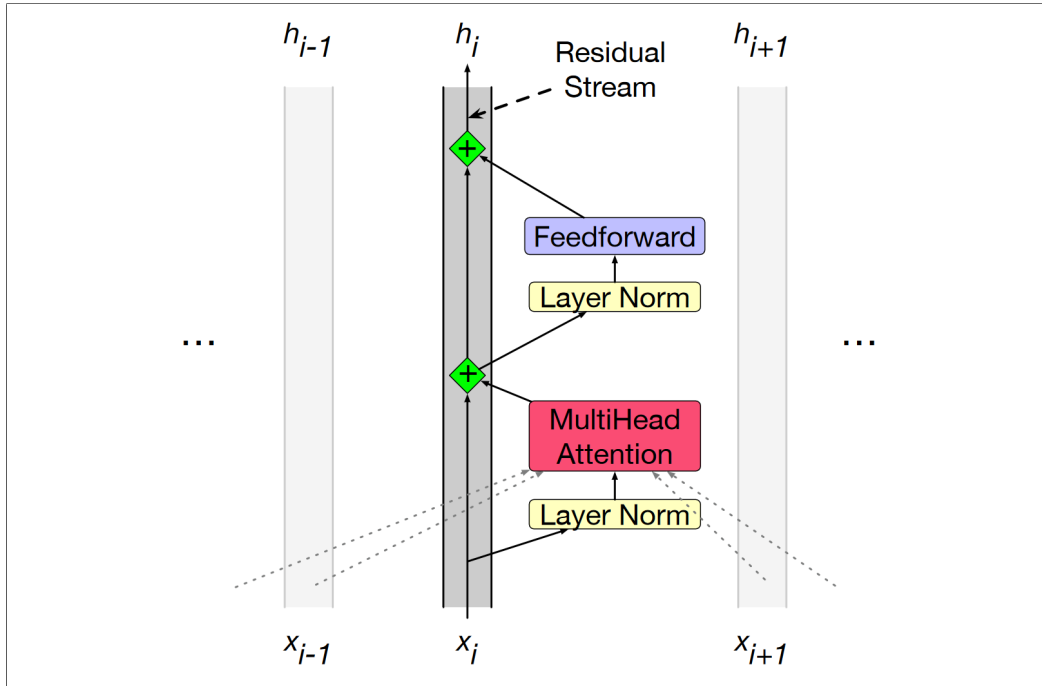


Figure 2.3: The architecture of a transformer block. It shows the residual stream, where results from the *multi-head attention* and *feedforward network* are continuously integrated. Image from Jurafsky and Martin (2025).

The *transformer block* is the core of the transformer architecture, processing an input vector and producing an output vector of the same size, enabling easy stacking. It is best understood as a residual stream (see Figure 2.3) in which the initial input vector is continuously passed forward, with the output of each sub-layer added back into the stream via residual connections. Each block contains three components:

- **Layer Normalization:** Its role is to stabilize and improve training by ensuring the values remain in a range that facilitates gradient-based training. In the prenorm architecture (Xiong et al. 2020), it is applied before the main layers to standardize vector elements. For a given token vector $x \in \mathbb{R}^d$, the mean μ and variance σ^2 are calculated across its d elements:

$$\mu = \frac{1}{d} \sum_{i=1}^d x_i, \quad \sigma^2 = \frac{1}{d} \sum_{i=1}^d (x_i - \mu)^2$$

The normalization is then applied per element. For each element x_i , the normalized value is scaled and shifted using the learnable parameters γ and β

$$\text{LayerNorm}(x_i) = \gamma \left(\frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta$$

where ϵ is a small constant added for numerical stability.

- **Multi-Head Attention Layer:** Computes the contextualized representation of a_i (Section 2.4.1).
- **Feedforward Network:** A two-layer, fully-connected neural network. Its weights are shared across token positions within the sequence but are different from layer to layer in the stack.

The processing flow of the prenorm architecture is as follows: The initial input vector x_i is passed through layer normalization, followed by the multi-head attention layer. The attention output is then added back to the residual stream. This combined vector is passed through another layer normalization, then through the feedforward network, and its output is added back into the residual stream again to produce the final output h_i . The prenorm architecture was introduced by Xiong et al. (2020). It maintains gradient stability and allows for faster convergence, making it the de facto standard in modern transformer architectures. In the original architecture (postnorm, Vaswani et al. (2017)), layer normalization was applied after the sublayer (attention or feedforward network).

2.4.3 Input Encoding

Input processing begins with token embedding. The embedding matrix E is of shape $|V| \times d$. Here, the base d -dimensional vectors for all words of the vocabulary V are stored.

An input string is tokenized, and the corresponding embeddings are looked up. This embedding lookup is achieved by multiplying the input's one-hot vector representation by the embedding matrix. These one-hot vector representations of the input token sequence are combined into a matrix to retrieve all embeddings at once for further parallel computation.

However, the self-attention mechanism does not know the order of the tokens. The position must be explicitly encoded. This is done by combining the token embedding (E_i) with a positional embedding (P_i) at each position i : $X_i = E_i + P_i$. These positional embeddings can be learned (absolute position) or derived from predefined, fixed functions (e.g., sinusoidal functions) to aid generalization to

long sequences. The combination results in the final input matrix X of shape $N \times d$ (where N is the sequence length), containing the composite word + position embeddings for all tokens in the context window.

These embeddings dynamically change as they pass up the residual stream, incorporating more context at each layer.

2.4.4 Language Modeling Head

Any given transformer can be specialized via a head (e.g., a classification head for sentiment analysis). For a causal (left-to-right) language model that processes a sequence of length N , the *language modeling head* maps the output embedding of the final token h_N^L at the final layer L to a "probability" distribution over the vocabulary. This process involves a linear layer U , often implemented by weight tying the matrix to the transpose of the input embedding matrix ($U = E^T$). Weight tying means using the same weight matrix for two different components in the model (here, the input embedding and the output projection), reducing the total number of parameters and improving efficiency. This transpose is often called the unembedding layer. The unembedding step generates the logits u (real-valued scores), which are normalized by a softmax function to generate the final distribution y :

$$u = h_N^L E^T$$

$$y = \text{softmax}(u)$$

2.4.5 Training

Pre-training

The transformer (as presented in this section with a causal language modeling head) is pre-trained by minimizing cross-entropy loss (negative log-likelihood) on the next-token prediction task (Section 2.3.3). In a self-supervised fashion, the correct next word in the training sequence is used to compute the loss. This loss is averaged over all tokens in the sequence and minimized via gradient descent.

A key benefit of the transformer architecture is that the entire input sequence can be processed in parallel (Section 2.4.6) via the attention mechanism, unlike previous sequential models, thereby improving computational efficiency during training. This way, the transformer architecture enables modern language models to achieve extremely large parameter sizes.

Parameter-efficient Fine-Tuning

The goal of fine-tuning is to adapt a pre-trained model to a new domain or task by continuing training on small, relevant datasets. Due to the massive scale of LLMs, full fine-tuning is prohibitively expensive in terms of compute, time, and memory. The high computational cost of full fine-tuning led to the development of parameter-efficient fine-tuning (PEFT) methods that update only a small subset of the total parameters.

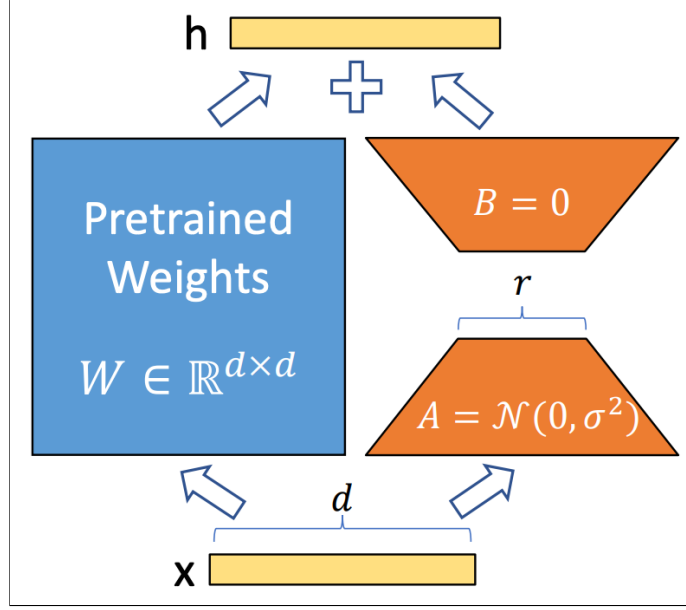


Figure 2.4: LoRA adapter. Only matrices A and B are trained. Image from Hu et al. (2022).

Low-Rank Adaptation (LoRA) (Hu et al. 2022) is a popular and effective PEFT technique used in this dissertation (Section 5.4). LoRA (see Figure 2.4) operates by freezing the original large, pre-trained weight matrices (W) and approximating the required weight update (ΔW) using a pair of small, low-rank decomposition matrices, A and B ($\Delta W \approx AB$). LoRA is typically applied to matrices in attention computation, but many variants of LoRA exist that apply the technique to other weight layers. LoRA offers three key advantages:

1. It significantly reduces hardware requirements (compute and memory) for fine-tuning.
2. It does not add time during inference because the small AB matrices can be merged back into the frozen base matrix W .
3. It allows LoRA adapters to be easily changed. One can build separate for different domains and swap them in and out without affecting the massive base model.

2.4.6 Computational Efficiency

Since the calculation for each token is independent, the transformer architecture allows parallel computing. The inputs for an entire sequence of N tokens are packed into a matrix X ($N \times d$), allowing all key, query, and value matrices (W^Q , W^K , W^V) and attention scores to be computed simultaneously via efficient matrix multiplication. However, for autoregressive (left-to-right) LMs, the model must not be allowed to see future tokens. Hence, a causal mask is applied to the $N \times N$ attention score matrix. The upper-triangular portion (scores relating a query to future keys) is set to $-\infty$. This ensures the softmax output for future tokens is zero, preventing the model from cheating by seeing the future.

During inference (text generation), the *KV cache* is a technique to achieve high processing speed. It prevents redundant computation by storing the *key* and *value* vectors for previously generated tokens in memory, ensuring that only the newly appended token is processed at each decoding step. While effective at reducing latency, the KV cache introduces significant memory overhead, particularly as sequence lengths increase. Recent advancements, such as PagedAttention (Kwon et al. 2023), have addressed these bottlenecks by implementing dynamic memory allocation strategies similar to virtual memory management in traditional operating systems. PagedAttention partitions the KV cache into non-contiguous memory blocks, significantly reducing memory waste and enabling the efficient handling of longer sequences and larger batch sizes. This technique is integrated into the vLLM² library used in DATS to accelerate LLM inference speed and optimize hardware utilization. This is especially important for interactive features of DATS that require responsiveness, such as *Perspectives* (Section 5.4), the *LLM Assistant* (Sections 5.5 and 6.6), and its extension, the three-stage *Annotation Assistant* (Section 6.7).

2.5 Masked Language Models

The bidirectional transformer encoder, most famously implemented in the BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al. 2019), introduces a pretraining paradigm that differs from the next-token prediction task used in causal models. The unidirectional, left-to-right constraint of decoder LLMs is a limitation for many language understanding tasks, such as sequence classification (Section 2.5.4) or sequence labeling (Section 2.5.4), where access to the future context is highly beneficial.

This pretraining paradigm is named after its main objective: *masked language modeling* (MLM). MLM (Section 2.5.2) involves corrupting an input sequence and training the model to predict the missing word from both left and right context simultaneously, thereby enabling bidirectionality. A bidirectional encoder transforms input embeddings ($x_1 \dots x_N$) into output embeddings ($h_1 \dots h_N$) using attention (Section 2.4.1). These outputs are contextualized representations of each input token, and the ability to use the full context (both sides) makes them particularly useful for applications that require a decision based on the token in context.

These models produce contextual embeddings, in which each word instance is represented by a different vector based on its context. This contrasts with static models (e.g., word2vec, Section 2.1.4), which compute a single fixed vector for each unique word. Although causal LMs also generate contextual embeddings, the representations learned via the bidirectional MLM objective work especially well for transfer learning. Consequently, bidirectional encoders are often called encoder-only models because they produce an encoding of the input. They are generally not used for generation; their primary focus is computing representations for downstream tasks (Section 2.5.4).

Transfer learning (Pan and Yang 2010) has emerged as a cornerstone of modern NLP, enabling models to leverage knowledge acquired from a data-rich

2. <https://vllm.ai>

source domain to improve performance on a specific target task. These learned representations enable the transfer learning approach of pretraining and fine-tuning: During the pretraining phase, a language model learns from massive amounts of text and instantiates rich, general, and robust representations of word meaning. These high-quality representations enable the model to easily learn the specific requirements of a downstream task via fine-tuning.

2.5.1 Architecture

Bidirectional encoders are based on the transformer block (consisting of attention, feedforward network, and layer normalization, Section 2.4.2). The main architectural difference from decoder LMs is the use of non-causal attention. Instead, when processing a token i , the model can attend to all tokens in the sequence, including those that follow ($i + 1, \dots, N$). This is implemented by removing the causal masking from the multi-head attention mechanism, which was necessary in causal LMs to prevent looking ahead (Section 2.4.6).

MLMs tend to be much smaller than causal LLMs. Example pretrained models include:

1. BERT³: Features a vocabulary size of $\approx 30,000$ tokens, input context window $N = 512$, model dimensionality $d = 768$, $L = 12$ transformer blocks, and $\approx 100\text{M}$ parameters.
2. XLM-RoBERTa⁴: Features a vocabulary size of $\approx 250,000$ tokens, input context window $N = 512$, model dimensionality $d = 1024$, $L = 24$ transformer blocks, and $\approx 550\text{M}$ parameters.
3. ModernBERT⁵: Features a vocabulary size of $\approx 50,000$ tokens, input context window $N = 8192$, model dimensionality $d = 768$, $L = 22$ transformer blocks, and $\approx 150\text{M}$ parameters.

2.5.2 Pretraining

Masked Language Modeling

Eliminating the causal mask makes the next-token prediction task trivial. Hence, a different kind of language modeling task is necessary to train bidirectional encoders. The model is trained on a fill-in-the-blank task, also called a cloze task (Taylor 1953). This task is one of many methods that corrupt the input and require the model to recover the original, a general approach known as denoising.

The core training method for BERT and all of its descendants is the MLM task. The MLM procedure presented in the original BERT paper manipulates 15% of tokens: 80% of the time, the token is replaced with the special token `[MASK]`; 10% of the time, it is replaced with a randomly sampled token; and 10% of the time, the token is left unchanged. For example, given the corrupted sentence "The `[MASK]` of Hamburg is so beautifully `[MASK]`", the model must predict the missing tokens

3. <https://huggingface.co/google-bert/bert-base-uncased>

4. <https://huggingface.co/FacebookAI/xlm-roberta-large>

5. <https://huggingface.co/answerdotai/ModernBERT-base>

(e.g., 'sky' and 'gray'). The use of the `[MASK]` token in pretraining creates a mismatch with fine-tuning (where no `[MASK]` tokens are present). Therefore, this mixed manipulation strategy is used to mitigate this mismatch and force the model to learn context-based prediction for all tokens, rather than relying solely on the presence of the `[MASK]` token.

Training minimizes cross-entropy loss (Section 2.3.3) only for the manipulated tokens (M), computed using the language modeling head (Section 2.4.4). The loss is the average negative log probability the model assigns to the correct word for each masked item:

$$L_{MLM} = -\frac{1}{|M|} \sum_{i \in M} \log P(x_i | h_i^L)$$

Here, $P(x_i | h_i^L)$ denotes the probability of the correct word x_i given the corresponding contextual embedding h_i^L of the last layer L . Only 15% of input tokens contribute to the loss, making this process relatively inefficient compared to next-token prediction, where every token contributes.

Next Sentence Prediction

To capture knowledge about sentence relationships (e.g., entailment or discourse coherence), BERT introduced a second objective: *next sentence prediction*. The model is presented with pairs of sentences and must predict if the pair consists of two adjacent sentences (50% of the time) or two randomly selected, unrelated sentences (50% of the time). Input sentences are formatted with the special token `[CLS]` prepended and separated by `[SEP]`. A dedicated next sentence prediction head (a linear classifier) uses the output vector h_{CLS}^L associated with the `[CLS]` token of the last layer L to make a two-class prediction. Again, cross-entropy loss is the training objective.

2.5.3 Contextual Embeddings

A masked language model generates contextual embeddings for each input token, representing aspects of the token's meaning in context. The output vector h_i^L from the final layer L for an input token x_i can be used as this representation. However, it is common practice to compute the contextual embedding by averaging the outputs of multiple last layers of the model (e.g., averaging the last four layers, Vaswani et al. (2017)).

These vectors represent the meaning of a word instance in a specific context, as opposed to static embeddings (e.g., word2vec, Section 2.1.4), which compute a single fixed vector for each unique word. A challenging aspect when modeling words is their ambiguity (polysemy), meaning the same word can have multiple meanings (e.g., "*bank*"). Contextual embeddings reflect this ambiguity by having distinct representations that cluster in different regions of the vector space, corresponding to other senses.

The similarity between two contextual embeddings is measured using cosine similarity (Section 2.1.5). However, embeddings are inherently similar, with vectors often pointing in the same direction, leading to high cosine scores even when the true semantic relation is absent. There are approaches to mitigate

this issue, such as standardizing all embeddings within a given corpus (Timkey and van Schijndel 2021).

2.5.4 Fine-Tuning for Downstream Tasks

To adopt a pre-trained LM for a specific task, it is fine-tuned by adding a task-specific neural network (a special head) on top of the model's output. Typically, the model's parameters are frozen, and only the head's parameters are learned using task-specific labeled data. Though sometimes minimal adjustments are made to the final transformer layers, or the whole model is trained end-to-end.

Sequence Classification

Sequence classification, also called text classification, is the task of assigning a single label to an entire sequence of text. Examples include sentiment analysis (e.g., classifying a review as positive/negative, Pang et al. (2002)) and spam detection (Sahami et al. 1998).

To enable classification, the entire sequence must be represented as a single vector. This can be achieved by taking the sum or mean of the last layer's token embeddings. For BERT-style models, however, this is typically achieved by taking the output vector associated with the special prepended $[CLS]$ token (h_{CLS}^L). This vector is passed to a lightweight classification head (W_C , typically a single linear layer), followed by a softmax function to generate label predictions:

$$y = \text{softmax}(h_{CLS}^L W_C)$$

Sequence-Pair Classification

Sequence-Pair classification involves classifying the relationship between two input sequences. Examples include natural language inference (Dagan et al. 2006; Bowman et al. 2015), where the objective is to determine whether a "premise" sentence entails, contradicts, or is neutral toward a "hypothesis" sentence, and paraphrase detection, which identifies whether two sentences convey the same semantic meaning.

The two sequences are formatted with the $[SEP]$ token placed between them. Classification is performed identically to *sequence classification*: by using the $[CLS]$ token vector as input to a classifier head. This architecture is often called a *cross-encoder* model because the attention mechanism computes relationships across the two sequences at every layer.

Bi-Encoders: For large-scale applications such as information retrieval, semantic search, or clustering, where millions of sentence pairs need to be compared efficiently, the *cross-encoder* setup is impractical. This is due to its computational complexity: it must process all pairs of N sequences to generate a single similarity score, resulting in N^2 comparisons, which is intractable.

The *bi-encoder* architecture (Reimers and Gurevych 2019) solves this efficiency problem by converting each sentence into an independent, fixed-size vector (a sentence embedding). Two separate encoders (typically with shared weights)

process sentence A and sentence B independently, generating two distinct embeddings (u and v). The similarity between the sentences is then calculated using the cosine similarity (Section 2.1.5):

$$\text{similarity}(A, B) = \text{cosine}(u, v)$$

Bi-encoders are typically trained using a contrastive learning objective (like the triplet loss or softmax loss with in-batch negatives), which forces the model to maximize the cosine similarity between related sentence pairs (positives) and minimize it for unrelated pairs (negatives). The primary advantage is that the embedding for each sentence can be precomputed and indexed (e.g., in a vector database), enabling high-speed retrieval.

The triplet loss (Schroff et al. 2015) is a contrastive objective designed to refine the embedding space by processing three distinct inputs simultaneously: an anchor sequence (a), a positive sequence (p) that shares the same label or semantic meaning as the anchor, and a negative sequence (n) that differs. Let $u_a, u_p, u_n \in \mathbb{R}^d$ denote the fixed-size embeddings generated by the bi-encoder for the anchor, positive, and negative inputs, respectively. The objective function seeks to minimize the distance between the anchor and the positive embedding while ensuring that the distance to the negative embedding is greater by at least a predefined margin α :

$$\mathcal{L}(a, p, n) = \max(0, \|u_a - u_p\|^2 - \|u_a - u_n\|^2 + \alpha)$$

By enforcing this margin, the model learns a representation in which semantically related instances are clustered together and dissimilar instances are pushed apart by at least distance α . In practice, the effectiveness of this approach relies heavily on triplet mining strategies, which select "hard" negatives (instances that the model currently maps close to the anchor) to ensure a meaningful gradient signal during training.

Instruction-Finetuned Embeddings: A significant limitation of traditional bi-encoders is that they produce task-agnostic embeddings: the same vector is generated for a given text regardless of whether the goal is clustering, retrieval, or classification. Instruction-finetuned embedding models, first described in the INSTRUCTOR framework by Su et al. (2023), address this by allowing users to provide natural-language instructions that guide the embedding process. Such INSTRUCTOR-style embedding models are used in this dissertation to implement the Perspectives feature (Section 5.4).

The INSTRUCTOR model is trained on a massive variety of tasks and domains, where each input consists of a task instruction (e.g., "Represent the sentence for clustering:") paired with the text itself. This allows the model to generate task-specific representations. Functionally, the instruction acts as a context that modulates the attention mechanism, ensuring the resulting vector captures the semantic features most relevant to the specified objective. This architecture represents a significant advancement over standard bi-encoders, as a single model can adapt its semantic space to suit the requirements of diverse downstream applications without the need for task-specific fine-tuning.

Sequence Labeling

Sequence labeling involves assigning a label chosen from a fixed set to each token in the sequence. It is traditionally defined as a supervised learning problem. This means a machine learning algorithm is trained on a dataset that has been pre-labeled with known categories (e.g., person, organization, location). In sequence labeling, a prediction is made for every token in a sequence, determining whether it belongs to a specific category or the outside class. Hence, a model needs to assign a corresponding label to each element in an input sequence, ensuring that contiguous parts of the text (spans) are correctly classified.

The most popular task is named entity recognition (NER) (e.g., tagging person, location, organization). NER is part of many downstream NLP pipelines, such as linking information to structured knowledge sources (such as Wikipedia) and measuring sentiment or attitudes toward a particular entity.

The NER task is transformed into a token-by-token classification task using BIO tagging (begin, inside, outside), which was introduced by Ramshaw and Marcus (1995). This allows NER to be treated as a sequence labeling problem. Many variations of this schema exist, for example, IO tagging. This schema is beneficial in that it requires the model to learn fewer classes, but it cannot distinguish between adjacent entities of the same type.

The output vector h_i^L corresponding to each input token i is passed to a dedicated classifier head (W_K) that produces a softmax distribution over the set of possible tags:

$$y_i = \text{softmax}(h_i^L W_K)$$

Here, y_i denotes a vector that assigns a score to each possible tag. The final tag sequence is typically generated greedily by taking the most probable tag for each token:

$$t_i = \text{argmax}(y_i)$$

While greedy decoding is common, more sophisticated approaches exist. For example, a *conditional random field* (Lafferty et al. 2001) layer can be added on top of the classifier head to learn tag-level transitions and improve sequence-prediction coherence. This architecture is explored to implement the three-stage *Annotation Assistant* for sequential sentence classification in Section 6.7.

2.6 LLMs used in the Dissertation

The specific open-weight large language models utilized in this work (especially in the evaluation conducted for the *LLM Assistant* and *Annotation Assistant* in Sections 5.5, 6.6, and 6.7) include Llama (Meta), Mistral (Mistral AI), and Gemma (Google), each representing state-of-the-art developments in the field.

2.6.1 Llama 3.1

Llama 3.1 (Touvron et al. 2023; Grattafiori et al. 2024) is an openly accessible language model developed by Meta, released in July 2024. The model is available

in three parameter configurations: 8 Billion (8B), 70 Billion (70B), and 405 Billion (405B) parameter count. The model family was pretrained on a corpus exceeding 15 trillion tokens, sourced from a diverse, publicly available online dataset optimized for multilingual text and code. The context window supports up to 128,000 (128K) tokens across all variants. Llama 3.1 is multilingual, explicitly supporting eight languages: English, French, German, Hindi, Italian, Portuguese, Spanish, and Thai. The model operates as a text-only system, accepting and generating multilingual text. It is released under the Llama 3.1 community license, which permits both commercial and research use and explicitly permits using the model’s outputs to improve other LLMs.

2.6.2 Mistral NeMo

Mistral NeMo (Jiang et al. 2023; MA Team 2024) is a 12 Billion (12B) parameter model resulting from a technical collaboration between Mistral AI and NVIDIA. The model is deployed under the Apache 2.0 License, which allows commercial adoption and customization. Mistral NeMo uses a context window of 128,000 (128K) tokens and is trained on a proprietary dataset that includes a large proportion of multilingual and code-centric data. The model employs the Tekken tokenizer, based on Tiktoken, which improves compression of source code and other non-English languages. It supports various languages, including English, French, German, Spanish, Italian, Portuguese, Chinese, Japanese, Korean, Arabic, and Hindi. Mistral NeMo is a text-only, decoder-only transformer architecture.

2.6.3 Gemma 2

Gemma 2 (G Team 2024) is an open-weight language model from Google DeepMind. It is offered in three parameter sizes: 2 Billion (2B), 9 Billion (9B), and 27 Billion (27B). The model was trained on diverse public datasets covering web content, books, and research papers, with smaller variants using knowledge distillation. The training data is primarily focused on English-language content. The model utilizes an 8,000 (8K) token context window. Gemma 2 is provided as a text-only model under specific Google terms that allow for research and commercial use.

2.6.4 Gemma 3

Gemma 3 (G Team 2025) is the third generation of Google’s lightweight open models, built from the same technology as their Gemini models. It is released in multiple sizes: 270 Million (270M), 1 Billion (1B), 4 Billion (4B), 12 Billion (12B), and 27 Billion (27B) parameters. The Gemma 3 architecture supports multimodality, enabling the larger 4B, 12B, and 27B variants to process both image and text inputs, while the 270M and 1B models remain text-only. The model was trained using distillation and post-training techniques, leveraging a multimodal dataset that includes text, images, and videos for alignment. The context window is 128,000 (128K) tokens for the larger models (4B, 12B, 27B), with the smallest variants (270M, 1B) supporting 32,000 (32K) tokens. Gemma 3 supports over 140 languages. Licensing follows specific Google terms.

This chapter established the technical foundation of modern NLP, most importantly, the power of large language models and contextualized embeddings to represent linguistic meaning. In the context of this dissertation, these technologies aid qualitative analysis of social reality and support discourse analysis. In this dissertation, these technologies are used to support the qualitative analysis of social reality within the discourse analysis framework. Hence, we turn to the second pillar of this interdisciplinary work in the next chapter. It establishes the theoretical and procedural foundations of the empirical social sciences, with a particular focus on qualitative discourse analysis.

3

Background – Discourse Analysis

Contents

3.1	Empirical Social Sciences	46
3.1.1	Definition	46
3.1.2	Research Approaches	47
3.1.3	Research Process	48
3.1.4	Quality Criteria	49
3.2	Grounded Theory	50
3.2.1	Definition	50
3.2.2	Methods	51
3.3	Hermeneutics	54
3.3.1	Definition	54
3.3.2	The Hermeneutic Attitude	55
3.3.3	The Interpretation Process	56
3.4	Discourse Analysis	57
3.4.1	The Discourse	57
3.4.2	Discourse Analysis	59
3.4.3	The Meaning of Power	61
3.4.4	Variants of Discourse Analysis	62
3.5	Procedure of a Discourse Analysis	62
3.5.1	Phase 1: Preliminary Work	62
3.5.2	Phase 2: Corpus Formation	62
3.5.3	Phase 3: Reconstructing the Discursive Formation	63
3.5.4	Phase 4: Fine-grained Analysis of Function & Power	64
3.5.5	Phase 5: Social Analysis	65
3.6	Sociology of Knowledge Approach to Discourse	65
3.6.1	Hermeneutics & Reflexivity	65
3.6.2	Grounded Theory as an Analytical Tool	66
3.6.3	Analytical Focus	66

This chapter provides the methodological and theoretical framework from the empirical social sciences that guides the development of the Discourse Analysis Tool Suite (DATS). As this dissertation aims to augment qualitative research with computational methods, it is important to understand the standards and iterative logic of the social scientific traditions being supported.

This chapter is largely based on the comprehensive book on methods of empirical social research by Baur and Blasius (2022a), with each section of this chapter drawing on relevant chapters of that book. We begin with an overview of empirical social research (Section 3.1), based on Baur and Blasius (2022b), to define the paradigms of qualitative and quantitative inquiry. The chapter then details two foundational methodologies: Grounded Theory (Section 3.2), following Strübing (2022), and Hermeneutics (Section 3.3), based on Kurt and Herbrink (2022). The core of the chapter (Sections 3.4 and 3.5) explores the principles and procedures of Discourse Analysis, drawing on Traue et al. (2022) and Keller (2011a). Finally, Section 3.6 presents the Sociology of Knowledge Approach to Discourse (SKAD) by Keller (2011b). This approach, which synthesizes Hermeneutics and Grounded Theory, serves as the primary guide for both this dissertation and the Discourse Analysis Tool Suite, dictating the requirements for the features presented in subsequent chapters.

3.1 Empirical Social Sciences

3.1.1 Definition

Empirical Social Research (ESR) forms the methodological backbone of the social sciences. It is the systematic collection and evaluation of data on social phenomena, structures, and human behavior using standardized procedures (e.g., observation, surveys, experiments). The social sciences thus understand themselves as sciences of reality. This means that research is grounded in empiricism (e.g., observable, measurable realities) rather than in purely theoretical or philosophical considerations. ESR provides the rules by which data are collected, linked to theories, and subsequently evaluated, making theoretical statements and predictions accessible to empirical verification. ESR is a component of numerous disciplines, including sociology, political science, education science, psychology, communication sciences, and economics. However, its relevance extends to interdisciplinary fields, such as human-computer interaction, where its methods are used to determine user needs (e.g., surveys/interviews) and to evaluate system effectiveness (e.g., experiments) (Baur and Blasius 2022b, p. 1).

The central goal of ESR is to collect and explain social reality to answer concrete research questions. This includes describing, explaining, and prognosticating social realities, verifying existing theories, and developing new knowledge about society (Baur and Blasius 2022b, p. 3).

It is assumed that theory and empiricism are inseparably linked. Researchers must link data and theories (Baur and Blasius 2022b, p. 3), whereby three logical forms of conclusion are distinguished (Reichert 2022, pp. 80–83):

- **Deduction:** Derivation of specific hypotheses from a general theory and their empirical testing (often dominant in quantitative research).
- **Induction:** Generalization of specific empirical cases into a general theory. Collection of empirical data to develop patterns and connections, leading to the generation of new, more general theories (often dominant in qualitative research).
- **Abduction:** A creative "mental leap" or "flash" triggered by surprising data that cannot be explained by existing rules, leading to the invention of a new theory to make the observation comprehensible. Unlike the other two, it is the only logical operation that brings a truly new idea into the world.

The logic of research is ultimately understood as a three-stage iterative process. It begins with abduction, where a "fitting" explanation is found for surprising observations to generate a hypothesis. In the second stage, deduction is used to derive specific predictions from this hypothesis, which are then systematically tested in the third stage through induction. If the empirical materials do not verify the assumptions, this cycle repeats until a coherent connection between theory and data is established (Reichertz 2022, p. 83).

3.1.2 Research Approaches

ESR employs a range of methods that can be divided into two fundamental approaches, which, in turn, significantly shape the research process. The methods of ESR can be broadly divided into quantitative (number-based, aiming for generalization) and qualitative (content-based, aiming for in-depth understanding) approaches. These two paradigms differ in their logic, their research goal, and the role of the researcher.

Quantitative: The quantitative paradigm (Baur and Blasius 2022b, pp. 8–9) primarily focuses on hypothesis testing, measurement, and generalizing results. The research process follows a successive or sequential logic: theoretical reflection comes first, followed by individual steps, such as instrument development, data collection, representative sampling, preparation, and evaluation, processed sequentially. Methodologically, the aim is to exclude the researchers' subjectivity as far as possible to ensure objectivity. Each work step is optimized for its own purpose and should be as independent of the researcher as possible. The data basis for analysis consists of numbers, measurements, and standardized data derived from the collected, representative data.

Qualitative: The qualitative paradigm (Baur and Blasius 2022b, pp. 9–10), however, aims for in-depth understanding and the generation of new theories. It proceeds from the realization that complete exclusion of the researchers' subjectivity is not possible and that social objects can change permanently. This paradigm, therefore, emphasizes that the researchers' subjectivity is an important research resource. Scientific objectivity here is established not through the independence of the person, but through the reflection of the methodological approach and its

traceable disclosure. The research process is typically designed to be circular or iterative: Data collection, selection, analysis, and theoretical reflection alternate. The methods are flexibly tailored to the requirements of the theory, the research question, and the object. The data basis consists of texts, observations, and unstructured, unstandardized data.

Mixed Methods: The mixed-methods approach (Kelle 2022, 163, pp.167–168) combines quantitative and qualitative methods to leverage their respective strengths and compensate for weaknesses. The assumption is that the two methods are not incompatible opposites but rather complement each other. The combination should therefore be used when it best suits the respective research question and subject area. Various research designs exist here. For example, qualitative interviews can inform the development of items for a subsequent quantitative survey (exploratory design), or quantitative results can be deepened and explained through subsequent qualitative interviews (explanatory design). The data basis is accordingly a combination of both data types.

3.1.3 Research Process

The research process itself is independent of the chosen paradigm. While Baur and Blasius (2022b, pp. 6–7) describe the research process as a series of challenges to be solved based on a specific research logic, these systematic steps can be conceptually grouped into four overarching phases. While in quantitative approaches these phases are processed sequentially, qualitative and mixed-methods designs often proceed through this process in a circular, iterative manner, with the phases of data collection and analysis, in particular, repeated multiple times (Baur and Blasius 2022b, pp. 10–19):

I. Conceptualization & Planning Phase: This phase serves to ground and structure the entire research project. It begins with developing a research question. This is followed by a detailed engagement with the existing literature and the state of research, including the acquisition of reliable information. A key step is the theory connection, in which the understanding of theory-empiricism is clarified (linking of data and theories, inferring theories from data, explaining empirical phenomena), and the research question is situated within existing theories. Based on this theoretical preliminary work and knowledge of the literature, the *research design* is developed. In this step, the entire process is planned, the procedure is defined, and essential decisions regarding data selection, collection, and evaluation are made. The design encompasses selecting a suitable *research approach* (quantitative, qualitative, or mixed-methods) and determining the *data collection instrument* (e.g., a questionnaire, interview guide, or observation scheme). The research design can differ significantly depending on the chosen approach.

II. Empirical Phase: Before actual data collection, ethical and legal questions must be clarified, a consideration that runs throughout the research process. This includes compliance with the rules of good scientific practice and ethics codes, the

protection of research subjects (e.g., from harm and the observance of their rights), the handling of sensitive topics, and questions of data sharing. It is important to note that new data are not always collected, as this is complex and expensive. Researchers increasingly rely on existing datasets for secondary analysis, which is necessary to analyze social processes that unfold over time. An optional, but recommended, step is a pretest to check the instruments or the research design on a small sample for troubleshooting. This step is mainly used in quantitative research. This is followed by sampling, which involves selecting the sample from the population. In quantitative research, achieving a representative sample is the goal, whereas in qualitative research, the focus is on making a thoughtful selection of data to compile the data corpus. In the field phase, the data are finally collected systematically using the developed instruments.

III. Evaluation & Interpretation Phase: Regardless of the thoroughness and care in data collection, this phase almost always begins with data preparation and cleaning, a time-intensive step. This includes, for example, transcribing qualitative interviews or storing quantitative data in a form suitable for statistical evaluation. The actual data analysis follows this: the application of suitable statistical (quantitative, e.g., descriptive statistics, correlation or regression analyses) or interpretive (qualitative, e.g., content analysis, grounded theory (Section 3.2), discourse analysis (Section 3.4)) procedures for evaluating the data. This analysis step represents a significant part of the research work. The phase ends with the interpretation and reconnection to theory, during which the results are interpreted in relation to the research question and compared with the initial theory or the hypotheses.

IV. Concluding Phase: The results are documented and archived to ensure comprehensibility, later re-analyses, and secondary analyses. Finally, the results are presented and published (e.g., in scientific presentations, publications, and dissertations).

3.1.4 Quality Criteria

To meet scientific standards, ESR applies rigorous quality controls. Regardless of the chosen approach, ongoing self-assessment, process management, and quality assurance are standard practices (Baur and Blasius 2022b, p. 10). The quality criteria, however, differ according to the research approach.

Quantitative criteria are based on "classical" metrics: (a) *objectivity*, the independence of the research result from the person of the researcher; (b) *reliability*, the trustworthiness and measurement accuracy of the instrument used; and (c) *validity* of the measurement, ensuring that what is intended to be measured is actually measured (Krebs and Menold 2022).

In qualitative research, the application of these classical criteria is often viewed with skepticism because they are often incompatible with the approach's non-standardized nature. For instance, the traditional understanding of reliability is often unsuitable for qualitative data like narrative interviews. Instead, quality is ensured through specific strategies (Flick 2022):

- **Triangulation:** The use of multiple methods, data sources, researchers, or theories to capture the complexity of the phenomenon and identify contradictions.
- **Member Checking:** Returning results to the research participants to check for plausibility and agreement.
- **Analytic Induction:** Systematically searching for "negative cases" or outliers that contradict a developed hypothesis to refine the theory.
- **Transparency:** Documenting all methodological decisions to ensure the research process is traceable and comprehensible for the reader.

Earlier influential frameworks sought to reformulate classical criteria into a more "naturalistic" set of standards. The most widely recognized of these is the *framework of trustworthiness*¹ proposed by Lincoln and Guba (1985), which includes:

- **Credibility:** Ensuring research results are believable from the participants' perspective.
- **Transferability:** The degree to which results can be transferred to other contexts.
- **Dependability:** The stability of the data and research process over time.
- **Confirmability:** The extent to which results are grounded in data rather than researcher bias.

In particular, the methods of qualitative empirical social research are considered in this thesis. Therefore, the essential analytical strategies of qualitative social research are presented below. Throughout the dissertation, I will repeatedly draw on the following concepts, particularly those from discourse analysis, to motivate my contributions.

3.2 Grounded Theory

3.2.1 Definition

Grounded Theory (GT) is an ESR research strategy that belongs to the qualitative-interpretive methodologies. It was first defined by Strauss and Corbin (1990) and Strauss et al. (1996). The term Grounded Theory refers to both the research process (grounded in the data) and the results, namely, the object-related theories developed in the process.

GT is not to be understood as a rigid methodology, but rather as a flexible framework of methodological proposals for research practice. This also means that it is not bound to specific forms of data acquisition or data materials. Interview studies are used just as much as document analyses, video analyses, or field research (Strübing 2022, pp. 592-594).

1. <http://www.qualres.org/HomeLinc-3684.html>

The central goal of Grounded Theory is the generation of object-related theories that should exhibit practical explanatory potential. It aims to create knowledge that can help solve practical problems in a specific field of action (Strübing 2022, pp. 591,601). The generated theories are thereby always considered unclosed or incomplete.

3.2.2 Methods

The research logic of Grounded Theory (Strübing 2022, pp. 592-594) consists of parallelizing the steps of data collection, data analysis, and theory generation. These three activities are not conceived as steps to be managed sequentially but are operated simultaneously to influence one another productively. Analytical ideas from data analysis influence the development of the theory, which in turn shapes the data collection process (e.g., by using other data types or modifying data collection methods). The analysis thus begins already with the first case. The two central methods of GT are *theoretical sampling* and the *constant comparative method*, which are described next.

Theoretical Sampling

Theoretical sampling (Strübing 2022, p. 596) is the strategic, successive control of data collection by the emerging theory and replaces traditional, predetermined sampling. The goal of theoretical sampling is to shape the selection of cases so that they contribute maximally to the development and refinement of the theoretical concepts.

The sampling is conducted throughout the project and is guided by selection criteria derived from the emerging theory. A prior determination of the sample is thus excluded, as the relevance of the selection criteria remains open at the outset. The criteria for homogeneity (minimal contrasting) and heterogeneity (maximal contrasting) must stem from the ongoing theory-generation process to ensure the relevance of the comparisons (this is explained in more detail in the next section).

The iterative process begins with analyzing a first case and developing initial analytical concepts and categories through coding (Section 3.2.2). The emerging theory subsequently provides the criteria for selecting the next relevant data, after which the material is systematically sought and analyzed. This process is continued until *theoretical saturation* (Section 3.2.2) is reached.

Constant Comparative Method

The constant comparative method (Strübing 2022, pp. 594–597) is a central method for the practical analytical work of GT and is interwoven with theoretical sampling. It is a continued iteration of two alternating comparison modes:

- **Minimal Contrasting (homogeneous cases):** First, comparative materials/-cases are consulted, which are as similar (homogeneous) as possible to the first piece of material in the content dimensions determined as relevant. The goal is to elaborate on the core of a case typology or a theoretical concept by asking: *What remains constant regarding the phenomenon despite all variation?* This leads to the development of stable properties (dimensions)

which form the core of the theoretical concept. This mode is maintained until theoretical saturation is reached.

- **Maximal Contrasting (heterogeneous cases):** Subsequently, deviant (heterogeneous) cases are specifically sought out. The goal is to test the preliminary theoretical conceptualization, e.g., to verify the scope of the concept developed so far. The central question is: *What changes — and what remains stable — when we allow the context of the cases to vary?* The result is information about the scope of the concept itself as well as of valid alternative concepts.

This interplay occurs continuously and at various analytical levels (e.g., comparing interviewees as cases, comparing an individual's definitions of a situation across changing circumstances).

Coding

Data interpretation and analysis in Grounded Theory is achieved through coding. Analytical ideas, concepts, and categories are developed from the data material. Strauss et al. (1996) typically distinguishes three modes of coding (Strübing 2022, pp. 597-602):

Open Coding: Here, thematic access to the material is created, relevant pieces are selected, isolated, and they are analyzed in detail. At the beginning of the analysis or when developing new thematic dimensions, a line-by-line analysis is employed to open the text and explore its dimensions of meaning. Here, the researchers' theoretical sensitivity (e.g., their ability to recognize patterns and meanings in the material) is particularly important. A strict variant of such sequential analysis is applied in Hermeneutics (Section 3.3.3).

Axial Coding: This coding is oriented towards the question of causes, circumstances, and consequences of different variants of a concept. The goal is to create networks of meaning that comprehensively explain the focused category by coding around its *axis*. Here, elements such as conditions, interactions, strategies, consequences, and context are structured to further explain a concept type.

Selective Coding: This coding begins with the decision on the key category that best addresses the research problem. The entire analytical structure (concepts and relationships) is readjusted and integrated into a coherent theoretical design. The key category is pointing the way for continued theoretical sampling and data collection.

Throughout the coding process, new categories are constantly created, further developed, and reflected upon. This often results in texts being coded multiple times throughout a project.

Theoretical Saturation

Theoretical saturation (Strübing 2022, p. 595) serves as a termination criterion for the constant comparative method of theoretical sampling.

A step of theoretical conceptualization is considered saturated when the continued inclusion of further homogeneous cases yields no additional properties. As soon as no new analytical insights are gained and a continuation in the same way promises no further knowledge gain, theoretical saturation is reached, and the comparative method is terminated.

Memo Writing

As detailed by Bryant et al. (2007), memo writing is an essential component of Grounded Theory. A memo is the written record of a researcher's thinking. Memos are simply notes written for each evaluation step and serve as tools for reflection and learning.

Its primary purpose is to document and systematically further develop the theory. Memos record methodological considerations, analytical steps, discovered patterns, and personal thoughts that arise during data analysis. For example, they document how specific key categories gained importance or how interpretations changed.

Memos serve as the basis for the later written presentation of the theory. They are written continuously throughout the analysis. A memo typically includes a date, an identifier (title/code), and a summary of the data to provide context for interpretation. Continuous memo writing is a central aspect of the organization of the interpretative work.

Example

To illustrate the application of Grounded Theory, consider a research project investigating how professional software engineers integrate large language models into their coding workflows.

The researcher starts by observing three senior developers using an AI coding assistant. Through open coding, the researcher breaks down the data line by line, identifying initial codes such as *prompt tinkering*, *blind copy-pasting*, and *verification fatigue*. A memo is written at this stage reflecting on a recurring tension: *"Developers seem to trade deep architectural understanding for immediate syntactical speed"*.

Using theoretical sampling, the researcher decides to interview junior developers next (minimal contrasting) to see whether the *"speed vs. understanding"* tension is consistent across experience levels. After finding that junior developers rely even more heavily on AI, the researcher then seeks out developers working on safety-critical medical software (maximal contrast). Using the constant comparative method, the researcher observes that while web developers prioritize *"velocity"*, medical software engineers adopt a *"verification-first"* mindset. This leads to the refined category of *trust modalities*.

During axial coding, the researcher connects the central category *trust modalities* to its causal conditions (*task complexity*), strategies (*iterative prompting*), and consequences (*technical debt vs. rapid prototyping*).

In the selective coding phase, the researcher identifies the key category: *augmented problem solving*. This core category integrates all previous codes

into a coherent theory explaining how the developer's role shifts from a "*writer of code*" to an "*orchestrator of automated logic*".

The researcher continues to sample diverse cases, e.g., open-source contributors, data scientists, and embedded systems engineers. When these new cases no longer provide new properties for the *augmented problem solving*, theoretical saturation is reached. The final theory is then documented, supported by the transparency provided by the memos written throughout the process.

3.3 Hermeneutics

3.3.1 Definition

Hermeneutics is a methodology and a set of methodical procedures within qualitative empirical social research. It serves the in-depth analysis of empirical individual cases (e.g., interviews, observations, documents) (Kurt and Herbrink 2022, p. 607).

Its goal is reconstructive understanding: the societal meaning of social actions, interactions, and their products must be deduced and reconstructed step by step from the actors' perspective. This involves identifying the specific interpretation routines, action patterns, rituals, and symbols through which individuals provide meaning and structure to their everyday lives. The focus lies on the fundamental problems that actors attempt to solve through their behavior, situated within their biographical, socio-historical, and cultural contexts. Ultimately, it seeks to uncover the latent structures of meaning and the possibilities for agency—exploring how individuals might think, interpret, or act differently within their respective subjective realities (Kurt, 2022) (Kurt and Herbrink 2022, pp. 611–612).

Hermeneutics aims at a *type-forming* understanding of social contexts of meaning, not at the complete understanding of individual persons (Kurt and Herbrink 2022, pp. 611–612). At the end of the analysis, the individual case is reconstructed as the particular solution to a general problem, to infer from the individual case to the general (inductive reasoning). The developed type is a generalized solution structure or a social pattern that explains how actors solve a fundamental societal problem (e.g., a communication problem) in a specific way.

For example, a common societal communication problem is organizational ambiguity (unclear roles, shifting expectations). An actor might be studied who, when faced with this ambiguity, systematically defaults to seeking written confirmation for every verbal instruction. Through the hermeneutic analysis of several such cases, the researcher would infer the generalized solution structure, or the social type: the "*pragmatic clarity seeker*". This type explains a generalized pattern in which the fundamental societal problem of uncertainty is solved by consistently externalizing and documenting communication to minimize personal risk. Importantly, the type "*pragmatic clarity seeker*" is not a description of a single person, but rather a persona; it is a reusable, generalized model of a solution structure to the problem of ambiguity.

Hermeneutics achieves scientific soundness by linking interpretation processes to the following methodical goals:

Objectivity: Objectivity is achieved not by eliminating subjectivity, but by controlling it (Kurt and Herbrik 2022, p. 622). The entire interpretation process is controlled by a strict testing procedure (*sequence analysis* and *falsification*) to exclude arbitrariness (Section 3.3.3). Only in this way can the interpretation be intersubjectively comprehensible. The control of one's own pre-understanding and teamwork help to accomplish the transition from everyday understanding (*constructions of the 1st degree*) to scientific understanding (*constructions of the 2nd degree*), explained in Section 3.3.2.

Traceability: The entire interpretation process, including all discarded hypotheses and deliberations, must be continuously and seamlessly documented to ensure transparency and facilitate later critical review.

3.3.2 The Hermeneutic Attitude

To methodologically control for the researchers' subjectivity and thus achieve scientific insights, a specific attitude must be adopted. This hermeneutic attitude is a methodized reflection that moves the researcher from the everyday mode to the scientific mode. These principles must be kept in mind to adopt the hermeneutic attitude (Kurt and Herbrik 2022, pp. 613–614):

- Nothing may be accepted as self-evident. The researcher must consciously fade out pre-understanding and doubt their own prejudices. The principle is to engage with the data as if it were completely new.
- The seemingly self-evident must be transformed into non-self-evident through estrangement. The willingness to be repeatedly irritated by the data and one's own interpretations is the source of new possibilities for understanding and scientific creativity.
- In contrast to pragmatic everyday understanding, the researcher must adopt a distanced attitude. Researchers should establish an impersonal relationship with their own thoughts and feelings and interpret them at a distance.
- The researcher does not strive for clarity; rather, they embrace ambiguity. The unresolved question of how to interpret a phenomenon (in one way or another) opens up new paths for comprehension.

To clarify the necessary reflective distance, a distinction is made between two levels of construction (Kurt and Herbrik 2022, pp. 612–613):

- Constructions of the 1st degree (everyday understanding) are the meaningful constructions of the everyday world that are carried out by the actors (the researched persons) themselves. It is automatic, reflexive, and unconscious understanding. Actors use these constructions for orientation and communication. For example, an employee says: *"That's an interesting idea"*. In everyday life, their colleague immediately understands: *"That's not an interesting idea; that's a polite rejection"*.

- Constructions of the 2nd degree (scientific understanding) are the conscious, methodically controlled constructions of the researcher. It is a reflexive, self-observing, and controlling understanding that takes conscious distance from the object. The task of Hermeneutics is to make the constructions of the 1st degree of the actors the object of scientific reflection. The researcher asks, for example: *Which latent meaning structures make it necessary that a polite rejection is packaged as praise?* Researchers must thereby develop the capacity for critical self-reflection.

3.3.3 The Interpretation Process

The entire interpretation process serves to control subjectivity and is based on the interplay between two attitudes: the *sense of possibility* and the *sense of reality* (Kurt and Herbrink 2022, p. 620). It is methodically controlled by the *sequence analysis*. The *sense of possibility* describes the ability to develop multiple hypotheses for a single sequence. The central question is: *What could the actor have done or meant instead?* This promotes creative theoretical imagination. The *sense of reality* refers to the empirical testing of these hypotheses against the actual material (the subsequent sequence or context). The function is to control and falsify the developed hypotheses.

The interpretation process and its interplay occur within the sequence analysis. The analysis is conducted fine-grained and sequentially (unit of meaning for unit of meaning) (Kurt and Herbrink 2022, pp. 617–622):

1. **Sequence Isolation:** The current text segment (sequence) is isolated, and all prior knowledge about the context is consciously faded out (estrangement).
2. **Generation of Readings:** Multiple, even improbable, hypotheses are developed.
3. **Testing of Hypotheses:** The anticipations of the readings are tested against the actual following sequence.
4. **Falsification:** Hypotheses that contradict the subsequent sequence are eliminated. The text serves as an objective corrective instance, whereby the interpretation is scientifically substantiated.

This logic can be illustrated by the example of Kurt and Herbrink (2022, p. 616) involving the opening sequence of a text: *"Once upon a time..."*. In the first step of isolation, the researcher focuses solely on this phrase while systematically blocking out prior context. During the subsequent generation of readings, multiple hypotheses emerge; for instance, *Reading A* might assume the start of a traditional fairy tale, leading to the expectation that the next sequence will introduce a protagonist such as *"a princess"* or *"a miller"*. Simultaneously, *Reading B* might interpret the phrase as a nursery rhyme, suggesting a rhythmic continuation like *"an apple"*. When the actual following sequence—*"a politician"*—is revealed in the testing phase, both previous readings are immediately falsified. The introduction of a contemporary political figure shatters the established genre expectations of the fairy tale, necessitating a shift in the interpretation toward a new hypothesis: that

the author employs a fairy-tale framing as a satirical device. Through this process, the case is eventually typified as a specific solution to the problem of portraying a politician as a storyteller or a liar. This continuous oscillation between possibility and reality ensures that the resulting interpretation is objectively grounded in the text's corrective power rather than a mere projection of the researcher's bias.

3.4 Discourse Analysis

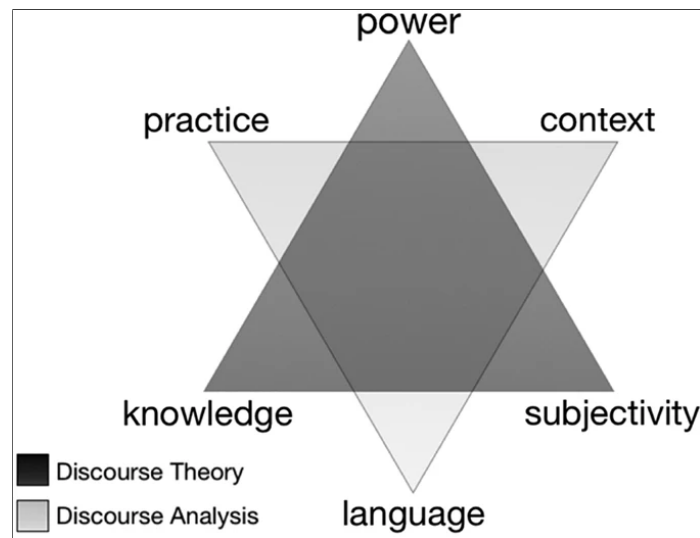


Figure 3.1: Main focus of discourse analysis and discourse theory: "The triangle pointing upwards represents the three main points of interest of discourse theory—an amalgam for relatively abstract and not necessarily linguistic approaches to discourse initiated by post-structuralist authors such as Derrida, Foucault, Laclau, and Mouffe. These points of interest are power, knowledge, and subjectivity. The downward-pointing triangle designates the heterogeneous field of enquiry known as discourse analysis. Here, discourse is problematized as an issue of contextualized language use and practice". Quotation and Image from Zienkowski (2017).

3.4.1 The Discourse

In its broadest sense, a discourse refers to the production of social meaning through the representation and constitution of objects in communicative processes (Traue et al. 2022, p. 627). It is not an individual viewpoint or a private opinion, but a collective regulatory system for producing knowledge about a specific object. A discourse determines how an object is spoken about, thought about, and acted upon, functioning as a socially anchored structure of knowledge and power. It makes an object comprehensible, interpretable, and manageable on an overarching, collective level.

A decisive criterion for determining a discourse is whether different practices relate to or establish an *oversituational*, *overindividual*, and at least provisionally stable meaning (Traue et al. 2022, p. 627). By an *overindividual* system of knowledge is meant a system in which knowledge exists outside and independently of

individual actors (e.g., scientists, politicians, or users) and guides their interpretive processes. By *oversituational* stability, it is meant that the meanings produced by the discourse are not limited to a single interaction (e.g., a meeting or an article), but are oversituational, provisionally stable figures of meaning. The discourse is thus the rulebook of definitions, fears, promises, and narratives that determine how society interprets and regulates the object. Examples include discourses on medicine, technology, or sustainability.

Discourse Terminology

The Archaeology of Knowledge by Foucault (1972) remains the primary source for the analytical elements used to reconstruct historically specific and socially situated orders of knowledge. A *statement* (*énoncé*) is regarded as the smallest and most elementary building block of the discourse. To analyze a *discursive formation*, Foucault identifies four fundamental elements of discourse, the *rules of formation* (Traue et al. 2022, pp. 633–635):

Object: The object is the empirical phenomenon to which a meaning is assigned in the discourse. It is the central theme that is defined, structured, and made evaluable by the collective knowledge system (the discourse). Discourse analysis assumes that the object is not "naturally given", but is produced (constructed) through speaking and the practices connected with it (institutions, laws, technologies). Examples include (a) Artificial intelligence: *How is AI defined in research, politics, and the press? As an opportunity, risk, tool, or autonomous entity?* (b) Sustainability: *What exactly does "sustainable" mean (ecological, social, economic)?* (c) Health: *What is considered "sick" or "normal"?*

Example: In software engineering, "technical debt" is an object. It is not a physical thing, but a discursive object constructed by practitioners to make "invisible" code quality issues visible, measurable, and manageable by management.

Enunciative Modalities: This (*modalités énonciatives*, "modalities of speaking") refers to the status of the speaker and the institutional sites from which they speak (e.g., a hospital, a laboratory, or a press room). It encompasses how an object is constructed through specific forms of expression—such as descriptions, narratives, classifications, and models—and the logical connections among these elements.

Example: In AI ethics, a statement made from the "institutional site" of a university ethics board is viewed as a "regulator", whereas the same statement made by a lead engineer at a tech giant is viewed as "industry standards".

Concepts: Concepts serve to link and relate objects to one another. It is examined in which thematic and strategic context a concept appears and how its use changes. Analysis focuses on the correspondence between different terms: *With which other themes does a concept align? In what relations does it stand to others? How does its use allow for, or even force, the addressing of certain topics?*

Example: The concept of "open source" is linked differently depending on the discourse. In a commercial discourse, it is linked to "cost-efficiency" and "innovation speed"; in a socio-political discourse, it is linked to "transparency", "freedom", and "digital sovereignty".

Strategies: Strategies encompass everything that actors and actor groups intentionally or unintentionally strive for. Strategies are the intentional or unintentional goals that actors pursue through a statement. These are anchored in the discourse's institutional, economic, and technical conditions. Strategies encompass the theoretical choices that emerge from a discourse. They also represent turning points at which different paths of development are taken, based on institutional, economic, or technical conditions.

Example: In AI, "connectionism" (neural networks) became a dominant strategy over "symbolic AI" (logic-based) through strategies of funding and hardware availability.

These four *rules of formation* represent the conditions under which the objects of a discourse are arranged. They describe the emergence and disappearance of concepts and statements, justify the establishment of an object, and provide insight into the strategic orientation of the discourse's architects. As Foucault defines them, *rules of formation* are conditions of existence (but also conditions of coexistence, maintenance, modification, and disappearance) for everything within a *discursive formation*. Further, the *rules of formation* allow certain things to be said while others are silenced. They dictate which statements are acceptable within their framework—that is, which statements possess a "truth value". Statements can therefore lie inside or outside the "truth" of the discourse; those on the outside may be dismissed as mad, radical, eccentric, old-fashioned, or ultimately irrelevant (Traue et al. 2022, p. 635).

Example: The rules of formation for "agile software development" determine that progress must be discussed in terms of "sprints", "velocity", and "backlogs". If a developer attempts to discuss software progress as a "meditative craft", they are operating outside the rules of formation, and their statement is likely to be dismissed as irrelevant.

Together, they form the "big picture"; the *discursive formation* that holds the entire knowledge system together. The *discursive formation* is the larger knowledge complex. It is evidence that the discourse exists as an *oversituational, overindividual* system, and is not merely a random accumulation of texts.

Example: "artificial intelligence" is a discursive formation. It is not just a collection of Python libraries (like PyTorch or TensorFlow), but an entire system of rules that dictates which objects are studied (neural networks), who can speak with authority (machine learning engineers), which concepts are valid (stochastic gradients), and which strategies are pursued (generalizing intelligence via scale).

3.4.2 Discourse Analysis

Discourse analysis is a multidisciplinary research perspective whose methodology draws on concepts from sociology, linguistics, philosophy, communication sciences, psychology, computer science, and statistics. It has been used worldwide in the social and cultural sciences since the 1970s and applies to a wide range of subjects. The research logic aims to develop an understanding of current and historical phenomena in their genesis or specific structure by investigating

either extended time periods (historical method) or differences between orders (comparative method) (Traue et al. 2022, p. 628).

Goals

The central concern of discourse analysis is the investigation of the interplay between a statement and its societal consequences. However, it is not about investigating concrete statements, but rather about their interactions. Discourse analysis investigates the relationship between knowledge and power by exploring: *Which knowledge has prevailed in this discourse and is generally accepted? And: How does this knowledge exercise power, e.g., how does it steer our ways of thinking and acting?* Concretely, discourse analysis pursues three consecutive goals (Traue et al. 2022, p. 640):

1. **Deconstruction:** The goal is to disassemble the discursive formation into its systemic individual elements (objects, enunciative modalities, concepts, strategies). This goal is also described as the reconstruction of the inner-discursive order or the inner structure. This is the necessary groundwork for identifying the discursive formation and for making the mutual stabilization of knowledge and power comprehensible.
2. **Function & Power Analysis:** Here, the goal is to explain the functionality of the discourse. As soon as the rules of formation (this inner structure) are reconstructed, it is connected to societal power relations and non-discursive practices (e.g., material conditions such as organizations, infrastructures, and technology). The mechanisms of power that stabilize the discourse are explained.
3. **Societal Analysis:** The overarching goal is to reconstruct societal power relations and thereby to deconstruct the "naturalism of knowledge" (its self-evidence). Discourse analyses are, alongside the analysis of knowledge, always also institutional or societal analyses (Traue et al. 2022, p. 640).

Keller (2011a) presents a list of guiding questions for discourse research. I find these helpful to understand more concretely how these goals are approached:

- When does a specific discourse appear or disappear again?
- How, where, with which practices and resources is a discourse (re-)produced?
- Which linguistic and symbolic means and strategies are employed?
- Which fields of phenomena are thereby constituted and how?
- Which formations of objects, enunciative modalities, concepts, and strategies does a discourse contain?
- What are its structuring processes and modalities?
- What are the decisive events in the course of a discourse, and how does it change over time?

- How is a discourse reflected in dispositifs?
- Which actors occupy the speaker positions with which resources, interests, and strategies?
- Who is the carrier, addressee, and audience of the discourse?
- Which modes of appropriation can be traced?
- Which references does the discourse contain to other discourses?
- How can a discourse be related to spatially-temporally more or less far-reaching social contexts?
- What (power) effects result from a discourse, and how do these relate to societal fields of practice and "everyday representations"?
- What explanations exist for the characteristics of a discourse?
- Which phenomena are explained by the discourse analysis?

3.4.3 The Meaning of Power

The understanding of power used in discourse analysis fundamentally differs from the everyday conception of power as compulsion or possession. Power here is anonymous, structural, productive, and circulates throughout the entire social structure (Traue et al. 2022, pp. 629,635):

- **Power is seen as the production of truth.** Power determines what counts as "*true*", "*rational*", and "*normal*". Discourse excludes specific perspectives by labeling them "*unscientific*" or "*marginal*". For example, the medical discourse defines what a "*disease*" is and which symptoms warrant treatment.
- **Power establishes subjects.** Power constructs who we are (the subjects) and which roles we are supposed to take on. It creates defined identities and instructions for action. For example, the discourse on mental health constructs the "*depressed patient*" with specific characteristics and treatment obligations.
- **Power steers actions.** Power does not compel, but rather guides and governs people's actions by establishing specific ways of thinking and solutions as the only rational ones. For example, the discourse on prevention and self-optimization guides individuals to monitor their health voluntarily.
- **Power defines the limits of the sayable.** Power determines who is allowed/legalized to speak and whose statements carry weight (knowledge). For example, in the discourse on technology development, a data scientist's statement often carries more weight than an ethicist's or an ordinary user's.

3.4.4 Variants of Discourse Analysis

There are many variants of discourse analysis (Traue et al. 2022, pp. 629–633), which differ in (a) their theoretical origin (e.g., linguistics vs. sociology), (b) the type of data examined (e.g., purely linguistic vs. visual/audiovisual), (c) the methodological approach (e.g., qualitative-interpretive vs. quantitative-lexicometric), and (d) the focus of the analysis (e.g., relations of power vs. subject constitution). Common to all variants, yet with differing emphases, is the question of how the relationship between knowledge, power, actor constellations, and social construction processes is to be understood. Some of these variants include: Lexicometric Discourse Analysis (Pêcheux 1995), Discourse Linguistics (Bernstein 1971), Critical Discourse Analysis (Fairclough et al. 2011), Governmentality Studies (Bröckling et al. 2000), Discourse Analysis in Gender Studies (Engel 2009), and Visual Discourse Analysis (Maasen et al. 2006).

The procedure of the Sociology of Knowledge Approach to Discourse (SKAD) will be discussed in more detail in Sections 3.5 and 3.6. This variant of discourse analysis played a central role in motivating, informing, and ultimately developing the Discourse Analysis Tool Suite (Chapter 4). The D-WISE project (Section 1.3) placed particular emphasis on methodological support for SKAD.

3.5 Procedure of a Discourse Analysis

The procedure for social scientific discourse analysis and the Sociology of Knowledge Approach to Discourse (Keller 2011b) does not follow a rigid, one-size-fits-all schema. Instead, the analytical elements and levels of description must be developed in accordance with the specific research object. However, the process generally proceeds from the initial formation of a corpus to the coding of statements and the reconstruction of discursive formations, and finally to analyses of the discourses' function and power, and societal analysis (Traue et al. 2022, pp. 633–642).

3.5.1 Phase 1: Preliminary Work

The preliminary work lays the theoretical foundation and provides orientation in the research field. First, the *theoretical clarification* takes place, encompassing the precise determination of the research object and the clarification of the theoretical terminology (object, enunciative modalities, concepts, strategies). The research question must be discursively informed and aim to investigate knowledge and power. Next, this phase continues with the exploration of the field (Traue et al. 2022, p. 636): This is an exploratory investigation conducted through a literature review, document inspection, and, if possible, the acquisition of one's own experiences through participant observation, to develop the necessary theoretical sensitivity for corpus formation.

3.5.2 Phase 2: Corpus Formation

The compilation of the corpus (Traue et al. 2022, pp. 635–636) determines the object, the content, and the scope of the discourse analysis:

For the compilation of the corpus, only *natural data* is considered, which is *process-generated*, meaning that the data is not produced by the researchers themselves, but is found in the field (e.g., newspaper articles, literature, photography, statements of discourse producers). The spectrum of data materials that is considered is broad and encompasses all kinds of documents, newspaper articles, literary sources, as well as images and videos.

The sampling strategy is oriented toward the *theoretical sampling* (see Section 3.2). Here, the corpus is determined at the outset but can be dynamically expanded or restricted throughout the research. The exploratory investigation of the first phase guides the initial compilation.

The criterion for the validity of the corpus is the determination and evidence of one or more discursive formations. This means the corpus must be compiled in a way that clearly enables the identification of recurring patterns in which concepts, strategies, and objects consistently overlap. This constructed corpus must serve as empirical proof that a stable, collective knowledge system exists and is not merely a random accumulation of individual statements. Furthermore, the corpus must permit the generation of falsifiable statements about the chosen object. Essentially, the corpus must be strategically structured to both show the pattern (the discourse) and allow rigorous testing and correction of hypotheses about it.

3.5.3 Phase 3: Reconstructing the Discursive Formation

This phase serves the deconstruction, one of the goals of discourse analysis. Here, the discursive formation is disassembled into its individual elements to disclose the oversituational figures of meaning and their conditions of existence.

Developing the Discursive Formation

The working out of the discursive formation (Traue et al. 2022, pp. 637–638) is the first central evaluation step and extends over the entire research period. The discursive formation is the specific gestalt or the internal structure of the discourse. It describes how the formal and content-related elements of the discourse are arranged. Across the various investigation periods, the discursive formation and its rules of formation are developed by elaborating the objects, enunciative modalities, concepts, and strategies of the discourse. Importantly, the actual elements of this discursive formation are not known before data analysis. They must be inferred from empirical data across statements.

Coding

Coding (Section 3.2.2) is the methodical means to elaborate the rules of formation (Traue et al. 2022, p. 638): In this process, the discursive elements occurring in the texts (concepts, object descriptions, argumentations, interpretive figures, metaphors, etc.) are identified and isolated. This serves as preparatory work to recognize the knowledge and to bridge the gap between the empirical evidence (*what is said*) and the theoretical concepts (*what the said means*).

3.5.4 Phase 4: Fine-grained Analysis of Function & Power

This phase relates to one of the goals of discourse analysis. Here, based on the previously identified rules of formation, the discursive formation is elaborated and connected to societal power relations to explain how mechanisms of power stabilize the discourse. The analysis proceeds on two levels (Traue et al. 2022, pp. 641–642):

The first level (inner-discursive order) analyzes how a discursive practice becomes a stable order of knowledge. The production of truth is determined by clarifying which rules of the sayable are present. This includes analyzing which speaker and subject positions are dominant, which concepts and objects are dominant or taboo, and which scarcities of legitimate statements exist. This analysis explains the discourse's internal functionality.

The second level analyzes how the discourse's functionality relates to the outside. In other words, the inner order is related to the outside of the discourse, particularly to the general historical-societal situation. The limits of the discourse are determined, which requires moving beyond purely linguistic analysis. It examines the structural, institutional, economic, and professional power constellations in which the discourse is embedded. For this purpose, linguistic forms are connected with non-linguistic objectivations. The central question is which non-discursive practices accompany the production of knowledge and how these functionally stabilize the system of knowledge. Thus, it is analyzed how an outside of the discourse is constructed and which techniques are employed therein.

Methodically, this two-level analysis is based on the coding work from Phase 3. Typical statements are subjected to a fine-grained analysis to understand the specific manifestations of the discourse. This analysis could focus on several key components (Traue et al. 2022, pp. 638–639), including:

1. **Interpretative Patterns:** Fundamental meaning-generating schemata that are spread by discourses. *Which interpretive patterns are produced in the discourse, how do they change over time, and what are their conditions of emergence?*
2. **Classifications:** These represent institutionalized forms of social typification that are fixed and collectively accepted within the discourse. They function by summarizing disparate things into defined classes or group formations. These classifications are not neutral; they carry inherent moral or political content. The analysis must therefore determine which specific classifications are undertaken within the discourse and which evaluations or attributions they entail. *Which classifications (group formations) are undertaken within and through the discourse, and which attributions do these contain?*
3. **Narrative Structures:** The narrative structures that connect interpretive patterns and classifications into a story with plot, personnel, and moralization (dramaturgical plot). *How are the patterns and classifications connected into a story?*
4. **Speaker Positions:** The conditions of access and participation in the discursive events, to determine who is heard as a legitimate expert and who

is excluded from the discourse. *Who is admitted as a legitimate expert or addressee in the discourse?*

5. **Non-discursive Practices:** non-linguistic objectivations (e.g., laws, technologies, material infrastructure) that are part of the discourse. *Which infrastructures or institutionalized routines are produced or stabilized by the discourse? How do they manifest their power in social reality?*

3.5.5 Phase 5: Social Analysis

The social analysis is the concluding synthesis and the critical integration of the research results. The second level of the power analysis serves as the empirical bridge to the social analysis.

Social scientific discourse analyses are always institutional or societal analyses (Traue et al. 2022, p. 640). The goal is to reconstruct the power relations expressed in discourses as societal power relations, thereby deconstructing the naturalism of knowledge (its self-evidence). This final phase shows that what is taken for granted in a society is, in fact, the result of historical processes and a stabilization of specific power structures.

3.6 Sociology of Knowledge Approach to Discourse

The Sociology of Knowledge Approach to Discourse (Keller 2011b) understands itself as a comprehensive method for researching social realities that manifest in linguistic and material practices. The social scientific discourse analysis presented in the two preceding Sections 3.4 and 3.5 draws on SKAD in its methodological and analytical fundamentals. In this chapter, we want to emphasize the central contributions of Reiner Keller to discourse analysis.

SKAD is distinguished by integrating Foucault's discourse analysis into the interpretive paradigm of sociology. While Foucault's discourse analysis often focused on anonymous, overarching power structures, SKAD shifts the focus toward the discursive construction of social reality through intersubjective and medially mediated communication (Keller 2011b, p. 48). This approach understands actors not as mere subjects of power, but as rule-interpreting agents who actively participate in and position themselves within discursive events (Traue et al. 2022, p. 632; Keller 2011b, p. 54).

To provide a structured foundation for this analysis in empirical research practice, SKAD incorporates the methodologies of Hermeneutics (Section 3.3) and Grounded Theory (Section 3.2).

3.6.1 Hermeneutics & Reflexivity

SKAD is fundamentally hermeneutically oriented. Keller (2011b, p. 61) emphasizes the unavoidability of a hermeneutic-interpretive attitude (Section 3.3.2) throughout the entire research process. Discourses are deeply rooted in social contexts, which is why interpretation by researchers is necessary.

To guarantee the scientific quality criteria of objectivity (the control of subjectivity) and traceability, Keller (2011b, pp. 61–62) demands consistent methodical

reflection. This reflection serves, on the one hand, to make the entire interpretation process (from material selection to the final interpretation) transparent and traceable, thus securing the empirical validity of the procedure. The establishment of this transparency is considered a necessary condition for the empirical grounding of SKAD. On the other hand, the reflection is about fading out the researcher's prior knowledge, doubting one's own prejudices, and thus adopting a distant attitude to control subjectivity.

3.6.2 Grounded Theory as an Analytical Tool

To transfer the interpretive openness of Hermeneutics into a methodically controllable procedure, SKAD relies on Grounded Theory methods (Keller 2011b, p. 63). GT delivers the instruments to develop new, data-based concepts and theories from empirical material. SKAD thus uses the GT methodology but applies it to a specifically discourse-analytical epistemic interest.

In particular, the coding techniques (open coding, axial coding, selective coding) and the *constant comparative method* are used to analyze the material, generate concepts, and relate them to one another. Thus, new data are constantly compared with previously coded data and with developed categories or theories. The first goal is to develop the elements of the discourse step by step, iteratively, from the empirical data themselves.

Another element of GT adopted by SKAD is theoretical sampling for corpus formation, along with the closely related concept of theoretical saturation. These methods enable a targeted, theory-driven, and successive expansion of the material until no new relevant information contributes to further theory generation, and the categories are sufficiently developed and linked.

To fulfill the demands of Hermeneutics for objectivity and traceability, the researchers are encouraged to continuously record their interpretations, methodical decisions, and emerging theoretical ideas in memos.

3.6.3 Analytical Focus

Beyond the methodological grounding, Reiner Keller established two central analytical foci in discourse analysis: not treating discourse as a mere abstract text system, but rather grasping it as a social practice encompassing the investigation of practices, infrastructures, materialities, and actors (Keller 2011b, p. 48). Both focuses were also briefly described in the preceding section:

Dispositif Analysis: This focus includes the materiality and infrastructure of the discourse. In particular, the non-discursive practices are examined (e.g., the spatial, institutional, and technological conditions that enable, structure, and accompany discourses). The dispositif analysis thus examines the anchoring of the discourse in social reality (Keller 2011b, p. 55).

For example, the discourse on "trustworthy AI" is not merely a linguistic debate but is materialized in an extensive dispositif. This infrastructure includes non-discursive practices such as implementing "ethics-by-design" checklists in corporate software development lifecycles, specialized GPU hardware clusters required to run models, and institutional settings such as "AI ethics boards"

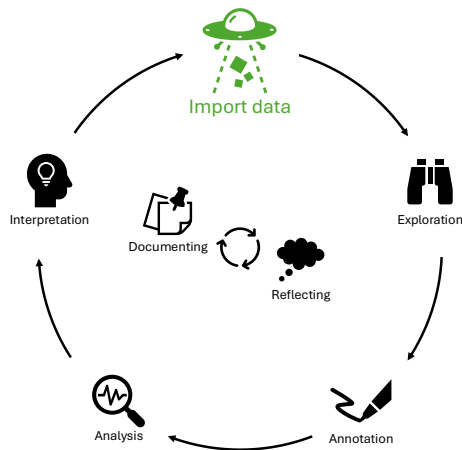
and the legal framework of the EU AI Act. These material and organizational elements function as a "strategic ensemble" (Keller 2011b, p. 56) that responds to the urgency of societal safety concerns by standardizing and disciplining how AI is built and deployed.

Subject Position Analysis: Here, the focus is on analyzing actors' and speakers' positions. It examines how individuals are positioned, addressed, and constituted as subjects within discourses, and how they inhabit, resist, or negotiate those positions (Keller 2011b, p. 55).

For example, the discourse on "AI-assisted development" (e.g., regarding tools like GitHub Copilot²) creates new subject positions for developers. They are increasingly addressed not as "implementers" of logic, but as "AI orchestrators" or "code reviewers". This subjectivation entails new instructions for action, such as the mandate to "critically verify all automated snippets", and a shift in professional identity from an "independent creator" to an "efficient manager of automated outputs". Research could then explore how developers adopt or resist these roles, for example, by insisting on "human-centric" coding practices.

The theoretical frameworks of Grounded Theory, Hermeneutics, and the Sociology of Knowledge Approach to Discourse provide the methodological standards required for discourse analysis. However, the practical application of these methods is increasingly hindered by the scale of digital data. To bridge this gap, the qualitative requirements of the empirical social research must be translated into a functional technical system. The following chapter introduces the Discourse Analysis Tool Suite, an integrated, open-source platform that embodies these methodological principles. We move from the conceptual foundations of discourse research to the architectural and functional implementation of the software foundation that supports the iterative research process.

2. <https://github.com/copilot>



4

Discourse Analysis Tool Suite

Contents

4.1	Motivation	70
4.2	Related Tools	71
4.3	Feature Overview	80
4.4	Terminology	82
4.5	Technical Architecture	82
4.5.1	Backend Implementation	83
4.5.2	Client Server Communication	85
4.5.3	Frontend Implementation	86
4.5.4	Deployment & Orchestration	86
4.6	Import & Preprocessing	87
4.6.1	Motivation	87
4.6.2	Challenges	88
4.6.3	Data Import & Preprocessing in DATS	88
4.7	Introducing Alice	91
4.8	Conclusion	92

This chapter introduces the Discourse Analysis Tool Suite (DATS) (Koch et al. 2022; Schneider et al. 2023), the software framework of this dissertation, within which all research contributions were developed. DATS¹ was developed to address the limitations of existing qualitative research tools—specifically in terms of multimodality, scalability, and accessibility of NLP technology—within an integrated, open-source environment. Importantly, the development of DATS is a collaborative effort; as the lead developer, I contributed over 60% of the total commits and edited over 70% of the codebase. Beyond my technical contributions, I supervised and managed the integration of work from several student assistants. The platform also benefited significantly from the expertise of my colleagues (and co-authors on several related publications), Florian Schneider and Fynn

1. <https://github.com/uhh-lt/dats>

Petersen-Frey, who primarily contributed to backend development. Especially during the foundational phase, they contributed substantially to the platform’s initial conceptualization and backend architecture.

This chapter lays the groundwork for subsequent chapters by establishing the context and rationale for developing DATS (Section 4.1) and its potential to address the challenges researchers in the humanities face. Section 4.2 provides a comparative analysis of related tools, situating DATS within the landscape of proprietary and open-source CAQDAS and corpus linguistics software. Next, this chapter provides a comprehensive overview of features (Section 4.3) and defines the core terminology used within DATS and throughout this dissertation (Section 4.4). We then detail the technical architecture (Section 4.5), based on Schneider et al. (2023), explaining the distributed backend, the AI model serving layers (vLLM and Ray), and the interactive React frontend.

A central focus of this chapter is the import and preprocessing pipeline (Section 4.6), which utilizes machine learning to convert multimodal data (text, image, audio, and video) into a unified, searchable text domain. Finally, we introduce Alice (Section 4.7), a research persona who will guide the reader throughout this dissertation through an idealized COVID-19 news discourse analysis, demonstrating how the platform’s capabilities support the iterative journey from data ingestion to interpretation.

4.1 Motivation

Discourse analysis (DA, see Sections 3.4 and 3.5) is a complex and time-intensive methodology. DA relies on manual coding and interpretation, which limits studies to small corpora. Even when focusing on small amounts of relevant material, this manual process requires immense effort. With the ongoing digitalization of public life, there has been unprecedented growth in the amount of data available online. Social networks, forums, and websites are a central arena for social discourse and opinion formation. The vast amount of available, potentially relevant data exceeds the capacity of conventional DA methods, creating a methodological gap for researchers.

The Discourse Analysis Tool Suite was developed to close this gap by addressing fundamental limitations in current research practice. The D-WISE research project (Section 1.3) had three primary motivations related to the challenges of digital discourse analysis that drove the design of DATS:

- **Multimodality:** Modern discourse appears across diverse channels, including text, image, audio, and video. DATS should accommodate the full range of data formats relevant to discourse analysis, including linguistic, visual, and auditory materials. This is a key challenge faced by digital humanities due to the lack of appropriate multimodal methods.
- **Scalability:** The increasing availability of large-scale data requires efficient computational methods. Hence, DATS should improve the efficiency and analytical quality of hermeneutic analyses by enabling the processing of large volumes of multimodal data, thereby supporting big data applications in digital humanities research.

- **Open-Source Platform:** The absence of a free, open-source platform designed explicitly for digital discourse analysis is the final motivation. Existing commercial or proprietary tools are often unavailable to small research teams in digital humanities.

To integrate natural language processing and artificial intelligence methods to support digital discourse analysis, the DATS team prioritized developing a unified, stable platform. We deliberately chose to develop an integrated tool suite rather than a collection of loosely coupled, standalone tools. This integrated approach was chosen because fragmented research workflows, where researchers use separate tools for tasks like corpus management, coding, and visualization, introduce issues that the unified environment avoids:

1. **Data Inconsistency:** Using multiple tools requires constant data export and re-import, compromising version control and hindering the traceability of the research workflow.
2. **Workflow Friction:** Switching between disparate software environments disrupts analytical flow and increases the researcher's cognitive load.
3. **Collaboration:** Disconnected tools complicate data sharing and synchronized work among research team members.

The primary objective of DATS was to provide comprehensive digital support within a single, integrated tool for the iterative, hermeneutic research process at the heart of discourse analysis. This included satisfying the core requirements established above: support for multimodality, scalability, and the ability to process large volumes of multimodal data. By consolidating all necessary discourse analysis functions (Section 1.4)—data preprocessing, exploration, annotation, analysis, and interpretation—DATS aims to serve as a flexible digital environment that provides a seamless end-to-end workflow. This aims to ensure robust data consistency, minimize technical overhead, and make it easier for digital humanities researchers to adopt and apply advanced NLP methods.

4.2 Related Tools

This section presents several existing Computer-Assisted Qualitative Data Analysis Software (CAQDAS) related to our work on the Discourse Analysis Tool Suite. Computer-Assisted Qualitative Data Analysis Software refers to a class of tools designed to help researchers manage, organize, analyze, and interpret unstructured data. This data typically includes text (transcripts, documents, literary works), as well as audio, video, and image files. The primary function of CAQDAS is to provide a systematic environment for developing theory, applying codes and categories to segments of data, performing sophisticated queries across the coded material, and generating visual outputs to synthesize findings. While the qualitative designation is key, modern CAQDAS also integrates capabilities for mixed-methods research, combining qualitative data interpretation with quantitative analytical features, such as statistical processing of document attributes or code frequencies.

Proprietary Tools

The software landscape can be broadly segmented into established proprietary platforms and a growing ecosystem of open-source solutions. The proprietary market, dominated by commercial vendors, typically offers mature, feature-rich, and well-supported tools that cover a wide range of qualitative and mixed-methods needs. The following sections detail the most popular proprietary systems.

MAXQDA: MAXQDA² is distinguished by its comprehensive support for qualitative, quantitative, and mixed-methods research designs. Its functional objective is to serve as a unified analytical environment for managing and interpreting heterogeneous data materials. The platform offers extensive input compatibility, accommodating standard text formats (PDF, DOCX, TXT), diverse media streams (MP4, MP3, including transcription/time-stamping for synchronized analysis), and structured data formats such as SPSS (.SAV) and Excel (.XLSX) for integrating quantitative variables. Furthermore, the system is equipped to handle unstructured web content and supports direct data ingestion from platforms such as YouTube and Twitter.

The system's functionality centers on coding. The coding mechanism allows researchers to establish a hierarchical code taxonomy using manual methods such as drag-and-drop coding and in-vivo coding. Later, the annotations can be explored and retrieved by a sophisticated query engine that supports complex search criteria based on Boolean logic, regular expressions, code co-occurrence, and document variables. For analytical synthesis, MAXQDA includes various visual analytics components, such as the *Code Matrix Browser* for visualizing code frequency across documents and *MAXMaps* for generating conceptual models and relationship graphs. Documentation is maintained through memos, which can be attached directly to codes or data segments, and through a dedicated logbook for process journaling.

In recent iterations, MAXQDA has incorporated advanced generative AI capabilities under the brand names *AI Assist* and *MAXQDA Tailwind*. This functionality leverages large language models to enhance analytical throughput by automating demanding tasks. Key LLM-driven services include automated summarization of documents or coded segments, AI coding (providing suggested codes or subcodes with explicit commentary on the reasoning for researcher validation), and a conversational interface that enables natural language dialogue with the coded data, providing traceable references to the source text. Additionally, the platform employs machine learning for high-accuracy, multilingual transcription services.

MAXQDA projects are operated locally, preserving data integrity on the user's device. However, its advanced features, namely *AI Assist* and the *TeamCloud* collaboration system, necessitate cloud processing. The vendor adheres to stringent data security protocols, including GDPR compliance and a zero-data-retention policy for LLM-based processing, ensuring that data used for analysis is not stored or used for model training. The tool is widely adopted across the social sciences and education, specifically supporting mixed-methods studies. Collaborative research is supported through *TeamCloud*, which facilitates asynchronous work

2. <https://www.maxqda.com/>

management, project merging, and the intercoder agreement tool for calculating and reviewing coding reliability across distributed team members.

NVIVO: NVivo³ is designed to facilitate qualitative and mixed-methods research. It addresses the challenge of managing and interpreting heterogeneous collections of unstructured data. The platform provides extensive data ingestion capabilities, supporting standard textual formats (e.g., PDF, DOCX), structured survey output, and diverse multimedia materials, including audio, video, and images. Furthermore, NVivo incorporates advanced functionality, such as the *NCapture* tool, for integrating and analyzing unstructured social media and web content.

The system supports several analytical functionalities. The platform allows researchers to establish hierarchical coding schemes ("*nodes*"). For data exploration, NVivo provides querying mechanisms that enable researchers to run complex searches across coded data, alongside dedicated text analysis tools for word frequency counts and word cloud generation. Further analytical synthesis is supported by comparative tools (e.g., cases, classifications, attributes) and visual analytics (e.g., charts and mind maps) to illustrate relationships. NVivo provides memo functionalities for researchers to record analytical reflections, notes, and interpretations throughout the research process. Additionally, it maintains an audit trail documenting the analytic process, providing a transparent record of the research.

In recent versions (NVivo 15), the system integrated the *Lumivero AI Assistant*, an advanced capability leveraging large language models to enhance analytical throughput. Specific LLM-driven services include automated document and segment summarization, sentiment analysis, thematic analysis, and AI coding suggestions.

NVivo is established within the research community, primarily supporting qualitative and mixed-methods studies across the social sciences. NVivo projects can be maintained locally, ensuring data integrity on the user device. However, when using the AI assistant, data is temporarily transmitted for processing and is governed by a strict zero-data-retention and deletion policy after the task to ensure data security. Collaborative research is supported through multiple models, including project merging and the NVivo *Collaboration Cloud*, which facilitates real-time, cross-platform co-coding and incorporates tools for permission management and intercoder agreement calculation to ensure reliability across research teams.

Atlas.ti: Atlas.ti⁴ is a widely adopted CAQDAS that supports multimodal data processing across various research methodologies. It provides an integrated environment for the systematic analysis and theory development from unstructured data. The platform supports the ingestion and synchronization of heterogeneous data, including standard textual documents (PDF, DOCX), all major multimedia formats (audio, video, images), and social media data. The analytical units are quotations and coded segments, allowing researchers to segment and code material.

Atlas.ti includes several functionalities. Exploration mechanisms include a search system that leverages lemmatization and synonyms, as well as Boolean

3. <https://nvivo.de/>

4. <https://atlasti.com/>

queries across coded segments and document metadata. For coding, the system supports both deductive and inductive coding strategies, enabling the construction of a flexible code taxonomy through hierarchical structures and the grouping of concepts via Code Families. Analytical synthesis is supported by tools such as code co-occurrence analysis and dedicated visualization modules, including *Networks* for conceptual mapping and data-driven graphics (*Sankey Diagrams* and *Tree Maps*). Project documentation and analytical reflexivity are managed through memos linkable to any project entity.

In recent releases, Atlas.ti has prioritized integrating advanced generative AI capabilities, powered by the OpenAI GPT models. This enables semi-automated and automated analytical services. Key LLM-driven functionalities include AI coding for automated descriptive coding proposal generation and *Intentional AI Coding*, which leverages researcher-defined research questions to guide the LLM's analysis and code specialization. Furthermore, the platform offers a conversational AI interface for natural-language querying of the dataset, along with proprietary NLP functions for named entity recognition, sentiment analysis, and transcription.

The tool supports both locally-managed desktop projects and a cloud-based web version. The platform is widely adopted across the academic and professional sectors, supporting both qualitative and mixed-methods research designs. For collaborative work, the web version provides real-time collaboration capabilities, and the system supports team reliability assessment via the inter-coder agreement tool. The vendor notes compliance with GDPR and implements a zero-data-retention policy for all data processed by the integrated LLM services.

While MAXQDA, NVivo, and ATLAS.ti are widely adopted and feature-rich platforms, the proprietary market also includes several notable contenders. QDA Miner⁵, for instance, emphasizes integrating qualitative analysis with statistical data exploration, enabling researchers to transition between text coding and quantitative methods seamlessly. Dedoose⁶ is distinguished by its purely cloud-based architecture, which facilitates real-time, cross-platform collaboration and is meant for large-scale, distributed team projects.

Traditional CAQDAS platforms are the industry standard for qualitative research. They offer robust features for manual coding, memoing, and organizing research projects. However, while these tools excel at managing "standard" qualitative datasets (e.g., dozens of interview transcripts), they often face issues when scaled to the thousands or tens of thousands of documents required for comprehensive data analysis. Furthermore, they require significant manual effort and have limited built-in capabilities for advanced automated NLP tasks such as large-scale semantic clustering or LLM-driven synthesis. Their focus remains primarily on the researcher's manual interpretation rather than providing a high-performance, semi-automated pipeline for identifying discourse-specific patterns across massive corpora.

5. <https://provalisresearch.com/products/qualitative-data-analysis-software/>

6. <https://www.dedoose.com/>

Open-source Tools

In contrast to commercial systems, open-source CAQDAS tools incur no cost and often offer community-driven extensibility. These alternatives are often favored by researchers in institutions or fields with budget constraints, or by those who aim to contribute to or customize the tool.

CATMA: CATMA (Computer Assisted Text Markup and Analysis) (Gius et al. 2023) is an open-source, web-based CAQDAS developed to integrate the qualitative methodology of hermeneutic close reading with quantitative text analysis methods. Its primary application area is literary studies and digital humanities research, focusing exclusively on textual data in standard file formats, but it lacks support for multimedia streams (audio/video). The system's features are centered on flexible, *undogmatic* annotation and maintain adherence to TEI/XML interoperability standards.

Data exploration leverages the *Analyze* module, which incorporates a custom query language for retrieving text and metadata, supplemented by a *Natural Language Query Builder* for enhanced accessibility. The *Tagger* module facilitates both deductive and inductive coding by enabling the construction of customizable, multi-level tagsets. Analytical output includes corpus statistics (e.g., frequency counts), alongside visual analysis components such as keywords in context, distribution graphs, and word/tag clouds, supporting the iterative cycle between micro- and macro-level text investigation. A comment feature allows users to add comments to text passages or annotations, which can also be used for team communication.

CATMA offers a modular approach to advanced capabilities. For example, the external Python package GitMA provides programmatic access to the raw TEI data layer, enabling custom extensibility, such as executing custom NLP pipelines or calculating inter-annotator agreement scores outside the web interface. Furthermore, Stereoscope⁷ is a web-based prototype for visualizing hermeneutic exploration and argument construction and can be integrated into the CATMA workflow.

QualCoder: QualCoder⁸ is an open-source, client-based application for Computer-Assisted Qualitative Data Analysis. It operates across multiple platforms (Linux, Windows, and macOS) and supports a heterogeneous input corpus, including various textual formats (DOCX, PDF, EPUB), images, audio, and video data.

The system's functionality centers on coding. It allows coding text, images, audio, and video materials using a hierarchical code taxonomy. Beyond manual methods, QualCoder integrates auto-coding features based on regular expressions and sentence segmentation. Retrieval is managed through standard search functions, a dedicated case matrix for cross-case analysis, and mechanisms for assessing inter-coder reliability using Cohen's Kappa. Further analytical outputs include visual rendering of the code-category hierarchy and exportable reports in HTML and CSV formats. QualCoder supports reflection and interpretation through journal notes and memos.

7. <https://stereoscope.threedh.catma.de/>

8. <https://qualcoder.wordpress.com/>

Recent development has integrated advanced AI capabilities through a module that leverages external large language models, specifically APIs for OpenAI (GPT), Anthropic (Claude), and Google (Gemini), and supports local models via Ollama. These features enable AI-assisted coding, topic analysis, and a conversational AI interface that provides source-traceable summaries of coded data. Architecturally, the primary project files are stored locally in an SQLite database, maintaining data integrity on the user's device and ensuring primary offline operation. However, the optional AI functionality temporarily transmits data to third-party cloud services; the vendor asserts a zero-data-retention policy for this LLM-based processing.

QualCoder is a single-user desktop application, so collaboration is limited to asynchronous project file transfer. The tool is used in the social sciences and complies with the REFI-QDA standard for cross-platform data exchange.

Corpus Linguistics Tools

A distinct yet related category of software arises from corpus linguistics. These tools are oriented toward quantitative text analysis—focusing on linguistic patterns, frequency counts, collocations, and statistical distributions across large text collections. While their main methodologies are statistical, they may also include features for manual categorization that bridge the gap to qualitative interpretation.

Corpus Explorer: CorpusExplorer⁹ is an open-source software primarily designed for corpus linguistic analysis and large-scale, explorative text mining. Its functional objective is to quantitatively analyze linguistic patterns in extensive text collections, serving as a primary tool for computational humanities. The system is architecturally designed for high-throughput text processing, leveraging a fast in-memory database to manage large-scale corpora. It supports over 100 diverse file formats, including linguistic XML standards, CSV, and JSON, thereby accommodating a wide range of textual sources such as transcripts, social media data, and scholarly documents.

The analytical architecture emphasizes automated, quantitative methods over manual, qualitative coding. Data preparation and annotation are highly automated, integrating external NLP tools, such as TreeTagger, OpenNLP, and Stanford POS taggers, for part-of-speech and named-entity recognition. For exploration, the system provides classical corpus linguistic functionalities, including keyword-in-context, n-gram, co-occurrence, and difference analyses. It includes various visualization and reporting modules, such as heatmaps and stylometrics. Interestingly, the tool operates on reproducible snapshots (sub-corpora), isolating analysis results from the source corpus.

CorpusExplorer does not incorporate generative AI services. Instead, its advanced capabilities are based on integrating external, established machine learning pipelines for linguistic tagging. For custom analytical needs, the platform provides a console-based interface and a comprehensive software development kit, enabling researchers to programmatically integrate custom NLP/ML models in languages such as R and Python, thereby supporting extensible research pipelines.

9. <https://corpusexplorer.de/>

CorpusExplorer is a single-user desktop application that ensures all project data is processed and stored exclusively on the user's local device. Hence, the platform lacks collaboration features. The tool is used within the computational and corpus-linguistic research communities.

There are various other tools for corpus exploration, quantitative analyses, and pattern identification. Voyant Tools¹⁰ is a web-based, open-source tool, offering a variety of text analysis and visualization functions, including word clouds, trends, and collocation analysis, designed for exploratory analysis without requiring installation. AntConc¹¹ is a standalone concordance program that provides detailed linguistic analyses, such as frequency lists and clusters of words and n-grams.

Tools like AntConc are powerful for identifying statistical patterns in language, such as collocations, concordances, and frequency distributions. While invaluable for linguistic analysis, they are often less flexible for the qualitative needs of discourse analysis, such as identifying narrative structures or managing the iterative, theory-building workflow typical of the Sociology of Knowledge Approach to Discourse (SKAD).

Task-specific Tools

While the platforms reviewed above offer integrated, comprehensive solutions, many open-source projects take a different approach, focusing their development efforts on optimizing a single task within the analysis pipeline. This specialization often results in highly performant tools for key tasks, such as annotation, corpus exploration, and processing of specific data modalities.

The process of systematic annotation is a foundational element in both CAQ-DAS and computational analysis (e.g., for training machine learning models). Consequently, a dedicated set of tools exists to streamline this task, particularly for textual data. Open-source solutions like INCEpTION (Klie et al. 2018) and Label Studio¹² provide collaborative, web-based environments that support complex annotation schemes, including sequence labeling, relational annotation, and document classification, making them effective for large team-based annotation projects. Similarly, tools like Text Annotator (Abrami et al. 2020) (a simple, client-side tool) and Taguette¹³ (a lightweight web-based application) focus on providing straightforward annotation and segmenting functionality, offering an accessible entry point for basic annotation tasks.

Such tools extend beyond text to other modalities. For instance, the proprietary software Transana¹⁴ is designed for the synchronized analysis of video and audio data, offering accurate time-stamping, frame-by-frame analysis, and integrated transcription capabilities, which are important for fields such as conversation analysis and behavioral research. In the digital humanities and historical research

10. <https://voyant-tools.org/>

11. <https://www.laurenceanthony.net/software/antconc/>

12. <https://labelstud.io/>

13. <https://www.taguette.org/>

14. <https://www.transana.com/>

domains, tools such as Recogito2 (Simon et al. 2017) specialize in geospatial and temporal coding, allowing researchers to tag locations, dates, and entities directly on maps and historical documents.

Other task-specific tools focus exclusively on the exploratory phase of analysis. The Unified Corpus Explorer (Bönisch et al. 2025) is a recent example of an AI-enhanced exploration tool for detailed, annotation-based analysis of large, complex datasets. It is built on the DUUI¹⁵ pipeline and the UIMA¹⁶ annotation format, designed to unify the heterogeneous landscape of NLP-annotated big data corpora across domains such as digital humanities. Its functionality is centered on advanced retrieval, offering multiple dedicated search layers: named-entity search, semantic similarity search, and semantic role search (enabling “*Who did what to whom*” queries). Further, the system integrates a retrieval-augmented generation chatbot that dynamically fetches context from the corpus, enabling multi-turn, question-answering interactions with the data. The system also supports analysis through a document reader with annotation highlighting and a 3D taxonomy view for geographical and taxonomic visualization.

Platforms like Inception focus on the technical aspects of annotation, often for training machine learning models. These task-specific tools provide sophisticated interfaces and features for tasks such as labeling or exploring data, but are typically not designed as end-to-end suites (from data ingestion and coding to final visualization and synthesis) for analyzing discourse.

The landscape of analysis software is vast and continually expanding. For researchers seeking a broader overview of tools suitable for digital humanities, text mining, and qualitative analysis, the forTEXT¹⁷ catalog provides an excellent, comprehensive, and categorized overview.

Distinguishing DATS from Existing Tools

The primary distinction of the Discourse Analysis Tool Suite (DATS) lies in its methodological and technological integration. While existing tools focus either on manual coding of small samples (CAQDAS) or purely statistical overviews of large corpora (corpus linguistics), DATS is specifically built to support the interplay of distance and depth required by modern discourse studies.

DATS is explicitly designed to handle large data volumes, addressing the “big data” challenge within the digital humanities. Technologically, the platform uses scalable database architectures (Section 4.5)—integrating relational systems, inverted indices, and vector stores—to manage document volumes that exceed the capacity of traditional proprietary software. Methodologically, DATS operationalizes this scale by supporting the ideas of Computational Grounded Theory (CGT) and Annotation Scaling (Section 6.4). CGT (Nelson 2020) is an iterative framework that leverages computational techniques to identify latent patterns, which are then refined through human interpretation and validated across the corpus. Annotation

15. <https://duui.texttechnologylab.org/>

16. <https://uima.apache.org/>

17. <https://fortext.net/>

Scaling (Section 6.5.3) facilitates the practical transition between these stages, operationalizing inductive qualitative codes into fixed annotation schemas used to train supervised machine learning models.

DATS integrates these concepts through features that support a workflow similar to CGT. In the pattern detection phase, *Perspectives* (Section 5.4) facilitates unsupervised NLP techniques (semantic clustering), enabling researchers to identify latent structures in massive corpora that would be invisible to manual reading, while statistics based on automated NLP pipelines (Section 4.6.3), such as named entity recognition, provide initial quantitative overviews to "*find the needle in the haystack*". To facilitate pattern refinement and guided deep reading, DATS provides an integrated annotation interface and semantic search capabilities, enabling researchers to isolate relevant segments and locate semantically similar occurrences across the corpus. Within this phase, the *LLM Assistant* (Sections 5.5 and 6.6) can be utilized to aid the researcher by suggesting relevant passages, increasing the efficiency of the interpretive process. Finally, DATS enables pattern confirmation by allowing researchers to "*close the loop*": the *Annotation Assistant* (Section 6.7) and related classification modules enable the fine-tuning of transformer-based models on human-validated annotations. Once applied to the entire corpus, analysis functionalities like code frequency analysis assist in the quantitative validation of the confirmed patterns across the rest of the "haystack". A similar workflow is also realized in the *Concept Over Time Analysis* (COTA, Section 7.4) extension. COTA integrates semantic search and annotation into a single human-in-the-loop feature, allowing researchers to define concepts, refine them through iterative feedback, and visualize their evolution over time.

Furthermore, DATS prioritizes accessibility through interactivity, democratizing advanced NLP technologies for researchers without computational expertise. It provides low-barrier graphical user interfaces for NLP techniques such as semantic search, large language models (*LLM Assistant*), clustering (*Perspectives*), and model training (COTA, *Annotation Assistant*). All of these functionalities are interactive, enabling a human-in-the-loop paradigm where researchers can iteratively refine analytical outcomes through trial and error. This interactive approach ensures that the machine remains a supportive instrument for the researcher's interpretive intent.

This synthesis of manual and automated analysis is further enhanced by the *Whiteboard* extension (Section 8.4), which serves as a unique 2D workspace for theory building. Unlike static network views of other platforms, *Whiteboards* allow for the spatial arrangement and linking of various materials—including NLP-generated objects (entities, topics), research artifacts (annotations, memos), and interpretations (links, notes)—thereby supporting the non-linear hermeneutic interpretation process central to discourse analysis.

The design of DATS is also explicitly aligned with the Sociology of Knowledge Approach to Discourse (Section 3.6), providing an integrated, end-to-end workflow for all related tasks (import, exploration, search, annotation, analysis, interpretation) that minimizes friction associated with switching between disparate software environments.

Finally, DATS distinguishes itself through a commitment to data sovereignty. In contrast to modern AI-augmented tools that rely on external, cloud-based APIs, DATS is designed for local deployment via containerization and uses efficient,

locally executable models. This "privacy by design" approach ensures that sensitive research data remains within the researcher's controlled environment, satisfying the privacy requirements of academic and institutional research.

4.3 Feature Overview

The functionality of DATS adheres to the abstracted discourse analysis process (Section 1.4) derived from the Sociology of Knowledge Approach to Discourse Analysis (Section 3.6). This section outlines the foundational capabilities supporting the five primary stages of the workflow: import, exploration, annotation, analysis, and interpretation. The core contributions of this dissertation are integrated into the DATS platform and examined in detail, with accompanying screenshots, in the Chapters 5–8.

Import: DA necessitates handling diverse data. DATS provides an import function that supports all standard text, image, audio, and video formats. Furthermore, an integrated URL importer allows direct ingestion of web content and associated media.

Imported data is processed through a machine learning pipeline for automatic enrichment: Video and audio data are automatically transcribed using a speech-to-text model. Image data undergoes object recognition and image captioning, which automatically generates a short description. Text data is enriched by first detecting the language and then using corresponding models to automatically identify named entities and keywords. This preprocessing pipeline is essential for converting all multimodal data into the text domain, allowing them to be queried and analyzed together. Additionally, semantic embeddings are calculated for both text and image documents. The import features are explained in detail in Section 4.6.3.

Exploration: To manage and explore large corpora, DATS provides advanced search and filtering capabilities that include the additional information extracted and generated during the import step. The standard search function supports querying for relevant phrases, concepts, or keywords using Boolean operators, fuzzy search, and prefix queries. This unified search works across all document types (text, image, audio, video) because the preprocessing pipeline converts all documents into the text domain. A filter mechanism enables detailed querying based on automatically extracted information, custom metadata, or named entities. Together, these two features enable detailed, fine-grained searches. Another feature integrates with this filtering mechanism: The cross-modal semantic search allows users to find semantically similar images and sentences based on a source sentence or image. This enables the user to quickly locate "*more of the same*" content across modalities.

For document organization, DATS includes two systems. The tag system enables users to define a hierarchy of categories to organize and categorize documents effectively. The metadata system allows users to define custom metadata fields (text, lists, numbers, dates) to store additional, relevant information for documents.

The basic exploration, search, and filtering features, along with their extension via *Perspectives* and the *LLM Assistant*, are explained in detail in Chapter 5.

Annotation: DATS provides two manual annotation functions to support qualitative coding: For images, users can define rectangular bounding boxes over image segments and assign them to a categorical code. For texts, arbitrary text fragments (spans) can be selected and assigned a category. Because of automatic transcription during import, this capability also extends to audio and video documents. Direct annotation of video or audio segments is not supported in DATS.

The flexible categorization system is key for discourse analysis. DATS includes predefined categories derived from automatic entity recognition (e.g., person, organization) and image object recognition (e.g., animal, furniture). To support the iterative, flexible coding process in discourse analysis, DATS allows users to create and manage their own hierarchical code system for assignment to text spans and bounding boxes.

The basic annotation features and their extension with the *LLM Assistant* and the three-phase *Annotation Assistant* are explained in detail in Chapter 6.

Analysis: DATS provides standard quantitative analysis functions and statistics primarily for document overview, data exploration, and simple evaluations.

1. **Word Frequency Analysis:** This feature filters stop words and aggregates term frequencies across the entire corpus or a filtered subset. Results can be visualized using a word cloud.
2. **Code Frequency Analysis:** This function aggregates and visualizes annotation counts (codes) across the corpus or sub-corpus (filterable by modality and annotator) using bar and pie charts. This shows both the frequency of code assignment and the frequency of text passages marked with a specific code.
3. **Contextual Statistics:** Code frequency analysis is applied dynamically to search results. This allows immediate visualization of key statistics, such as the most frequently mentioned entities (e.g., person, organization) within the current filtered search set. This capability extends to automatically recognized keywords and user-assigned tags.

The basic analysis features and their extension with the *Concept Over Time Analysis* are explained in detail in Chapter 7.

Interpretation & Reflection: To support the hermeneutic steps of interpretation and reflection, DATS implements two features. The memo functionality allows users to attach short text notes (memos) to nearly all research objects in DATS, including documents, tags, codes, and annotated text passages. This allows researchers to capture interpretive thoughts at the point of discovery. Additionally, DATS includes a logbook with a dedicated view that provides a tabular overview of all created memos, along with a text editor. While flexible in use, the logbook's primary purpose is to document the overall research process using accumulated memo data.

The basic interpretation and reflection features and their extension with *Whiteboards* are explained in detail in Chapter 8.

4.4 Terminology

Because DATS forms the foundation for this dissertation and all subsequent contributions, definitions of the terminology used throughout the tool suite and this dissertation are provided here:

1. **Document:** This term refers to any raw data material that researchers intend to analyze. A document is modality-agnostic and may be textual (e.g., a news article or website), an image (e.g., a photograph or chart), an audio recording (e.g., an interview), or a video (e.g., social media content).
2. **Tag:** A tag is part of a user-defined, hierarchical classification system (the tagset) used exclusively for organizing and categorizing documents. A single document can be associated with multiple tags (e.g., politics, news article, conflict).
3. **Metadata:** Arbitrary key-value pairs associated with a document, providing descriptive or statistical context. Typical metadata includes language, keywords, video/audio length, word count, publisher, author, and publication year.
4. **Annotation:** A segment or excerpt of a document marked by a user and associated with a specific code. For example, a user may select the text passage "*Eiffel tower*" in a news article and assign the code "*Sightseeing*".
5. **Code:** A code is part of a user-defined, hierarchical category system (the codebook) used exclusively to classify segments of documents. A code assigned by a user to a document excerpt forms an annotation. While DATS allows users to define custom code hierarchies, it also provides a small set of predefined categories, such as those typically used by named entity recognition systems (e.g., person, organization, location).
6. **Memo:** A short, structured textual note (with a title and body) that can be linked to virtually any DATS research object (documents, tags, codes, or annotations). Memos serve as a tool for capturing interpretive thoughts at the point of discovery.

4.5 Technical Architecture

DATS' architecture design prioritizes technical flexibility and long-term viability, serving as an extensible foundation for subsequent research and future feature development. The following description is based on (Schneider et al. 2023) and has been updated to reflect the current state of the architecture.

DATS is explicitly designed to address the primary technical requirements and challenges identified in Section 1.5: scalability and accessibility. The architecture

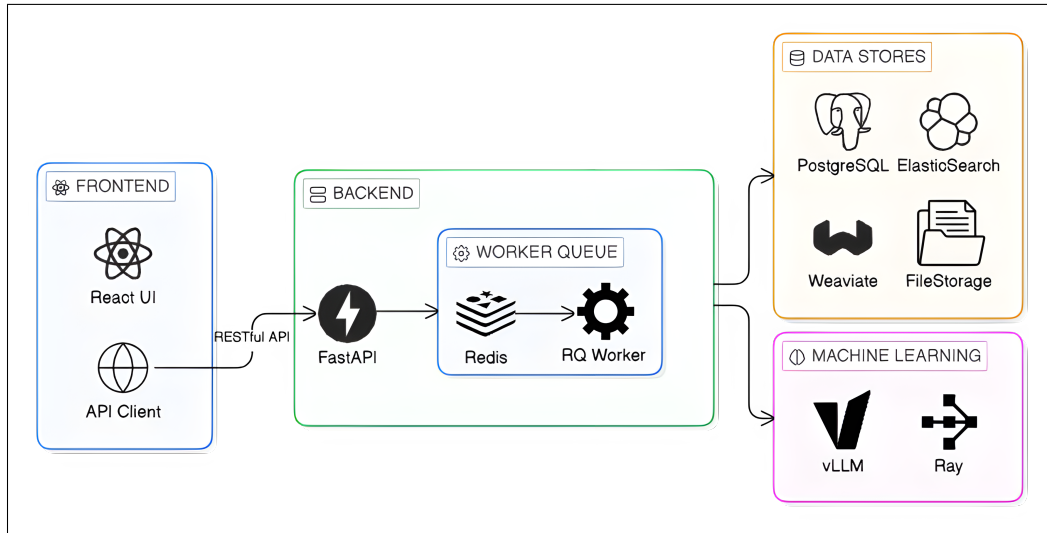


Figure 4.1: Architecture of the Discourse Analysis Tool Suite

must efficiently handle the processing, storage, and retrieval of large, growing, and multimodal data volumes. Further, the system must provide intuitive user interfaces that abstract complex computational processes, ensuring high usability for digital humanities researchers who may lack expertise in machine learning, natural language processing, and computer science.

DATS is implemented as a modern client-server web application. The system consists of a Python backend (Section 4.5.1) and a React frontend (Section 4.5.3) that communicate via a secure Representational State Transfer (REST) Application Programming Interface (API). The backend handles core functionality and data persistence, while the frontend handles the user-facing interface and interaction logic. A schema of the architecture is depicted in Figure 4.1.

4.5.1 Backend Implementation

An essential requirement for DATS is scalability, meaning the system’s ability to operate with large and growing amounts of data and many simultaneous users. This requirement was considered and implemented from the beginning by using only scalable software frameworks and libraries, and by deploying and orchestrating the system with modern platform-independent technologies. The backend is implemented entirely using Python and a variety of open-source libraries. Furthermore, the backend was designed and implemented using common software patterns and idioms to ensure high-quality software that meets essential criteria, including extensibility, availability, scalability, and maintainability. As a requirement for open-source compliance, we ensured that all third-party software dependencies are also open-source-licensed.

Data Storage & Retrieval: Arguably, the most crucial component of DATS is its underlying data model. This data model connects the elements of the business logic (e.g., projects, users, memos, or annotations) to heterogeneous, multimodal documents in their various representations.

Business logic data is modeled as SQL tables using the Python ORM framework SQLAlchemy¹⁸ and stored in a PostgreSQL¹⁹ database. SQLAlchemy is also used to execute SQL queries to retrieve structured data.

Text, image, audio, video, or mixed-modality documents are represented and stored in several ways. DATS utilizes a heterogeneous storage approach: The raw files uploaded by users for analysis within DATS are stored on disk. A relational database (PostgreSQL) stores project metadata, user data, annotations, and other structured business logic. It is the most important database, storing all relevant research materials. Additionally, textual data is stored in inverted indices using Elasticsearch²⁰ and is retrieved via *ES Query DSL* executed utilizing the Python Elasticsearch client library²¹. This database enables all standard textual search operations (e.g., keyword search and fuzzy matching). Semantic vector embeddings of entire documents or segments (e.g., sentences) are stored in Weaviate²² indices. This vector store is used for fast cross-modal semantic search.

This storage approach, integrating PostgreSQL for relational data, Elasticsearch for full-text search, and Weaviate for semantic search, delivers the performance and flexibility needed to handle the diverse data volumes and complex queries of DA research.

Job System: The backend incorporates a distributed task queue to handle computationally intensive tasks, such as data preprocessing, AI model training, and AI inference, which is crucial for ensuring system responsiveness and efficient resource utilization. Background workers, implemented with RQ²³ and Redis²⁴, execute these tasks asynchronously. Together, these technologies implement a flexible job system that allows DATS to schedule and run computationally intensive tasks in the background without blocking the main application flow. Depending on the job type, GPU-powered or CPU-powered RQ workers are employed.

Machine Learning Models: For machine learning tasks, DATS leverages specialized model serving technologies. DATS uses vLLM²⁵ and LiteLLM²⁶ to serve large language models. This combination enables the execution of complex natural language processing tasks with high throughput and low latency through batch processing, prefix caching, and paged attention (Kwon et al. 2023). Importantly, vLLM includes a constrained generation feature that allows the exact definition of model output formats. This ensures that all model responses can be parsed easily and correctly. vLLM supports the generation of structured outputs using xgrammar²⁷ (Dong et al. 2024) or guidance²⁸ (Moskal et al. 2025) as backends.

18. <https://www.sqlalchemy.org>

19. <https://www.postgresql.org/>

20. <https://www.elastic.co/>

21. <https://elasticsearch-py.readthedocs.io/>

22. <https://weaviate.io/>

23. (<https://python-rq.org>)

24. <https://redis.io/>

25. <https://vllm.ai>

26. <https://www.litellm.ai/>

27. <https://github.com/mlc-ai/xgrammar>

28. <https://github.com/guidance-ai/lguidance>

Ray²⁹ provides a framework for hosting and managing a variety of other machine learning models for classification, embedding generation, and general processing, ensuring scalability and efficient resource utilization. Specifically, Ray is used to host: (a) Spacy³⁰ models for named entity recognition and sentence splitting, (b) a Whisper³¹ model for automatic audio/video transcription, (c) SentenceTransformers³² models to compute image and sentence embeddings for the semantic search feature. Using modern libraries such as SentenceTransformers and vLLM makes switching the models served straightforward. This ensures that DATS stays up to date with recent advances in NLP and provides users with state-of-the-art models.

A central design pillar of the Discourse Analysis Tool Suite is the use of open-source and open-licensed libraries whenever possible. This philosophy extends to integrated models by exclusively using open-weight models and is driven by two primary considerations:

1. **Data Privacy and Sovereignty:** Academic research in the digital humanities often involves sensitive data. Local execution ensures that no data leaves the controlled research environment, satisfying the privacy requirements of universities and research institutions.
2. **Hardware Accessibility:** To ensure the tool remains usable in typical academic or resource-constrained settings, the architecture prioritizes "small" but state-of-the-art model variants. Using half-precision (FP16) and quantized variants, DATS enables high-performance inference on consumer-grade GPUs.

Consequently, DATS currently integrates the instruction-tuned, open-weight large language model Gemma 3 (G Team 2025). More details on Gemma 3 are in Section 2.6.

4.5.2 Client Server Communication

All DATS functionality is encapsulated within a REST API. The API is implemented using FastAPI³³. FastAPI automatically generates an OpenAPIv3³⁴ schema and a SwaggerUI³⁵ interface. This eases the development of user interfaces and other external applications that consume the DATS API.

To simplify communication and enforce strict type checking between clients and the server, the parameters and return types of all API endpoints are defined as data transfer objects (DTOs) using Pydantic³⁶ models. These DTOs are also widely used within the backend to decouple entities from database sessions, facilitate communication between different system components, and transmit data between background workers.

29. <https://www.ray.io>

30. <https://spacy.io/>

31. <https://github.com/openai/whisper>

32. <https://sbnet.net/>

33. <https://fastapi.tiangolo.com>

34. <https://www.openapis.org>

35. <https://swagger.io>

36. <https://pydantic.dev/>

To serve raw text, image, audio, and video files directly to the client, a high-performance web server, nginx³⁷, is employed.

4.5.3 Frontend Implementation

The frontend is implemented using React³⁸ and TypeScript³⁹ to create a dynamic and interactive web application.

It communicates with the backend via its REST API to enable all platform features. The process leverages the backend's OpenAPIv3 schema to ensure seamless, type-safe integration. A code generator, openapi-typescript-codegen⁴⁰, is used to generate an API client automatically. This generates client code that fully supports all backend functionality and includes TypeScript interfaces for all parameter and return types, as defined by the Pydantic DTOs. Data fetching, caching, and synchronization are then handled by TanstackQuery⁴¹. This middleware significantly improves application performance and responsiveness while reducing network bandwidth usage.

The client state is managed by Redux⁴², which provides powerful features essential to a complex research tool, such as state persistence across sessions.

Finally, the user interface and interaction adhere to the recognized open-source Material Design System by Google⁴³. Ready-made components from the MUI library⁴⁴ are used extensively. For complex data visualization, specialized libraries are employed: Recharts⁴⁵ for plotting and Material React Table⁴⁶ for tabular data views. These libraries, combined with custom-tailored elements, deliver an intuitive, accessible interface for digital humanities researchers.

4.5.4 Deployment & Orchestration

DATS is deployed using modern containerization technology, which ensures platform independence and operational flexibility. Every distinct system component — including all databases and indices, the API, background workers, and the frontend — is packaged into a separate Docker⁴⁷ container. These containers are orchestrated using Docker Compose files that define the entire DATS system stack. This deployment approach offers several technical advantages:

- The software can be conveniently deployed on any modern system or infrastructure with minimal configuration, making the DATS platform independent. This is crucial for local deployments, especially for projects that handle confidential or privacy-sensitive data.

37. <https://nginx.org/>

38. <https://www.reactjs.org>

39. <https://www.typescriptlang.org>

40. <https://github.com/ferdikoomen/openapi-typescript-codegen>

41. <https://tanstack.com/query/latest>

42. <https://redux.js.org>

43. <https://m3.material.io/>

44. <https://mui.com>

45. <https://recharts.github.io/>

46. <https://www.material-react-table.com/>

47. <https://www.docker.com/>

- The use of containers enables the flexible deployment of components across different machines. For example, computationally intensive components, such as background workers, can run on a dedicated GPU server. Memory-intensive components, such as indices and databases, can run on a storage-optimized server.
- The system can be readily scaled horizontally using container orchestration technologies such as Docker Swarm⁴⁸ or Kubernetes^{<https://kubernetes.io/>}, ensuring high availability and accommodating increasing user loads.

4.6 Import & Preprocessing

The import and preprocessing stages are fundamental to the entire discourse analysis workflow within DATS. While this dissertation focuses on later stages, a brief discussion of these initial steps is essential, as all downstream tasks depend on the quality, integrity, and structure of the input data.

4.6.1 Motivation

Preprocessing is the initial step to prepare data for subsequent analysis. This process involves taking raw, often unstructured data and converting it into a structured, standardized format suitable for both computational and manual analysis.

For text data, several standard practices have been established in natural language processing (Jurafsky and Martin 2025). These typically include text cleaning (removing noise, irrelevant characters, and document artifacts), tokenization (breaking text into discrete units), normalization (standardizing text properties such as case), and linguistic processes such as lemmatization or part-of-speech tagging. Historically, these linguistic operations facilitated the transformation of raw text into a reliable input format for computational models. With the introduction of embedding-based, pre-trained language models (Sections 2.3 and 2.5), however, operations such as lemmatization or part-of-speech tagging are often no longer required. This is because modern transformer-based models use subword tokenization and contextual embeddings, which allow them to capture semantic and grammatical relationships between different word forms directly from raw text without manual morphological reduction.

Because discourse analysis is a research methodology used across diverse disciplines, projects require the flexibility to handle a range of data sources, including text files, online archives, web pages, and social media content. Hence, managing these diverse data formats is crucial for a flexible analysis platform. As described by Franken et al. (2020), automated corpus creation approaches, such as crawling or focused crawling (Chakrabarti et al. 1999), are promising for digital qualitative analyses.

48. <https://docs.docker.com/engine/swarm/>

4.6.2 Challenges

However, retrieving, importing, and preprocessing heterogeneous data present several obstacles that distract researchers from their analytical tasks. These preparatory stages often constitute the most significant portion of the research timeline, yet their impact on final results is frequently disproportionate to the effort invested. As noted by Kunilovskaya and Plum (2021), data preprocessing often follows the Pareto principle, where *"80% of work generates only 20% of profits"*, consuming the majority of the total research time. Addressing the following technical challenges within an integrated tool suite offers a significant opportunity to solve common pain points for digital humanities researchers, greatly streamlining their workflows and allowing them to focus more on their analytical tasks, rather than dealing with technical complications:

- **Diverse Data Formats:** Researchers may need to deal with a wide range of input sources (e.g., text, audio, video, image, social media) coming in various formats (e.g., HTML, XML, TEI, JSON). Converting these varying formats to a single, standardized structure for unified analysis is a technically demanding, non-research activity.
- **Data Acquisition:** Acquiring data from platforms (e.g., X⁴⁹, Facebook⁵⁰, or Instagram⁵¹) or public sources (e.g., websites, forums, or blogs) necessitates developing specialized custom crawling software or navigating complex and often restrictive APIs. This work is time-consuming and requires specific technical skills.
- **Computational Resources:** Analyzing large-volume data (such as extensive social media streams, online news archives, or long-term archival data) imposes technical limits on standard research hardware. Processing these big datasets requires significant computational resources and expertise in high-performance computing.
- **Sensitive Data:** Discourse analysis projects may involve sensitive data that legally cannot be processed by commercial third-party services. This may keep researchers from using convenient, easy-to-use cloud-based commercial platforms.

4.6.3 Data Import & Preprocessing in DATS

The import feature of DATS is designed for maximum flexibility, enabling the ingestion of multimodal data across a wide range of file formats, including text, images, audio, and video. The following description is based on Schneider et al. (2023) and has been updated to reflect the current state of the preprocessing pipeline.

To facilitate the creation of large, diverse corpora, DATS provides two import mechanisms: (a) Users can upload data directly through the user interface, which is suitable for standard research files. This mechanism also supports uploading

49. <https://x.com/>

50. <https://www.facebook.com/>

51. <https://www.instagram.com/>

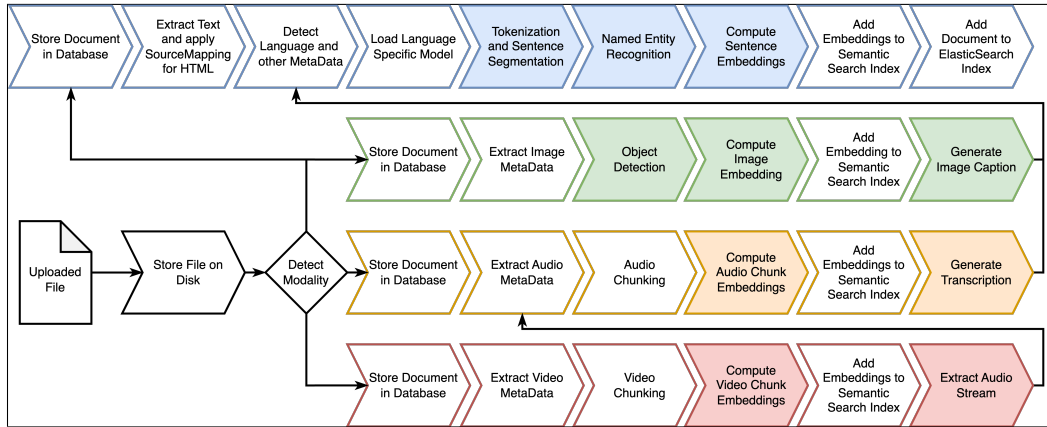


Figure 4.2: DATS multimodal preprocessing pipeline. Each step is executed by either a GPU or a CPU worker of the RQ-based job system. Steps involving machine learning models are highlighted. Figure from Schneider et al. (2023).

data archives, simplifying the import of large text corpora or publicly available datasets. (b) DATS allows researchers to provide a list of URLs for automated data acquisition, a functionality crucial for building corpora from web content. The technical implementation uses Scrapy⁵² and BeautifulSoup⁵³ to scrape website content, including text and embedded media such as images, videos, and audio files, which are then integrated into the DATS document model.

The preprocessing pipeline is a central component of the DATS architecture. It aims to extract as much information as possible from imported data, enriching the materials' metadata to enable complex search queries, filtering, and statistical analysis. Further, it is designed to convert all data to the text domain to support unified querying.

A schematic overview of this pipeline is shown in Figure 4.2. The pipeline is realized as a distributed job system using the RQ framework. When a document is uploaded, a series of actions (a flow) is applied based on its modality. Most flow outputs are stored in a PostgreSQL database. Embeddings of texts and images are stored in the Weaviate vector database to enable multimodal similarity search, and textual content is stored in Elasticsearch indices for efficient lexical search.

The steps of each flow are executed in parallel where possible, enabling fast processing and easy horizontal scaling by adding more workers. Currently, the system supports text, image, audio, and video documents, resulting in four distinct flows. Common to all flows is storing the uploaded file on disk and registering it in the database with basic metadata (e.g., data format, dimensions, duration).

Text: The text flow processes documents in standard formats, including HTML, PDF, Word, and TXT. Docling⁵⁴ handles content extraction for PDF and Word files, while Readability.js⁵⁵ is used for HTML to discard irrelevant structures (e.g., navigations or ads), preserving only the main content. Any embedded images, videos, or audio files are extracted and run separately through their respective

52. <https://scrapy.org/>

53. <https://beautiful-soup-4.readthedocs.io/>

54. <https://github.com/docling-project/docling>

55. <https://github.com/mozilla/readability>

processing flows. Depending on its length, the document is chunked (defaulting to 10-page splits) to ensure that subsequent transformer-based language models can correctly process the content within their token limits. Next, relevant metadata such as word counts or document length is computed. More importantly, the text's language is detected using GlotLID⁵⁶, which is a prerequisite for language-specific model application. For supported languages (English, German, and Italian), spaCy executes typical NLP preprocessing, including tokenization, sentence segmentation, and named entity recognition. Finally, sentence embeddings are computed with a pre-trained multilingual CLIP (Radford et al. 2021; Reimers and Gurevych 2020) model from the SentenceTransformers⁵⁷ (Reimers and Gurevych 2019) library.

Image: The image flow is designed to convert visual content into a searchable, text-based format. A pre-trained DETR model⁵⁸ (Zhu et al. 2021) is applied for object detection. Then, a global semantic image embedding is computed using the same CLIP encoder model as in the text flow. Finally, an image caption is generated using a large language model that processes both text and images with a simple, strict prompt to produce a concise, entity-aware caption. This caption is then passed directly through the text flow to enable both lexical and semantic search on the image's content. However, a current limitation is that the model only returns an English caption.

The following prompt is used to generate image captions: *Create a brief one-sentence image description. If you recognize any person or other named entities, use their names. Return the description in a single line without any new lines or special characters. Do not include any additional text or explanations. The description should be concise and to the point, focusing on the image's main subject. Avoid using overly complex language or jargon.*

Audio: The audio flow processes standard audio formats (e.g., MP3 or WAV) to make their content searchable. A textual transcription of the audio is generated using a Whisper (Radford et al. 2022) model. The resulting audio transcription is then treated as a standard text document and passed through the text flow. This integration supports both lexical and semantic search of spoken content, making audio files discoverable via text queries.

Video: The video flow processes standard video documents (e.g., MP4 or MOV) to analyze both the audio and visual streams. The audio stream is extracted from the video and immediately processed via the audio flow. This ensures that the spoken discourse within the video is transcribed and indexed for lexical and semantic search. The video's visual component is ignored. Frame-by-frame visual analysis or object detection is planned for future work.

The data import and preprocessing features in DATS address many of the pain points (Section 4.6.2) that researchers encounter in discourse analysis. By

56. <https://github.com/cisnlp/GlotLID>

57. <https://sbnet.net/>

58. <https://github.com/facebookresearch/detr>

supporting various data formats and providing a robust, automatic preprocessing pipeline, DATS helps researchers efficiently manage and analyze diverse data sources. The pipeline is explicitly designed to handle large amounts of data, leveraging scalable technologies to ensure efficient processing across modalities. Finally, it improves data quality by cleaning, standardizing, and enriching the input material, preparing it for subsequent computational and manual analysis.

4.7 Introducing Alice

Throughout this dissertation, the reader will be guided through an idealized research workflow that illustrates how the Discourse Analysis Tool Suite supports the iterative, methodologically rigorous demands of qualitative discourse analysis. This journey is narrated through the work of Alice, a digital humanities scholar who uses DATS to apply qualitative and quantitative methods.

Alice's background is in empirical social science, and her methodological expertise is rooted in the Sociology of Knowledge Approach to Discourse. Her research focuses on understanding German news discourse surrounding the COVID-19 pandemic from 2020 through mid-2023, using a corpus of ≈ 4000 articles from major German outlets: *Der Spiegel* and *Die Zeit*.

As Alice approaches this data corpus, her interests are broad and interpretive. She wants to understand the underlying structure of the debate itself. She wants to identify the central conflicts and interpretive patterns that governed the public conversation—for instance, how issues like vaccination were framed as a clash between "*Individual Freedom*" and "*Societal Obligation*", or how concepts like "*Trust in Vaccination*" were discursively constructed. Since discourse is dynamic, she also wants to map the historical evolution of these ideas, systematically observing when key debates peaked, how the framing of risks shifted over time, and pinpointing the major turning points in the pandemic discourse. Finally, she is interested in analyzing power and legitimacy by tracing which specific political and institutional actors dominated the conversation and which rhetorical strategies they used to influence the narrative at critical moments.

Throughout this thesis, Alice will demonstrate how the tools developed in this dissertation can be used to transform these qualitative research interests into empirically grounded analysis.

Project Setup with Alice

Alice begins by accessing DATS through her web browser. After logging in, she creates a new collaborative project for her COVID-19 news discourse analysis project and invites her colleagues.

To kick off the project, Alice opens the import dialog (see Figure 4.3). This interface streamlines the process of ingesting diverse data formats: First, she uploads her existing corpus, which includes a ZIP archive of potentially relevant German news articles (PDFs, DOCX, raw text) from *Der Spiegel* and *Die Zeit*, via the drag-and-drop area on the left (1). Later in her research project, Alice can incrementally expand her corpus by adding online news articles. She can use the URL input field on the right side (2) to paste a list of relevant article links

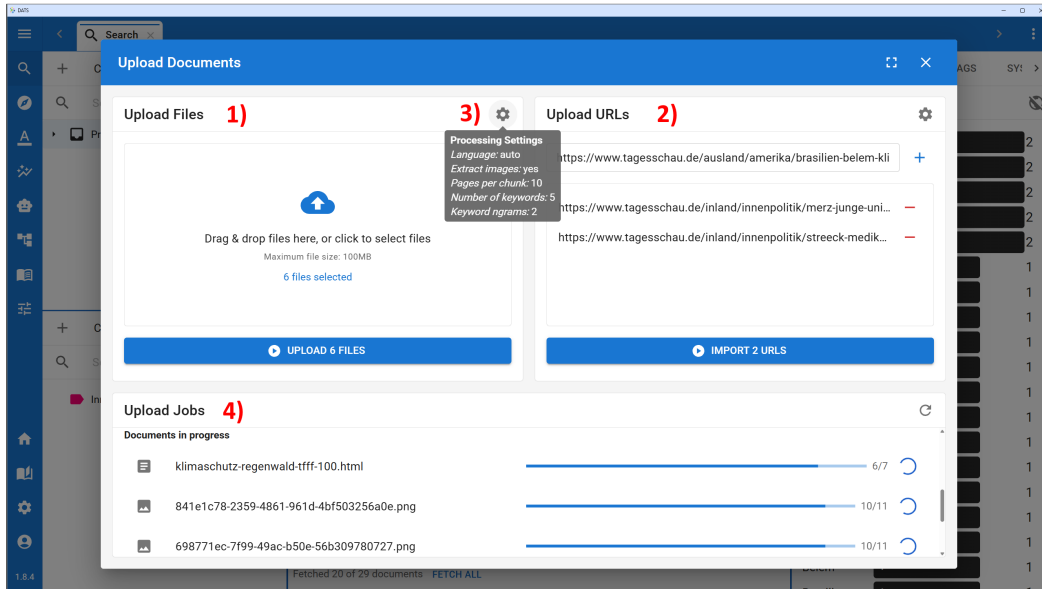


Figure 4.3: DATS import user interface. It supports file uploads via drag-and-drop (1) and automated website scraping via a URL (2). Preprocessing settings can be adjusted (3). The progress bar (4) monitors the preprocessing pipeline.

from news outlets that DATS should scrape. Alice can use the settings button to adjust several import parameters (3).

DATS is designed to handle this multimodality, processing raw text files, web pages (HTML), images, and even transcribing audio and video content. Once submitted, the material is routed through the automated preprocessing pipeline, which is monitored by a progress bar at the bottom (4). While her document corpus is processed by this machine-learning-powered pipeline, which automatically cleans, enriches, and prepares the documents for analysis, Alice takes a short break.

When the background processing is complete, Alice verifies the results using the health check interface (see Figure 4.4). This feature provides insights into the pipeline and the status of document preprocessing. She filters the view by modality (text, image, audio, video) using the toolbar at the top (1) to ensure her various files are correctly categorized and processed. The main table (2) lists each document and shows status icons for all preprocessing steps (e.g., language detection, entity recognition, and indexing). This allows Alice to identify errors quickly. If any preprocessing step fails (e.g., due to an error or a corrupted file), she uses the *retry button* (3) in this feature to re-run the process for those specific documents, ensuring data integrity before proceeding to the exploration phase.

4.8 Conclusion

This chapter presented DATS, the technical foundation upon which all core contributions of this dissertation are based. We detailed the tool suite’s core features (Section 4.3), established the DATS terminology (Section 4.4), and provided an in-depth technical overview of its architecture (Section 4.5). The focus of this technical chapter was on the robust, scalable software architecture and the multimodal preprocessing pipeline (Section 4.6), both of which are central to the

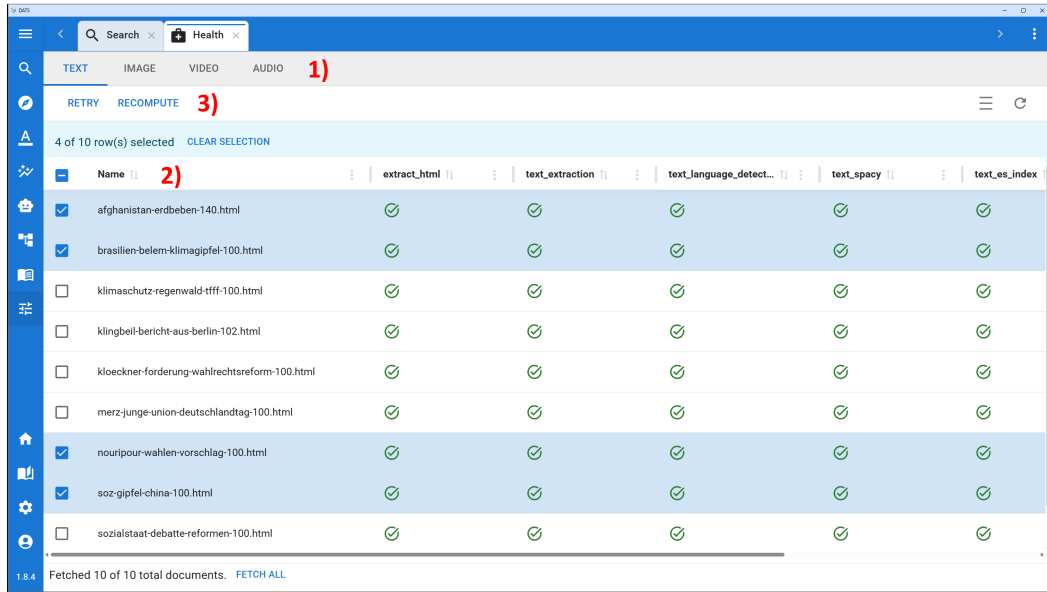


Figure 4.4: DATS health check user interface. It provides transparency into the preprocessing pipeline. Users can filter by modality (1), verify successful steps via status icons (2), and retry failed steps (3).

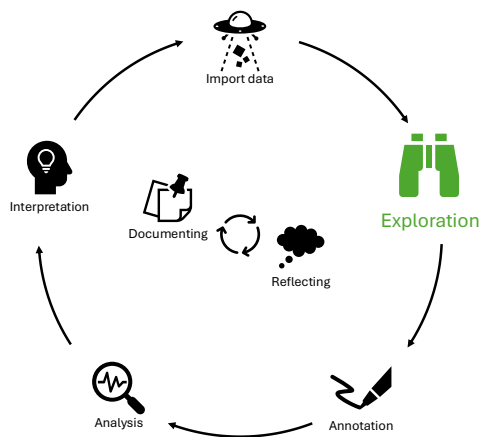
work that follows, as the extensions developed in this thesis build directly on this framework and data model.

Together with my colleagues Florian Schneider and Fynn Petersen-Frey, we have established DATS as a solid software foundation, especially for data preprocessing and complex search queries, with complete basic functionality that supports discourse analysis workflows. However, DATS' core functionality does not address all the challenges that researchers conducting discourse analysis face. Both qualitative analysis (via the annotation interface) and quantitative analysis (via word and code frequency) features are deliberately minimal. Similarly, support for the interpretation and reflection phase is minimal, relying solely on basic memo functionality. More significantly, the core features do not address the question of how to make large datasets truly manageable for qualitative analysis. Specifically, the challenges of obtaining an overview of research materials and qualitatively analyzing large volumes of data are not addressed. Furthermore, collaboration with project partners revealed a demand for more sophisticated analytical methods, such as a feature focused on discursive evolution (e.g., a timeline analysis) and a feature to synthesize and present all research findings (e.g., a whiteboard).

While the collaborative development of the DATS foundation is a contribution of this dissertation, its core contributions lie in addressing the aforementioned gaps. The following Chapters 5–8 will detail these challenges, explain the problems, and present features developed within this dissertation to evolve DATS into a more comprehensive and robust research platform for discourse analysis.

The Discourse Analysis Tool Suite provides the robust technical infrastructure and multimodal preprocessing pipelines necessary to handle large-scale digital data. However, importing and cleaning data is only the first step of the discourse analysis research process. Before a researcher can engage in fine-

grained qualitative analysis and interpretation, they must navigate the data to identify theoretically relevant materials and organize the corpus into meaningful structures. The next chapter addresses the exploratory phase of the discourse analysis workflow. We present two significant contributions to DATS—*Perspectives* and the *LLM Assistant*—which leverage advanced NLP techniques to transform large-scale, unstructured datasets into organized, navigable research corpora.



5

Exploration, Search & Organization

Contents

5.1	Motivation	96
5.1.1	Exploration	96
5.1.2	Search & Filtering	97
5.1.3	Data Organization	98
5.2	Challenges	99
5.3	Exploration, Search & Organization in DATS	101
5.3.1	Workflow with Alice	101
5.3.2	Addressed Challenges	102
5.3.3	Excursus: Configurable Similarity Search	103
5.4	Perspectives: Interactive Document Tagging in DATS	105
5.4.1	Introduction	105
5.4.2	Related Work	106
5.4.3	Workflow with Alice	108
5.4.4	Interactive Clustering	110
5.4.5	Evaluation	113
5.4.6	Integrating Perspectives in DATS	119
5.4.7	Conclusion	120
5.5	Exploring Large Language Models for Document Organization	121
5.5.1	Introduction	121
5.5.2	Related Work	122
5.5.3	Workflow with Alice	123
5.5.4	Benchmark Setup	125
5.5.5	Document Classification Evaluation	125
5.5.6	Metadata Extraction Evaluation	128
5.5.7	Integrating the LLM Assistant in DATS	131
5.5.8	Conclusion	132
5.6	Conclusion & Discussion	132

This chapter focuses on the second step of the discourse analysis workflow: exploration, search, and organization. In digital discourse analysis, this phase is critical for "*finding the needle in the haystack*" and establishing the systematic structure required for subsequent historical and comparative analysis.

We begin by discussing the motivation (Section 5.1) and challenges (Section 5.2) of exploring large-scale, heterogeneous corpora, highlighting the need for methods that support theoretical sampling and handle linguistic ambiguity. Section 5.3 reviews the baseline search and filtering capabilities of DATS, illustrated through the continuing workflow of our research persona, Alice.

The core of this chapter presents two major contributions:

1. Section 5.4 presents *Perspectives* (Fischer and Biemann 2026), an interactive document clustering extension that utilizes instruction-tuned embeddings and LLM-driven document rewriting to provide a holistic, aspect-focused overview of a corpus. We detail the development of an interactive 2D map and evaluate how few-shot fine-tuning of embedding models allows the system to adapt to a researcher's evolving intent.
2. Section 5.5 presents the *LLM Assistant* for organization, a feature that leverages the zero-shot capabilities of large language models for document tagging and metadata extraction. Based on our evaluation in Fischer and Biemann (2024), we demonstrate how LLMs can reduce the manual effort required to structure extensive document collections while maintaining high levels of user control and transparency.

By integrating these AI-augmented tools, this chapter addresses Research Questions 1, 2, and 6, demonstrating how interactive visualization and generative AI can effectively support the initial stages of the discourse analysis research circle.

5.1 Motivation

5.1.1 Exploration

In a discourse analysis project, the exploratory phase (Section 3.5, Phase 1) is important for grounding the research project. Its purpose is to develop the researcher's theoretical sensitivity and establish the basis for systematic (sub-)corpus construction.

In this phase of exploratory investigation of the research field, the aim is to develop a sense of the field to determine the subject and framework ("*What*") and the actors and fields ("*Who*") of the discourse. Exploration aims to establish initial familiarity with the social field under investigation and the organization of knowledge production. This familiarity should prevent subsequent methodological errors (e.g., arbitrary or incomplete topic selection) (Traue et al. 2022, p. 636). For example, researchers may use such questions to guide their exploration:

- What is the exact subject of the discourse (e.g., not just *COVID-19* or *pandemic management*, but the *socio-political debate surrounding mandatory vaccinations* within a specific geographical and temporal scope)?

- Which historical turning points or events have shaped the discourse? Which other discourses overlay or influence the investigated topic?
- Which thematic or geographical areas are relevant to my research question?
- Which other discourses overlay or influence my subject matter (e.g., the *civil liberties discourse* and the *public health discourse*)?
- Who are the relevant speakers and institutions (e.g., *experts, politicians, media, pharmaceutical industry, marginalized groups*)?
- In which institutional fields is the subject decided or debated (e.g., *constitutional courts, scientific advisory boards*)?

Exploratory research is not limited to textual material. It involves literature review, document analysis, and potentially participant observation. The data to be explored encompasses the full spectrum of discursive events, including texts and documents, newspaper articles, images, videos, artifacts, etc. Importantly, discourse analysis considers only "*natural data*", that is, data that already exist in the field and is "*process-generated*" (Traue et al. 2022, p. 636).

5.1.2 Search & Filtering

The need for targeted searching and filtering stems from the availability of large data volumes suitable for corpus creation.

The preliminary structure of the corpus is based on the initial exploration phase. Following the logic of theoretical sampling of Grounded Theory (Section 3.2.2), the corpus is adjusted by being expanded or restricted during the analysis (Traue et al. 2022, p. 636). Consequently, the corpus is not static, but changes dynamically throughout the research project. The constant comparative method (Section 3.2.2) guides this dynamic process to locate either homogeneous (similar) cases to articulate an object's core, or heterogeneous (different) cases to verify an object's range. For example, questions like the following may guide the construction and extension of the corpus. More guiding questions can be found in Keller (2011a, p. 92):

- Which media formats are crucial for this discourse (e.g., written reports, audiovisual media, artifacts, digital interfaces)?
- Which document types best represent the knowledge (e.g., policy papers, expert interviews, legal texts, popular news articles, social media statements)?
- Does the corpus include both dominant (e.g., government papers) and marginal (e.g., blogs) voices to analyze exclusion strategies?
- Is material collected across the entire relevant timeframe to analyze the genesis of the concepts and changes in concepts?
- Which material is typical for the discourse, and which material is extreme or disruptive, defining the limits of sayable?

Franken et al. (2020) propose a corpus reduction strategy. For extensive corpora (e.g., those generated by automated web crawling), filtering reduces the source material. An initially unknown corpus is continuously refined through automated filtering (e.g., based on pre-defined word fields or key concepts) to identify relevant sub-corpora for fine-grained analysis.

The aim of searching and filtering is not to obtain a statistically representative sample. Instead, it is to locate documents with maximum theoretical relevance for reconstructing the rules of formation (theoretical sampling, Section 3.2.2).

Specific objectives include identifying breaks and discontinuities by searching for statements outside the dominant knowledge (radical or disruptive utterances) or for correlations between concepts (Traue et al. 2022, p. 634).

Another objective is identifying rare actors. Filtering for marginalized discourse producers enables the analysis of the relationship between knowledge and power (e.g., who is legitimate to speak) (Traue et al. 2022, pp. 629,641).

Finally, an interest in comparative analysis is to locate documents that reveal differences between orders of knowledge or material that indicate correspondences between concepts (Traue et al. 2022, p. 628).

5.1.3 Data Organization

Data organization is a methodologically informed step in discourse analysis that enables and streamlines the analytical work across all phases. The systematic arrangement of documents is already interpretive and preconditions the discourse researcher's subsequent analytical performance.

Organizing and preparing the data cleverly supports the two analytical, interpretative methods (Section 3.4.2): (a) The *historical method* requires data to be sorted by time periods to track the evolution of concepts and arguments. (b) The *comparative method* aims to analyze differences between orders and, hence, necessitates the separation of data by orders (e.g., institutions, actor groups) to investigate the variability in how the same topic is discussed (Traue et al. 2022, p. 628).

Organization can be achieved through folder structures, metadata assignments, or document tagging and classification. These categories can be derived from the interpretative goals of discourse analysis (Section 3.4.2), going beyond purely descriptive metadata. For example, documents could be organized and categorized along the following dimensions. Additional guiding questions can be found in Keller (2011a, p. 100):

- **Time Period:** Capturing the time frame of publication or utterance is essential for the *historical method*, enabling the reconstruction of the historicity of concepts and arguments.
- **Source:** Specifying the source (e.g., academic journal, political party, mass media) supports the *comparative method* by facilitating the analysis of differences between institutional or media discourses.
- **Actor Position:** Defining the speaking actors (e.g., expert, decision maker, affected individual) is fundamental for analyzing speaker positions and the legitimacy of knowledge within the discourse.

- **Subject Type:** Capturing the specific facts or sub-topics addressed in the document (e.g., regulation, impact assessment, public relations) supports the verification of the discursive formation.
- **Function/Strategy:** Analyzing the intent behind the statement (e.g., legitimization, critique, delegitimization, problem formulation, solution) supports subsequent analysis of strategies, function, and power by making actor strategies visible.

Discussion

From the perspective of natural language processing, the methodological requirements of discourse analysis for exploration, search, and organization present noteworthy distinctions from typical NLP projects.

Standard NLP projects often prioritize statistical representativeness and the identification of high-frequency patterns to ensure that models generalize well across broad, unseen datasets. In this context, fixed and balanced corpora are essential for minimizing bias and ensuring the reliability of performance metrics. In contrast, the corpus construction logic in DA is not primarily focused on obtaining a statistically representative sample or identifying the most frequent patterns. Instead, it targets documents with the most significant theoretical relevance, even if they are numerically rare (e.g., a single influential policy paper). This dynamic corpus adjustment (theoretical sampling and the constant comparative method) contrasts with the often stable, balanced datasets used in NLP experiments and benchmarks.

Further, in typical NLP workflows, categories (used for document classification) are frequently designed to be mutually exclusive and descriptive (e.g., spam vs. non-spam or sentiment polarity) to facilitate high inter-annotator agreement and clear machine-learning objectives. This simplicity is a functional requirement for achieving scalable and repeatable automated classification. However, categories in DA are often interpretively charged and theoretically guided by analytical concepts (e.g., categories like legitimization or critique). Rather than providing neutral ordering, these categories serve as hypotheses to be investigated in the subsequent fine-grained analysis. Documents may be organized according to criteria such as *strategies*, *intentions*, *belonging to an order*, or *speaker position*, which requires complex, interpretative work.

5.2 Challenges

Discourse analysis involves demanding methodological steps, such as iteratively searching and organizing documents. When dealing with extensive corpora, these steps become particularly challenging. This section presents several challenges that need to be addressed to improve research efficiency, motivating the development of digital discourse analysis tools.

Managing Large Data Volumes: The available amounts of data that can be used to analyze a discourse are ever-increasing. Automated data access via

platforms, crawls, or APIs enables the creation of corpora of immense scale that researchers cannot thoroughly review manually. A particular challenge is establishing a systematic overview of this initially unknown corpus and identifying documents with theoretical relevance—the challenge of *"finding the needle in the haystack"*. As noted by Franken et al. (2020), who highlighted the potential of automated corpus creation methods when integrated with powerful filtering and search functions, social reality is complex and requires methodologically guided reduction to manageable segments via filtering. This requires indexing massive datasets to make extensive collections of multimodal documents highly searchable and navigable, thereby facilitating the selection of relevant content for detailed qualitative analysis. Developing access methods for large corpora – such as generic search, topic modeling, or sentiment analysis – is a prerequisite for corpus reduction and isolating theoretically relevant sub-corpora.

Data Heterogeneity: An equally challenging aspect is managing data heterogeneity. Discourse analysis corpora often comprise diverse data types (texts, documents, articles, images, videos, etc.). Handling and systematically incorporating this heterogeneity can present a challenge to both the researcher's methodology and digital tools.

Linguistic Ambiguity: Further difficulty arises from linguistic ambiguity. Since language is inherently ambiguous and highly context-dependent, precise search procedures are complicated. The meaning of a term depends on its specific discursive context, which quickly limits the effectiveness of algorithmic search strategies based on isolated keywords. Handling this context-dependent nature presents another challenge for digital tools. For example, a keyword search for *"virus"* may simultaneously retrieve results about malicious software (a computer virus) and a medical infection (the COVID-19 virus), requiring contextual differentiation that simple keyword matching cannot provide.

Theoretical Sampling: Challenges in search and document management also arise from the Grounded Theory methodology used in DA (Section 3.2). Identifying cases with minimal contrast (homogeneous cases for concept refinement) and maximal contrast (deviant/heterogeneous cases for scope verification) is a demanding process. Identifying these theoretically relevant cases is a persistent challenge throughout the research process.

Flexible Categories: Finally, unlike standardized methods, discourse analysis lacks a predefined set of analytical and descriptive categories. The analytical units and descriptive levels (e.g., organizational categories) must be developed flexibly from the empirical data, with specific reference to the subject matter. This prevents the a priori assignment of a standard organizational structure, requiring digital tools to offer high flexibility in category and folder management.

5.3 Exploration, Search & Organization in DATS

This section illustrates how to use the base exploration, search, and organization features in DATS. The technical implementation of these features is explained in Section 4.3. We will continue to follow Alice as she guides us through her research workflow.

5.3.1 Workflow with Alice

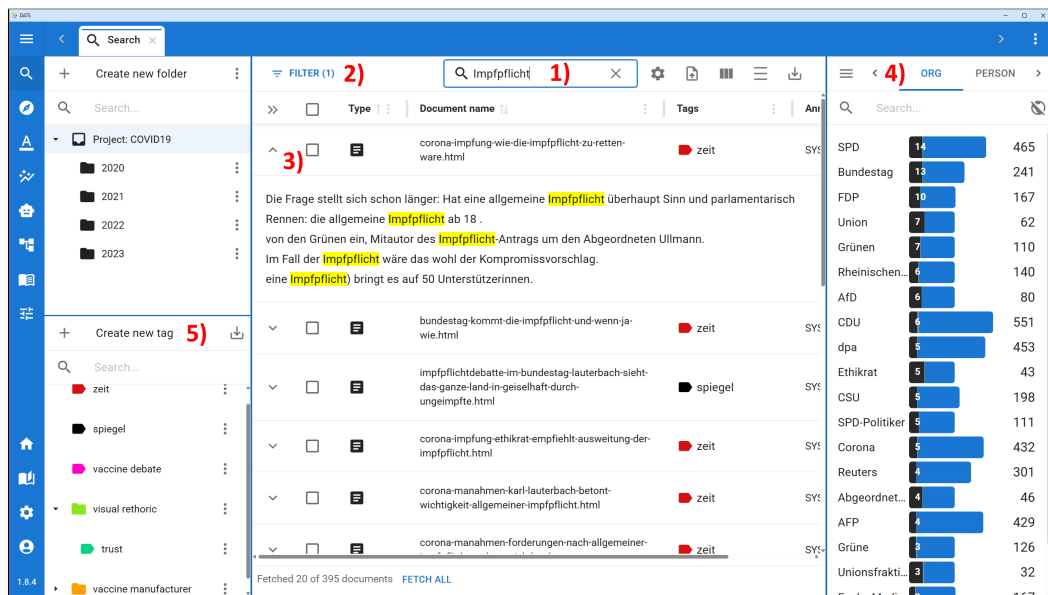


Figure 5.1: DATS search & filtering interface. It combines the search bar (1) and advanced filter menu (2) with the result table (3) and dynamic contextual statistics (4). Result rows can be expanded to show mentions in context. The tag and folder systems (5) allow saving relevant sub-corpora.

Alice has successfully imported and preprocessed her news article corpus (Section 4.7). She now begins the exploration phase to reduce her corpus to relevant sub-corpora needed for in-depth comparative analysis.

Alice uses the combined search and filtering interface (see Figure 5.1) to identify relevant documents based on her research interests: She enters queries such as "Impfpflicht" AND "AstraZeneca" into the search bar (1) to isolate the heated German debate surrounding compulsory vaccination and the side effects of a specific vaccine. She then uses the advanced filter menu (2) to narrow her search further by metadata fields, focusing only on articles published between January and June 2021 (when the debate was most intense) and filtering for specific sources (e.g., *Der Spiegel* or *Die Zeit*).

The central result table lists the reduced set of documents. Result rows can be expanded to show mentions of the query in short context windows (3). Simultaneously, the right sidebar dynamically updates with contextual statistics (4). Alice reviews the aggregated statistics of the most frequent named entities and keywords to confirm that the most frequently mentioned entities are indeed the relevant actors (e.g., the health minister or members of the permanent vaccination

commission ("Ständige Impfkommision")), validating that her filter successfully captured the core debate. Double-clicking a document opens it in read mode for fine-grained analysis. When Alice selects a document, an information panel replaces the contextual statistics on the right, allowing her to view and edit its metadata.

Satisfied with the precision of her query, Alice uses the tag system in the left sidebar (5) to apply the tag "Vaccine Debate 2021 Q2" to the entire result set, effectively saving this relevant sub-corpus for later use.

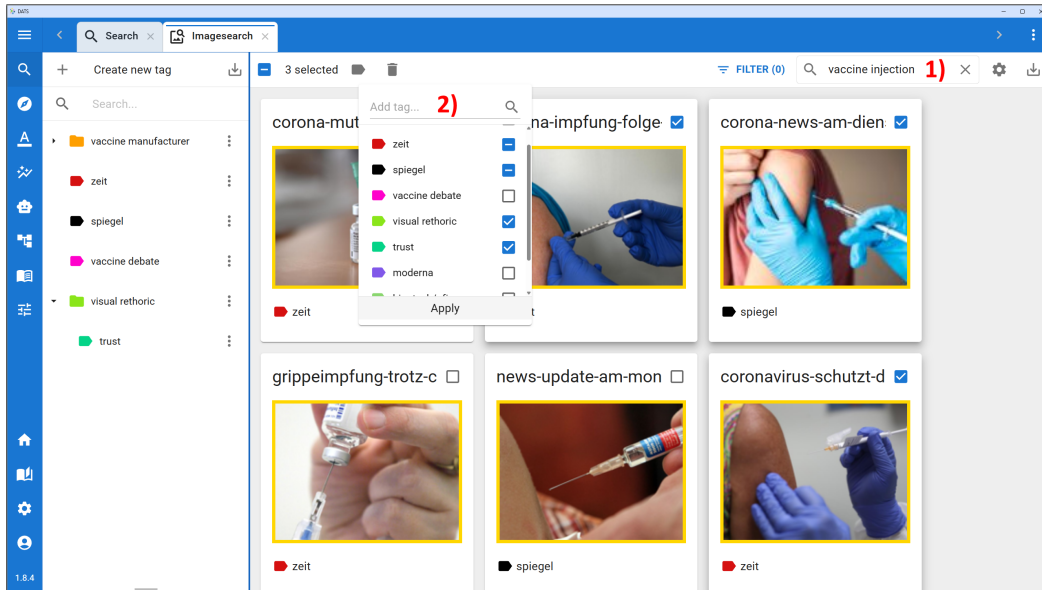


Figure 5.2: DATS multimodal semantic search interface. A natural language query (1) retrieves semantically similar images. The tag system (2) allows saving relevant results.

For more nuanced exploration that accounts for linguistic ambiguity, Alice uses the multimodal semantic search (see Figure 5.2). This dedicated view, accessible through the navigation panel, allows Alice to input any query (image or text) to find similar items. She enters a short phrase, such as "trust in vaccination", into the search bar (1). The system retrieves all text passages (sentences) that are semantically similar to this concept, even if they do not contain the exact phrase. Since documents from all modalities are represented in the same embedding space after pre-processing, the semantic search retrieves similar documents regardless of modality. Hence, DATS can also retrieve visually similar images, such as "injections of vaccines", which could be relevant for analyzing the visual rhetoric of public health campaigns. This feature aids her methodologically guided search for visually homogeneous cases. Alice can then apply a tag to selected images (2), such as "Visual Rhetoric: Trust", to mark these documents, saving them in a custom, user-defined collection for later analysis.

5.3.2 Addressed Challenges

The DATS features regarding exploration, search, and organization directly address several key methodological challenges identified in Section 5.2.

Firstly, they tackle the challenge of *managing large data volumes*: the implemented search and filtering procedures directly address this scalability challenge by enabling methodologically guided reduction of large corpora to manageable, theoretically relevant subsets. This is achieved through preprocessing extensive, multimodal data, enriching the documents with as much relevant information as possible. Documents are automatically annotated and indexed with diverse information, including metadata, statistical features, and NLP pipeline outputs such as keywords and named entities (Section 4.6.3). This enrichment serves two key functions: (1) It enables researchers to define complex queries using a combination of filters, statistics, and search terms. (2) It facilitates the discovery of interesting documents (e.g., documents featuring unexpected people or organizations) through statistical aggregation and visualizations. Consequently, DATS provides a robust technical foundation for substantial reduction of large, so far unmanageable corpora, preparing them for interpretative-hermeneutic analysis.

Secondly, DATS provides fundamental organizational features that allow documents to be manually structured using flexible tags, folders, and editable metadata – a capability crucial for developing flexible structures from empirical data.

Thirdly, the system’s design manages the challenge of *data heterogeneity* and *linguistic ambiguity*. Data heterogeneity is addressed primarily during preprocessing, where a unified pipeline converts all input modalities (text, images, etc.) into a text representation that enables retrieval via search and filtering. Further, the semantic search leverages text and image embeddings to support diverse modalities. Concurrently, the ambiguity inherent in language is mitigated by semantic search functionality, which relies on contextualized embeddings that capture polysemy and context dependence more effectively than traditional keyword-based search strategies.

Finally, semantic search helps users find “*more of the same*”, partly addressing the challenge of *theoretical sampling* by supporting the requirement of minimal contrasting (e.g., by finding homogeneous cases for concept refinement).

5.3.3 Excursus: Configurable Similarity Search

The challenge of identifying maximal contrasting cases (deviant or heterogeneous documents) remains complex, which motivated our development of the concept of configurable similarity search in parallel work (Hatzel et al. 2023).

Understanding why two documents (or a query and a document) are considered similar is non-trivial. While contextualized embeddings intuitively group paraphrases by semantic proximity, the interpretive rationale for why a specific document was retrieved often remains opaque. In applications, this lack of insight into the type of similarity (e.g., topic vs. style) hinders methodologically guided selection.

In this work, we approached this by modeling various dimensions of similarity, providing the user with insight into which specific aspects of the documents are similar. This was achieved by training sentence-similarity models on the SemEval 2022 Task 8 dataset (Chen et al. 2022) to capture distinct aspects of similarity (e.g., shared entities, similar style, or similar tone). These models were then evaluated on downstream retrieval tasks using weighted combinations of the dimensions. This research culminated in the conceptual introduction of an application that allows

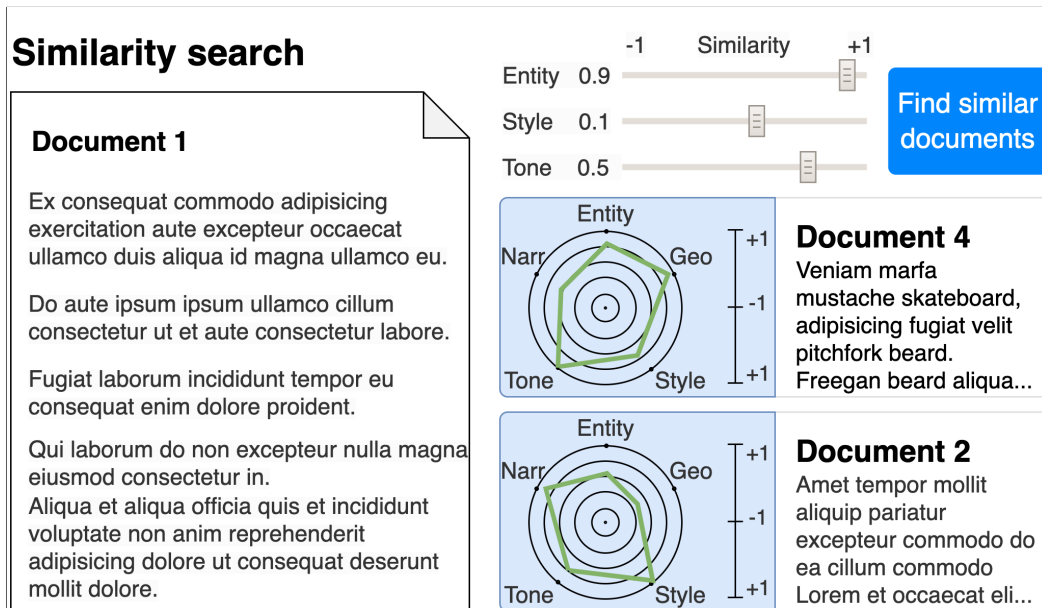


Figure 5.3: Configurable similarity search mockup. Left: current document. Right: configurable search and results that visualize its similarity across the various dimensions. Image from Hatzel et al. (2023).

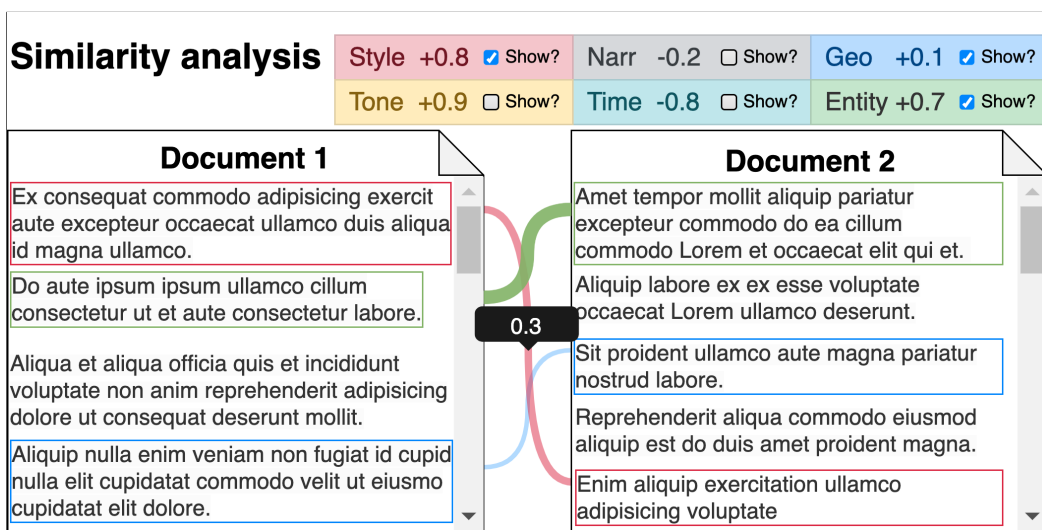


Figure 5.4: Document similarity analysis mockup. The similarity between two documents is broken down into sentence-wise, dimension-based similarity. Image from Hatzel et al. (2023).

for configurable similarity search (see Figure 5.3). This system should enable users to focus the search on specific aspects, for example, retrieving documents that cover the same topic but from a different perspective (as reflected in the style or tone). This system should also enable users to gain detailed insights into the similarities between two documents (see Figure 5.4).

This concept of configurable similarity search has interesting implications for theoretical sampling. Such a system could support the methodology of maximal contrasting by identifying relevant deviant cases. Researchers could precisely define the dimensions along which the documents should differ.

5.4 Perspectives: Interactive Document Tagging in DATS

5.4.1 Introduction

While DATS' core functionalities include search and filtering capabilities, users managing extensive corpora continue to face challenges with systematic exploration and organization. Traditional keyword-based or semantic search often proves insufficient and incompatible with the hermeneutic approach required for qualitative conceptualization. DATS core features cannot provide a systematic structure and holistic overview, yet document organization is vital for qualitative analysis and a prerequisite for advanced features such as timeline analysis.

Document clustering offers a promising solution, particularly for discovering emergent themes within a corpus, a key analytical interest for digital humanities scholars. Modern clustering-based topic modeling methods, such as BERTopic or GPTopic (Grootendorst 2022; Angelov and Inkpen 2024; Reuter et al. 2024), leverage pre-trained language model embeddings and have gained considerable traction due to their advantages in simplicity, modularity, and performance.

However, a limitation of fully automatic, unsupervised clustering methods is the risk of a "*take it or leave it*" outcome. If the clustering outputs do not align with the researcher's specific, evolving research questions or theoretical intent, the automated results are rendered ineffective for non-experts. The interactive topic modeling community (details in Section 5.4.2) addresses this gap by aiming to achieve a controllable, collaborative human-machine process that allows domain experts with no particular technical knowledge in machine learning to steer the model.

To address this gap, we introduce *Perspectives*, an interactive document clustering tool for DATS. This section is based on Fischer and Biemann (2026), who first detailed the extension's development and conceptual framework. Built on a clustering-based neural topic modeling pipeline, *Perspectives* is centered on an interactive document map designed to support exploratory workflows. While offering visualization and analysis tools, its primary goal within the DATS ecosystem is to help users systematically organize their materials into tagged datasets, thereby preparing the corpus for use with other analytical tools.

The development of *Perspectives* addresses two research questions of this thesis:

Research Question 1

How can vector-based document representations be leveraged to provide researchers with a flexible, interactive visualization of large-scale text corpora that enables initial explorative structuring and classification?

Research Question 6

How can AI-augmented methods—including visualization, annotation, and semantic search—be effectively designed within a human-in-the-loop framework to incrementally and rapidly adapt to evolving concepts and domain-specific inquiries in qualitative discourse analysis projects despite the scarcity of labeled data?

The underlying task tackled by this work is text clustering (Section 2.1.8). Typically, documents within the same cluster are more semantically similar to each other than to documents in different clusters, making it a fitting task for topic discovery, data organization, and exploration of large text corpora.

The contributions are:

1. We introduce a flexible, aspect-focused document clustering pipeline that combines (a) optional LLM-driven document rewriting to emphasize user-defined aspects, (b) instruction-finetuned embedding models (Su et al. 2023) (Section 2.5.4) to generate aspect-oriented document representations, and (c) few-shot fine-tuning of the underlying sentence transformer embedding model (Tunstall et al. 2022) to further align the representations with the scholar’s evolving analytical goals based on minimal user feedback.
2. We evaluate the pipeline on multiple datasets, showing the effectiveness of aspect-oriented representations, document rewriting, and fine-tuning.
3. We present *Perspectives*, an interactive human-in-the-loop document clustering system that integrates iterative model refinement into the pipeline. *Perspectives* provides an intuitive interface consisting of a dashboard of statistics for a quick overview and an interactive 2D map for users to explore, validate, and directly manipulate clusters with established refinement operations.
4. We demonstrate the practical utility and integration of *Perspectives* for digital humanities research by detailing an envisioned workflow within DATS. The workflow illustrates how digital humanities (DH) scholars can leverage the system’s interactive and steerable capabilities to explore corpora and prepare their data for subsequent in-depth qualitative and quantitative analyses.

5.4.2 Related Work

Clustering-based Neural Topic Modeling: The task of identifying topics in texts has evolved from traditional probabilistic models, such as Latent Dirichlet Allocation (Blei et al. 2003). Recent approaches, termed clustering-based neural topic modeling or topic discovery (Wu et al. 2024), build on pre-trained language model embeddings. Top2Vec (Angelov 2020) uses Doc2Vec (Le and Mikolov 2014) embeddings, reduces their dimensionality with UMAP (McInnes et al. 2020), clusters them using HDBSCAN (Campello et al. 2013), and identifies topics by finding words closest to cluster centroids. BERTopic (Grootendorst 2022) follows a similar pipeline but employs a class-based TF-IDF mechanism to extract descriptive

words for each identified cluster. Contextual Top2Vec (Angelov and Inkpen 2024) refines this by representing documents with multiple segment vectors for fine-grained topic segmentation. *Perspectives* adopts a general pipeline of embedding, dimensionality reduction, clustering, and representation to identify meaningful clusters, leveraging the flexibility and strong performance of these methods.

Guided/Interactive Topic Modeling: The topic modeling community has long recognized the need to make topic modeling more user-centric. Seed wordlists (Jagarlamudi et al. 2012), seed documents (Grootendorst 2022), or seed sentences (Fang et al. 2023) can guide initial topic formation. Further, interactive topic modeling systems address alignment with user intent by providing operations such as merging, splitting, or deleting topics, as well as adding or removing documents/words from specific topics (Fang et al. 2023). Seelman et al. (2024) proposes to update topic models by embedding user-assigned word labels to topics. GPTopic (Reuter et al. 2024) leverages LLMs to build an accessible chat interface to explore and refine topics through natural language.

Semi-supervised Clustering: The field of interactive (semi-supervised) clustering also acknowledges the value of integrating expert knowledge. While prior methods, such as hand-picked seed points (Basu et al. 2002) or pairwise constraints (Basu et al. 2002), offered control, they required extensive user feedback. Viswanathan et al. (2024) demonstrated that LLMs can guide semi-supervised clustering with minimal human input by generating and encoding task-specific keyphrases for document representation. In this work, we evaluate the effectiveness of task-specific keyphrases and summaries for adaptable, user-guided document representations.

Related Tools: Several systems facilitate the interactive exploration of document collections or topics. Nomic Atlas¹ is a notable commercial platform that offers high-performance 2D scatter-plot visualizations of document embeddings, enabling users to explore automatically extracted hierarchical topic clusters and interact with a chat assistant and filters. Related academic systems innovate with user-centric approaches, such as direct integration into the labeling workflow (Seelman et al. 2024), tracking model changes (Fang et al. 2023), or retrieval-augmented generation (RAG) based question answering (Reuter et al. 2024).

Visualization: Effective visualization helps explore and understand clusters. Standard techniques, such as top words, word clouds, document-cluster distributions, and 2D embedding projections (Wu et al. 2024), are all available in our extension.

Perspectives is built on the robust pipeline of clustering-based neural topic modeling. It integrates steering mechanisms inspired by guided topic modeling and semi-supervised clustering, such as LLM-driven document rewriting, instruction-tuned embedding models, and fine-tuning, to focus the analysis on user-defined

1. <https://atlas.nomic.ai>

aspects. Furthermore, *Perspectives* incorporates established refinement operations that draw on interactive topic modeling, enabling users to manipulate clusters interactively. While inspired by existing exploratory platforms, such as Nomic Atlas, it distinguishes itself by offering a user-centric human-in-the-loop (HITL) workflow tailored for DH: Users define the analytical lens before clustering and iteratively refine the outcomes afterward, with visualizations serving as a central component of this interactive sense-making process.

5.4.3 Workflow with Alice

This section illustrates an idealized workflow using *Perspectives*. The technical implementation is explained in the following sections. Alice will continue to guide us through her research workflow.

After exploring her corpus with DATS search and filtering tools (Section 5.3), Alice still lacks a clear overview of her data. Next, Alice wants to understand how topics and sentiments differed between *Der Spiegel* and *Die Zeit*. To do this, she uses the interactive clustering tool *Perspectives* to explore and categorize her corpus, preparing it for the later comparative analysis.

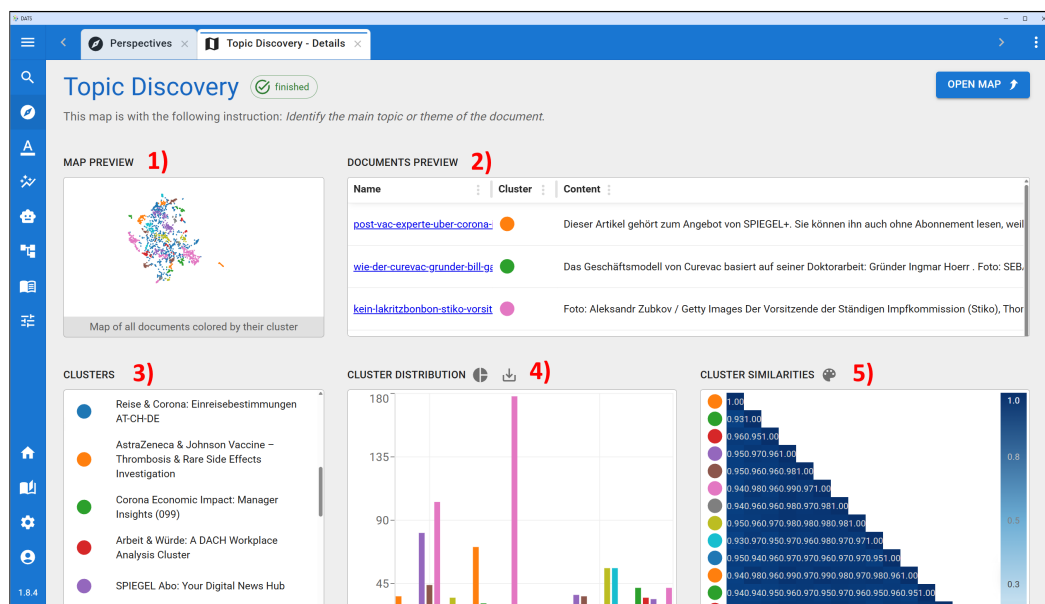


Figure 5.5: The *Perspectives* dashboard. It provides an initial, high-level overview of the document clustering. It displays the map preview (1), a list of documents (2), the identified clusters (3), cluster frequencies (4), and a similarity matrix (5) for inter-cluster relationships.

Alice opens the *Perspectives* extension and initiates the analysis. To ensure the clustering focuses on topic discovery relevant to her discourse analysis, she provides a custom instruction (e.g., "*Summarize this news article with a focus on the COVID-19 pandemic*") to guide document embeddings. The system generates a dashboard (see Figure 5.5) that provides a high-level overview: a map preview (1), a document preview (2), a list of identified clusters/topics (3), cluster frequencies

(4), and a similarity matrix showing semantic relationships between the identified clusters (5).

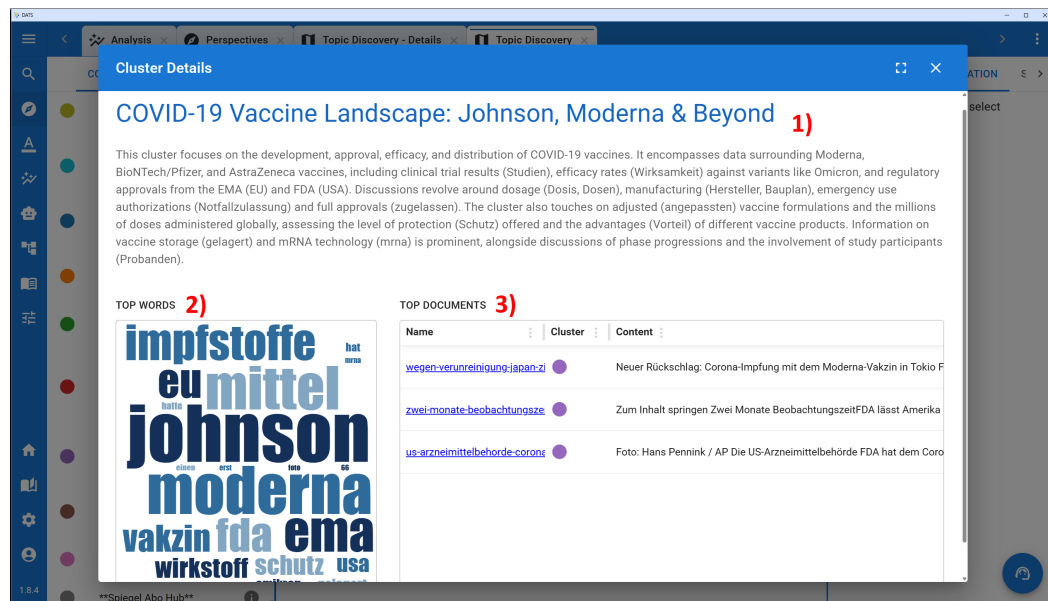


Figure 5.6: The *Perspectives* cluster detail dialog. It provides interpretability for a discovered topic, featuring an LLM-generated description (1), a word cloud (2) of descriptive keywords, and the top representative documents (3).

To interpret these initially unsupervised clusters, Alice clicks on a topic to open a dialog that includes further details (see Figure 5.6). She carefully reviews the LLM-generated title and description (1), the word cloud of most relevant keywords (2), and the top representative documents (3) to understand the semantic coherence of the cluster. She confirms broad topics like *"Economic Relief Measures"* and *"Lockdown Policy Debate"*. However, she notes that the highly relevant thematic area *"Vaccine Hesitancy"* is missing.

Alice then moves to the interactive *Perspectives* map (see Figure 5.7) to begin refinement. She navigates the 2D scatter plot (1), panning and zooming, and inspects the visually distinct clusters to identify documents related to the missing theme. She uses the integrated search (2) to locate relevant articles on vaccine skepticism, then selects them using the selection tools (3). Linked statistic and information panels (4) provide aggregated details, such as frequent keywords and named entities.

To explicitly capture her expected *"Vaccine Hesitancy"* theme, she uses the **add cluster** function (4) on her selection. If another cluster seems too broad (e.g., general *"Policy Debate"*), she uses **split cluster** (5) to compute granular sub-clusters. Conversely, she can **merge clusters** if several clusters represent a single cohesive theme (5).

With all topics identified, Alice improves their definitions by reviewing document-cluster assignments. Alice corrects misclassified articles using the **change cluster** function (6) and confirms assignments with **accept cluster assignment** (7). This minimal feedback (her accepted examples) is used to create a contrastive dataset. She then triggers the **refine model** process to fine-tune the underlying embedding

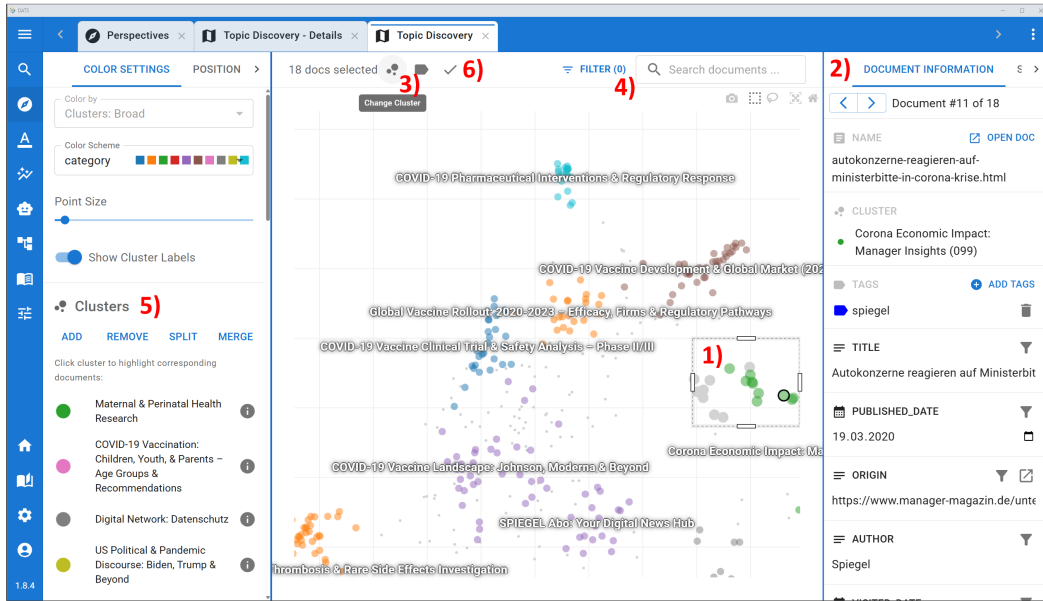


Figure 5.7: The *Perspectives* interactive 2D document map. Center: interactive scatter plot of documents colored by cluster. Hovering over a document previews its content. Left: settings & refinement operations, including the model fine-tuning operation. Right: statistics & information about selected documents. Top: toolbar with search & filtering. The user interface is inspired by popular clustering interfaces, such as Nomic Atlas, and leverages a familiar design to accelerate adoption.

model. The map reloads, showing improved clusters that more distinctly separate topics according to her analytical intention.

The power of *Perspectives* lies in its adaptability. Satisfied with the topic clustering, Alice can easily repeat this entire process to focus on sentiment analysis. Therefore, she creates a new perspective by providing a sentiment analysis instruction (e.g., *"Summarize the emotional tone of this article regarding COVID-19 policy"*). This generates a new dashboard and map visualizing sentiment-based clusters, which she can explore and refine using the same interactive tools.

Throughout her exploration, Alice can leverage an LLM-powered RAG system. By selecting a cluster or a set of documents, she can ask questions (e.g., *What are the main arguments in this cluster?*) to gain deeper insights.

Finally, Alice exports her refined topics (e.g., *"Economic Relief"*, *"Vaccine Hesitancy"*) and sentiment categories from *Perspectives* as analytical tags back into her main project. This structured output enables Alice to later compare the two news outlets based on topic distribution.

5.4.4 Interactive Clustering

As illustrated by the workflow, *Perspectives* facilitates interactive document clustering, empowering users to guide analysis towards specific aspects and iteratively refine categorizations. This process begins with an aspect-focused clustering pipeline and continues through HITL refinement operations. The pipeline is illustrated in Figure 5.8.

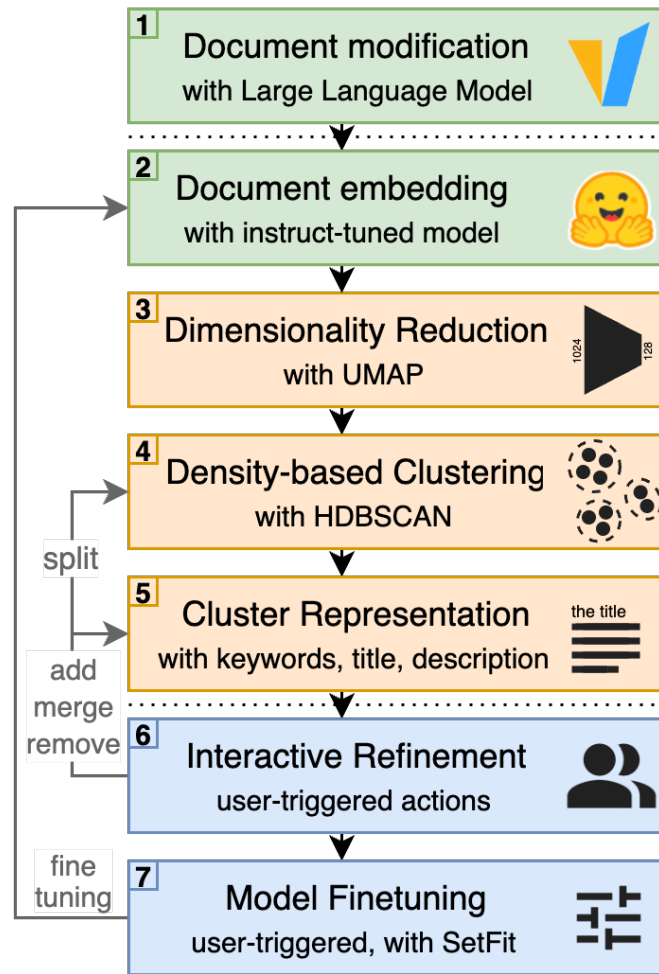


Figure 5.8: *Perspectives* interactive clustering pipeline. The initial clustering process is guided by providing rewriting and embedding instructions (green) to focus the document representations on user-defined aspects. The established core pipeline (orange) identifies clusters and builds various textual representations. Users can post-process the clustering (blue) using refinement operations (grey), which trigger some steps of the pipeline. Figure from Fischer and Biemann (2026).

We propose two strategies to define the analytical lens and initially steer the clustering process (green). First, we employ alternative document representations, in which an LLM rewrites documents based on a prompt (e.g., *"Summarize the document, focusing on the main topic"*), thereby modifying the content for use in clustering. This is a one-time process per perspective and results in a modified document collection. Second, we use instruction-finetuned embedding models, which generate aspect-oriented embeddings based on an instruction (e.g., *"Identify the topic of the document"*), capturing specific semantic facets. This results in a set of document embeddings, one for each document in the modified document collection.

We continue with established steps (orange): reduce dimensionality with UMAP to improve clustering and visualization, then use HDBSCAN to group documents and identify outliers. Cluster representations include keywords from c-TF-IDF, which are then used to seed an LLM to generate descriptive names and descriptions. Additionally, each cluster is represented by the mean of its

document embeddings, and its most representative documents are identified via cosine similarity to this centroid.

With this, the first pass of the clustering pipeline is complete: document embeddings (reduced to 2D using UMAP) are presented to the user as colored dots on a map. Next, users can post-process the clustering with established HITL refinement capabilities (blue), allowing iterative manipulation of clusters via several operations (grey arrows). These refinement capabilities are the core of *Perspectives*:

Change Cluster: Documents can be reassigned to a different cluster or marked as an outlier; representations of affected clusters are recomputed.

Add Cluster (Text): A provided name & description is embedded. Documents more similar to this embedding than their current cluster are reassigned; affected cluster representations are recomputed.

Add Cluster (Documents): Users select documents on the map to form a new cluster. Its representation is computed from these documents, and the representations of affected clusters are recomputed.

Merge Clusters: Documents from two clusters are combined to form a new one, and its representation is computed.

Split Cluster: The clustering pipeline is applied on a cluster’s documents, resulting in granular sub-clusters. The initial cluster is removed.

Remove Cluster: Deletes a cluster and reassigns its documents to their most similar cluster based on document-cluster cosine similarities.

In addition to these established operations, we propose embedding model fine-tuning to further align the underlying document representations with user intent. It is inspired by the findings of Thielmann et al. (2024). They demonstrated that with few (1-5) labeled documents per class, notable improvements in topic creation and coherence can be achieved. *Perspectives* operationalizes this by allowing users to **accept cluster assignments**. These assignments, serving as labeled examples, are used to create a contrastive dataset to fine-tune the underlying embedding model with the SetFit² library. Targeted fine-tuning aims to adjust the embedding space, pulling similar documents closer and pushing dissimilar ones apart (details are explained in Section 5.4.5). After fine-tuning, the entire dataset is re-embedded and re-clustered, resulting in an updated map that reflects the improved cluster structure.

HDBSCAN inherently identifies documents that do not belong to a cluster. *Perspectives* visualizes outliers as distinctively colored (small, grey) points on the map. They are considered during refinement operations. For instance, when a

2. <https://huggingface.co/docs/setfit>

new cluster is added, outliers can be assigned to it if they are sufficiently similar. Explicitly marking a document as an outlier excludes it from all computations.

5.4.5 Evaluation

We evaluate the effectiveness of our additions to the established clustering pipeline, which is designed to be steerable towards specific aspects. We analyze initial guidance through (1) document rewrites and (2) instruction-tuned embeddings, as well as our proposed refinement operation of (3) model fine-tuning with few data, including the choice of embedding model, the number of labeled examples required, and which instructions or prompts are effective.

KNN Accuracy Metric

The goal of *Perspectives* is to assist users in obtaining a solid document categorization using the 2D document map. While topic modeling inspired our work, we refer to the extensive evaluation by Angelov and Inkpen (2024), which demonstrated the superiority of clustering-based topic modeling approaches over traditional and neural methods on typical topic modeling evaluation metrics. Instead, we evaluate KNN accuracy, a standard metric for clustering interfaces. A good visualization should group documents with the same label and yield high classification accuracy within the visualization space (Pham and Le 2021). A k-nearest neighbors (KNN) classifier is trained to classify texts based on their 2D coordinates, and accuracy is reported. It mimics user behavior by deriving classes from dense clusters and can be seen as a proxy for end-to-end evaluation.

Datasets

To evaluate the effectiveness of steering and guidance, we selected various datasets that cover text materials (news articles, books, social media, and songs) and tasks (topic discovery, stance detection, frame analysis, and bias detection) similar to real-world research projects conducted by our project partners who actively use DATS.

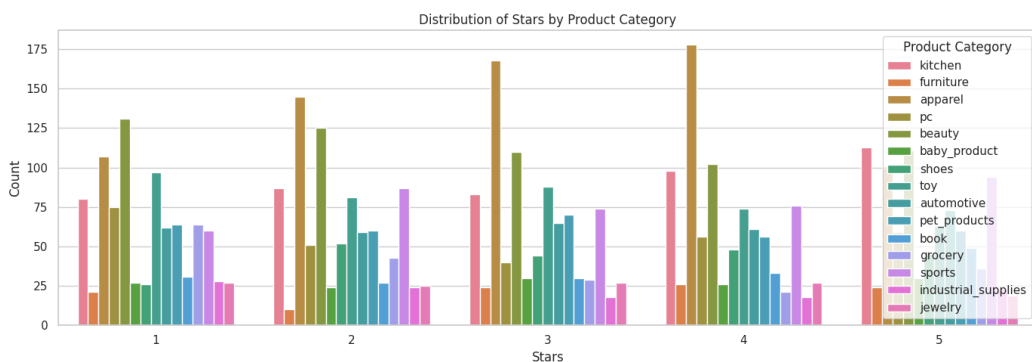


Figure 5.9: *Perspectives* experiments – distribution of stars and product categories in the Amazon test set. Figure from Fischer and Biemann (2026).

Amazon Product Reviews: A dataset of more than one million Amazon customer reviews³ that are organized by product category (*product*) and star rating (*stars*). The star rating is frequently used as a proxy for sentiment. We use the English split and consider reviews of the 15 most frequent categories. We sample 500 reviews per star for the test set. So it consists of 2,500 sampled reviews of 15 *product* categories. Figure 5.9 shows the distribution of categories and stars.

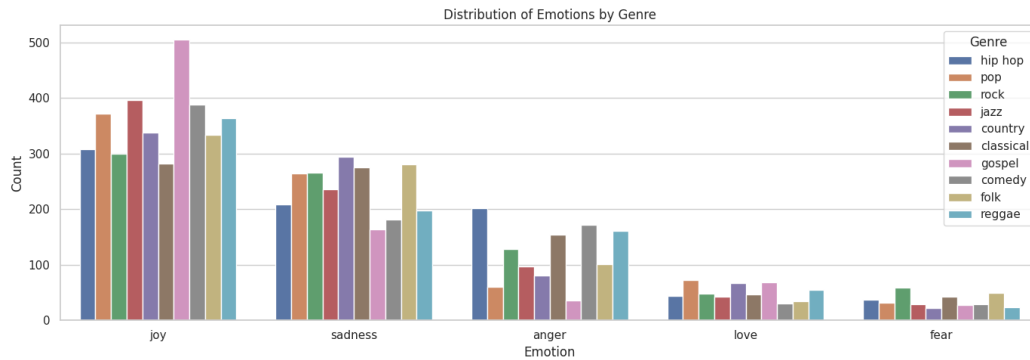


Figure 5.10: *Perspectives* experiments – distribution of emotions and genres in the Spotify test set. Figure from Fischer and Biemann (2026).

Spotify Songtexts: A dataset of approximately 500,000 song lyrics⁴ that are categorized by *emotion* and *genre*. We consider the top 10 genres and top 5 emotions for English song texts. We sample 500 song texts per genre for the test set. So it comprises 5,000 sampled song lyrics with 10 *genres* and 5 *emotions*. Figure 5.10 shows the distribution of genres and emotions.

20 Newsgroups: The 20 newsgroups dataset⁵ contains around 20,000 newsgroup posts across 20 topics. It is commonly used to benchmark topic modeling. We use 20% of the articles as a test set. The topics include, for example, *electronics*, *space*, and *politics*, and are distributed equally.

GVFC: The original Gun Violence Frame Corpus (S Liu et al. 2019) is a dataset of news article headlines, annotated into nine different perspectives/frames. We use an extension of the dataset, which includes three-sentence summaries of 1300 articles. The frames are *gun/2nd amendment rights*, *gun control/regulation*, *politics*, *mental health*, *school or public space safety*, *race/ethnicity*, *public opinion*, *society/culture*, and *economic consequences*.

Blurbs: The German blurbs dataset (Remus et al. 2019) was introduced for a hierarchical classification task at GermEval 2019. It consists of about 20k *blurbs* categorized into eight main genres, each with its own fine-grained sub-genres. We sample 1200 *blurbs* for testing. The main genres are *Literatur & Unterhaltung* (literature & entertainment), *Sachbuch* (non-fiction), *Kinderbuch & Jugendbuch*

3. <https://www.kaggle.com/datasets/mexwell/amazon-reviews-multi>

4. <https://www.kaggle.com/datasets/devdope/900k-spotify>

5. <http://qwone.com/~jason/20Newsgroups/>

(children & young adults), *Ratgeber* (guide), *ganzheitliches Bewusstsein* (holistic awareness), *Glaube & Ethik* (faith & ethics), *Architektur & Garten* (architecture & garden), and *Künste* (arts). In our experiments, we only consider these main genres.

Israel-Palestine: This stance detection dataset (Ali et al. 2025) comprises about 10k Reddit comments related to the Israel-Palestine conflict. The comments are categorized into three stances: *pro-Israel*, *pro-Palestine*, and *neutral*.

News Bias: This news dataset⁶ consists of 8.5k news events retrieved from AllSides. Organized by topic, it provides three *biases* (a categorization of each source’s political perspective: *left*, *center*, or *right*) for each news event. We sample 10k documents from the top 8 topics for testing.

Models

In alignment with the DATS model philosophy (Section 4.5), we use resource-efficient, locally running models to ensure data privacy and deployment in resource-constrained environments. For this experiment, we selected Gemma 3 (27B-instruct) (G Team 2025) for LLM-driven document rewriting, balancing performance with local resource constraints. It is a *lightweight* model from Google trained on 14T tokens, with a large 128k context window, and supports over 140 languages. More details are provided in Section 2.6. We choose *multilingual-e5-large-instruct* (500M) (L Wang et al. 2024) for the instruction-tuned embedding model. It is one of the best-performing models on the clustering subtask of the MTEB benchmark (Muennighoff et al. 2023) (as of September 2025) and meets our criteria. This model is initialized from XLM-RoBERTa (Conneau et al. 2020), continually trained on a mixture of multilingual datasets, and supports 100 languages.

Experiment Setup

First, we evaluate the impact of document rewriting and task-specific embeddings. Documents are rewritten into summaries and keyphrases using Gemma 3 (27B) with straightforward prompts. See Table 5.1 for an excerpt and Appendix Table A.2 for a comprehensive prompt overview. This process is handled by the vLLM and LiteLLM libraries for batch inference, with the *max tokens to generate* parameter set to 2048; other sampling parameters, such as *temperature* and *top p*, remain at their default values.

Then, the original documents and their two modified versions are embedded using the instruction-finetuned *multilingual-e5-large-instruct* (500M) embedding model. We use the SentenceTransformers library for this step with a *batch size* of 32 and *normalize embeddings* set to true. Short instructions are provided to the model to compute aspect-oriented embeddings. Again, see Table 5.2 for an excerpt and Appendix Table A.1 for a comprehensive instruction overview.

This results in 6 zero-shot scenarios per dataset: original text (*text*), summary (*summ*), keyphrases (*keyp*), embedded with (*+inst*), and without instructions.

6. <https://www.kaggle.com/datasets/articoder/news-dataset-for-news-bias-analysis>

Dataset	Prompt
emotion-summary	Write a concise summary (maximum 5 sentences) that focuses on the emotional tone of the following song lyrics. Analyze the lyrics to determine the main emotion being conveyed, and describe how it is expressed. Conclude with an emotion categorization:
emotion-keyphrases	Generate keyphrases (max 5 phrases) that describe the emotional tone of the following song lyrics. Focus on phrases that reflect the emotional tone, mood, or feelings expressed in the lyrics.
genre-summary	Write a concise summary (maximum 5 sentences) that focuses on the genre of the following song lyrics. Analyze the lyrics to determine the musical genre, referencing stylistic elements, themes, or influences. Conclude with a genre categorization:
genre-keyphrases	Generate keyphrases (max 5 phrases) that describe the musical genre of the following song lyrics. Focus on phrases that reflect the genre’s characteristics, style, or influences.

Table 5.1: *Perspectives* evaluation - an excerpt of prompts to rewrite original documents. Find the full overview in Appendix Table A.2.

Dataset	Instruction
emotion	Identify the emotion expressed in the (summary keyphrases) of a song text
genre	Identify the genre of the (summary keyphrases) of a song text

Table 5.2: *Perspectives* evaluation – excerpt of document embedding instructions. The instructions are provided to the embedding model and were adapted to resemble those of the model authors. Find the full overview in Appendix Table A.1.

Second, we evaluate the impact of fine-tuning the embedding model with few labeled examples (2, 4, 8, 16-shot) to better align the embedding space with validated clusterings.

To mimic users validating documents, we construct few-shot training sets by randomly sampling labeled examples from datasets. While real users may select more representative examples and most likely create an imbalanced dataset, this approach comes close to the intended usage scenario. In particular, we construct 10 distinct few-shot training sets per dataset and shot count (2, 4, 8, 16) by randomly sampling *num shots* of examples per class. These few labeled training examples are used to create contrastive datasets with an equal number of positive and negative pairs. These datasets comprise document pairs, where a pair is deemed positive (1.0) if both belong to the same class and negative (0.0) otherwise. Considering C classes with N documents per class, all combinations $(N \times |C|)^2$ are evaluated, but duplicate $(A, B) = (B, A)$ and identical (A, A) pairs are removed. To balance the datasets, positive pairs are oversampled, which enables substantial training even with limited labeled samples.

The two-stage training process of the SetFit library is used to fine-tune the embedding model. The first stage fine-tunes the embedding model for one epoch

	emotion	genre	product	stars	topic	frames	blurbs	stance	bias
text	45.10	27.20	36.72	47.54	71.17	59.78	73.84	51.19	47.20
+inst	50.60	27.50	55.76	58.86	71.24	65.00	73.25	48.21	48.51
keyp	49.15	34.85	54.64	46.34	65.70	60.34	76.73	61.63	50.76
+inst	49.51	34.90	60.64	49.68	65.40	58.31	76.66	63.14	50.69
summ	47.69	32.59	48.16	52.14	60.62	64.34	76.46	62.47	48.41
+inst	48.04	32.64	61.18	54.02	61.91	65.91	74.80	68.74	52.53
2-shot	50.70	36.63	62.98	59.71	71.36	65.10	76.79	70.17	52.72
4-shot	50.92	36.60	63.54	60.24	71.49	65.59	76.74	70.56	53.03
8-shot	51.64	36.66	63.62	60.09	71.58	66.92	76.92	71.06	53.09
16-shot	52.07	37.02	64.04	61.27	72.15	67.27	77.85	71.51	54.09

Table 5.3: *Perspectives* evaluation – document rewriting, instruction-finetuned embeddings, and few-shot fine-tuning. Top: unsupervised (*zero-shot*) results. Bottom: few-shot results using the best configuration (e.g. *text+inst* for emotion & topic). Reported results are KNN accuracy scores.

on a contrastive dataset. Here, cosine similarity loss is utilized to pull documents of the same class closer and push documents of different classes apart. Instead of updating model parameters, we use rank-stabilized LoRA adapters (Hu et al. 2022) to accelerate training and significantly reduce hardware requirements. The LoRA adapter (Section 2.4.5 is parameterized with a *rank* of 24, an *alpha* of 8, and a *dropout* of 0.1.

The second stage of the SetFit training involves training a differentiable classification head for 16 epochs on the updated embedding model. We set the *end-to-end training* parameter to true to continue training the embedding model. Since our focus is solely on the document embeddings, the classification head is discarded after training.

This results in 4 few-shot scenarios (one scenario per shot) per dataset. Few-shot experiments are repeated 10 times with different training sets. The results are averaged across runs to mitigate fluctuations.

Next, the dimensionality of the resulting embeddings (from both the zero-shot and few-shot approaches) is reduced to two dimensions using a GPU-accelerated UMAP variant from the cuML⁷ library, mapping them to the 2D visualization space. The parameters are configured as follows: *N components* set to 2, *N neighbors* to 15, and *min distance* to 0.0, using cosine similarity as the distance metric.

Finally, the KNN accuracy metric is computed on these reduced document embeddings using k-nearest neighbors classifiers of the sklearn library via 5-fold cross-validation, with the reported accuracies averaged across folds. All experiments were conducted on a single A100 GPU (80GB).

Results

Table 5.3 presents our evaluation results, and Figure 5.11 shows a detailed breakdown. Consistent with previous work, we observe that embeddings generated with instructions (*+inst*) outperform those without in 8 out of 9 datasets we evaluated. More interestingly, we observe that document rewrites (keyphrases or

7. <https://github.com/rapidsai/cuml>

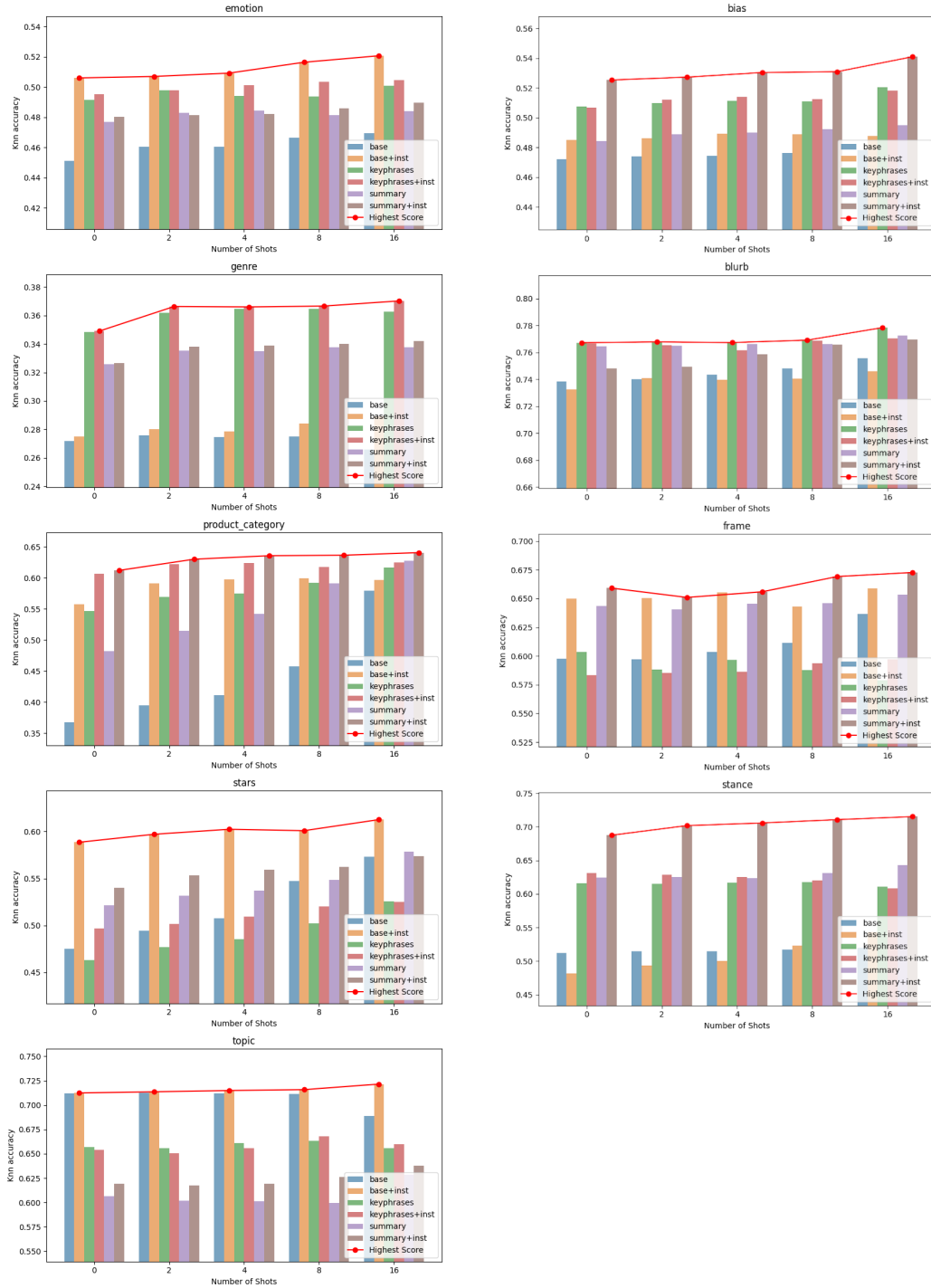


Figure 5.11: Detailed *Perspectives* evaluation with increasing number of labeled examples per class. Figure from Fischer and Biemann (2026).

summaries) prove beneficial in 6 out of 9 datasets: For example, *summary+inst* improves product categorization accuracy by 24.46 points, stance detection by 17.55 points, and bias detection by 5.33 points.

Table 5.3 also highlights our best-performing few-shot results. These scores were achieved using the best-performing *zero-shot* configurations, for example *summary+inst* for product, frames, stance, and bias; *keyphrases+inst* for genre, *keyphrases* for blurbs. We find that fine-tuning the embedding model with a

few labeled examples steadily improves KNN accuracy; however, it achieves only minor improvements (ranging from 2 to 3 points) over the zero-shot (no labeled examples) setting overall. This contrasts with the improvements in topic coherence observed by Thielmann et al. (2024) using 1-5 labeled examples per class. Still, our results suggest that incorporating more labeled examples could further enhance performance.

These findings guided the implementation of *Perspectives*: Task-specific instructions are mandatory for creating a perspective, as they consistently yield superior performance. The impact of document rewrites proved to be dataset-dependent. Therefore, providing a document-rewriting prompt is optional but recommended. To streamline the user experience, we integrated templates for tasks, such as emotion, sentiment, and topic clustering.

5.4.6 Integrating Perspectives in DATS

Perspectives is integrated into the DATS platform’s frontend (described in detail in Section 4.5) with Plotly.js⁸. This implementation leverages WebGL⁹ to enable efficient, interactive scatter-plot visualizations of large datasets.

The LLM-powered chat tool employs a simple retrieval-augmented generation approach: when users query selected documents or clusters, relevant sentences are retrieved from the vector database and fed to the LLM along with the query to generate contextualized answers.

To maintain user interface responsiveness during computationally intensive tasks of *Perspectives*—such as embedding, clustering, and model fine-tuning—the DATS backend incorporates GPU-powered RQ workers. This asynchronous architecture ensures that heavy processing is decoupled from the primary application logic. We integrate our proposed clustering pipeline, the *Perspectives* backend, into DATS using the same technology and parameters as in the experiments. Latency and response times were crucial design considerations when building this interactive system pipeline:

Document Modification: We use the Gemma 3 (27B) model. We achieve high performance for the required LLM-driven functionalities (like name generation and document rewriting) by combining vLLM and LiteLLM. This setup leverages batch processing, streaming, and prefix caching for rapid inference. While the initial, optional document rewriting process incurs latency depending on corpus size, it is an acceptable, one-time operation.

Document Embedding: We use the SentenceTransformers library with *batch size* set to 32 and *normalize embeddings* set to true to generate aspect-oriented embeddings with the instruction-tuned *multilingual-e5-large-instruct* (500M) embedding model. Computing document embeddings is significantly faster than modifying documents, as such models are typically much smaller (500M vs. 27B parameters), and while LLM generation is computationally iterative (requiring

8. <https://plotly.com/javascript/>

9. <https://www.khronos.org/webgl/>

N passes for N tokens), embedding models map the entire input into a vector space in a single forward pass.

Dimensionality Reduction: A GPU-accelerated variant of UMAP from the cuML library reduces the dimensionality of resulting embeddings. The *distance metric* is cosine similarity, *N neighbors* is 15, and *min distance* is 0.0. Here is a key distinction in the experiment setup: we apply dimensionality reduction twice, reducing the embeddings to 128 and 2 dimensions. The 2D representation is used for visualization, while the larger representation is used in the following clustering step.

Density-based Clustering: We use HDBSCAN for density-based clustering. The *min samples* is set to 40, and the *distance metric* is set to Euclidean.

Cluster Representation: A cluster in *Perspectives* is not only represented by the color/cluster assigned by HDBSCAN. In addition, based on 50 keywords extracted by c-TF-IDF, we utilize Gemma 3 to generate title & description. To ease parsing, we use vLLM’s constrained generation feature, which lets us precisely define the response format.

Interactive Refinement Operations: *Perspectives* implements various established refinement operations: adding, merging, removing, and splitting clusters, as well as adding or removing documents. The split operation is implemented using HDBSCAN with the same parameters as in the clustering step above. The other operations require no parameters. These refinement operations are fast, as their processing time scales with the number of clusters.

Model Fine-tuning: This special refinement operation is implemented with the SetFit library, as described in the experiment Section 5.4.5: First, a contrastive dataset is created based on user feedback. Next, in the two stages of the SetFit, the LoRA adapters inserted into the embedding model are first trained for one epoch on the contrastive dataset. Then, a differential classification head is trained for 16 epochs in an end-to-end fashion, further fine-tuning the embedding model. The classification head is discarded after training. Notably, this embedding model fine-tuning process is quick, requiring only one training epoch on a small labeled dataset.

5.4.7 Conclusion

This section introduced *Perspectives*, an interactive tool integrated into the DATS platform designed to help DH scholars explore and organize extensive document collections. We demonstrated an aspect-focused clustering pipeline that leverages LLM-based document rewriting and instruction-finetuned embedding models. Central to *Perspectives* is HITL refinement, providing a suite of operations to manipulate clusters and fine-tune the embedding model with minimal input. This iterative approach, visualized on an interactive 2D map and supported by a dashboard, empowers users to create meaningful categorizations. By integrating

these capabilities, *Perspectives* provides a powerful yet accessible tool that enables domain experts to leverage their knowledge effectively for categorization purposes. It transforms document clustering into a collaborative sense-making activity, prepares data for further analysis within DATS, and provides a valuable tool for user-centered DH research.

Still, the conducted experiments have limitations. Our experimental evaluation focused on a single LLM for document rewriting and a single embedding model. Different choices of LLMs or embedding models could yield varying performance outcomes. Furthermore, the few-shot training approach is sensitive to the selection of training samples. To mitigate this high dependency, we conducted 10 runs with different sampled training sets. However, an optimal combination of training examples may not have been discovered. Our experiments also did not involve extensive prompt engineering for document rewriting, as we limited the prompts to summarization and keyphrase generation. Exploring other prompts could potentially lead to further improvements. Further limitations regarding LLMs and accessibility of DATS are discussed in the overall thesis limitations in Section 9.2.

Code for replicating the experiments¹⁰ is available.

5.5 Exploring Large Language Models for Document Organization

5.5.1 Introduction

Several key steps in the qualitative data analysis workflow of DATS remain manual and resource-intensive. Specifically, researchers are engaged in a dynamic process of structuring their data and conducting in-depth qualitative analysis, which involves organizing and categorizing documents using metadata assignment and tag-based classification. Although DATS supports document tagging and metadata management, executing these tasks, particularly with large corpora, can be repetitive and time-consuming. The rise of large language models and their demonstrated capabilities for complex reasoning and zero-shot inference present new opportunities to streamline and enhance diverse QDA workflows.

This section is based on Fischer and Biemann (2024), who evaluated the effectiveness of LLMs in supporting these organizational tasks to boost researchers' efficiency and productivity within the DATS platform and introduced the *LLM Assistant* as a method for integrating these models into the ecosystem. The assistant is implemented as a flexible module, building a solid foundation for further enhancements and expansions.

The evaluation of LLMs for qualitative data analysis and the development of the *LLM Assistant* address the following research question of this thesis:

Research Question 2

How can large language models be utilized to support researchers with the organization of large datasets?

10. <https://github.com/uhh-lt/perspectives>

The practical application of LLMs in QDA for data organization primarily concerns two natural language processing tasks:

Text classification (see Section 2.5.4) is a NLP task that is traditionally defined as a supervised learning problem where the machine learning algorithm is trained on a dataset that has been pre-labeled with known categories (e.g., spam, sports, positive). Text classification, consequently, aims to automatically assign a pre-defined label or category to a new, unseen text document. However, the emergence of LLMs has enabled zero-shot classification, in which the model can accurately categorize text without task-specific labeled examples, relying instead on its vast pre-trained knowledge. This flexibility enables LLMs to assist with dynamic document tagging based on emerging qualitative categories.

Information extraction (IE) is an NLP task focused on automatically extracting structured information from unstructured and semi-structured text. This process involves transforming human-readable text into structured, factual data, typically represented as entities, attributes, and relationships (e.g., identifying "*Apple*" as a company and "*Tim Cook*" as its CEO). IE is generally regarded as a set of related tasks, including named entity recognition and relation extraction, and can be performed using both supervised and unsupervised approaches, as well as zero-shot approaches enabled by LLMs. The goal of IE is to make sense of massive text corpora by extracting relevant data for tasks such as database population, knowledge graph creation, or automated extraction of metadata fields (e.g., publication dates).

The key contributions are:

1. We articulate the envisioned AI-assisted data organization workflow within DATS, highlighting user needs and requirements.
2. We evaluate three open-weight LLMs (Llama 3.1, Gemma 2, and Mistral NeMo) on a custom-tailored benchmark for document classification and information extraction, demonstrating the promising potential of LLM integration within DATS. The benchmark consists of curated datasets aligned with real-world use cases in terms of domain and tasks, with a focus on English and German.
3. We report the integration of the *LLM Assistant* as an opt-in feature for English projects, paving the way for further enhancements and expansions.

5.5.2 Related Work

LLM Benchmarks: Several prominent LLM benchmarks have emerged, each with its unique focus and task coverage. MMLU (Hendrycks et al. 2021) evaluates models across diverse knowledge domains, while SuperGLUE (A Wang et al. 2019) emphasizes natural language understanding tasks like question answering and inference. BIG-bench (Srivastava et al. 2023) offers a vast collection of tasks spanning various modalities and difficulties, while HELM (Liang et al. 2023) provides a standardized framework for comprehensive LLM evaluation. MTEB (Muennighoff et al. 2023) has established itself as the standard benchmark for evaluating embedding models across a wide range of NLP tasks and languages. However, these benchmarks often lack a direct connection to specific real-world

applications, including qualitative data analysis. They may cover a broad range of tasks, but not necessarily those most relevant to QDA workflows. In contrast, Ziems et al. (2024) explores the potential of LLMs to transform computational social science by evaluating their zero-shot performance on a range of relevant English tasks. Their extensive evaluation, focused on taxonomic labeling and free-form coding, highlights LLMs' potential to augment social science research as zero-shot data annotators, strongly motivating our work. Still, many benchmarks prioritize English data.

Our benchmark is constructed to be relevant to the tasks and data encountered in QDA, incorporating datasets that approximate real-world scenarios within DATS. This approach aims to provide a relevant assessment of LLMs for the specific needs of QDA researchers. We aim to gather insights that can directly inform the effective integration and utilization of LLMs in qualitative research.

LLMs & QDA: Rasheed et al. (2025) explores the potential of LLMs to serve as data analysts in qualitative research within software engineering. Their approach uses a multi-agent model in which each LLM agent performs specific research tasks, such as interpreting text and interview transcripts, to automate standard qualitative analysis. Their findings suggest that LLMs can significantly accelerate data analysis, allowing researchers to handle larger datasets efficiently, which further motivates this work.

5.5.3 Workflow with Alice

This section illustrates an idealized workflow for document organization using the *LLM Assistant*. This scenario highlights potential areas where AI-powered automation could enhance productivity. The technical implementation is explained in the following sections. We will continue to follow Alice as she guides us through her research workflow.

Alice has leveraged automated clustering to organize her corpus into topics and sentiments (Section 5.4.3). Before moving to detailed qualitative coding, she uses the LLM Assistant to quickly enrich her document collection with necessary tags and metadata, overcoming the challenge of manually processing thousands of documents.

Informed by the previous exploration, Alice is interested in quantifying the comparative coverage of specific vaccine manufacturers in *Der Spiegel* and *Die Zeit*. To enable such a comparison, she must classify every relevant article.

Therefore, Alice first creates a hierarchical tag taxonomy in DATS under the parent tag "*Vaccine Manufacturer*", with three sub-tags: "*Biontech/Pfizer*", "*Moderna*", and "*AstraZeneca*". Then, she selects her entire corpus and opens the *LLM Assistant* (see Figure 5.12). She chooses *document tagging* (Step 1), then selects her new vaccine manufacturer tags in the subsequent step (Step 2). Next, Alice reviews the automatically generated zero-shot prompt (Step 3), which instructs the LLM to classify each document into one of the three vaccine categories. The job runs in the background (Step 4), and Alice grabs a coffee, waiting for the result. Once complete, the system presents a result review screen

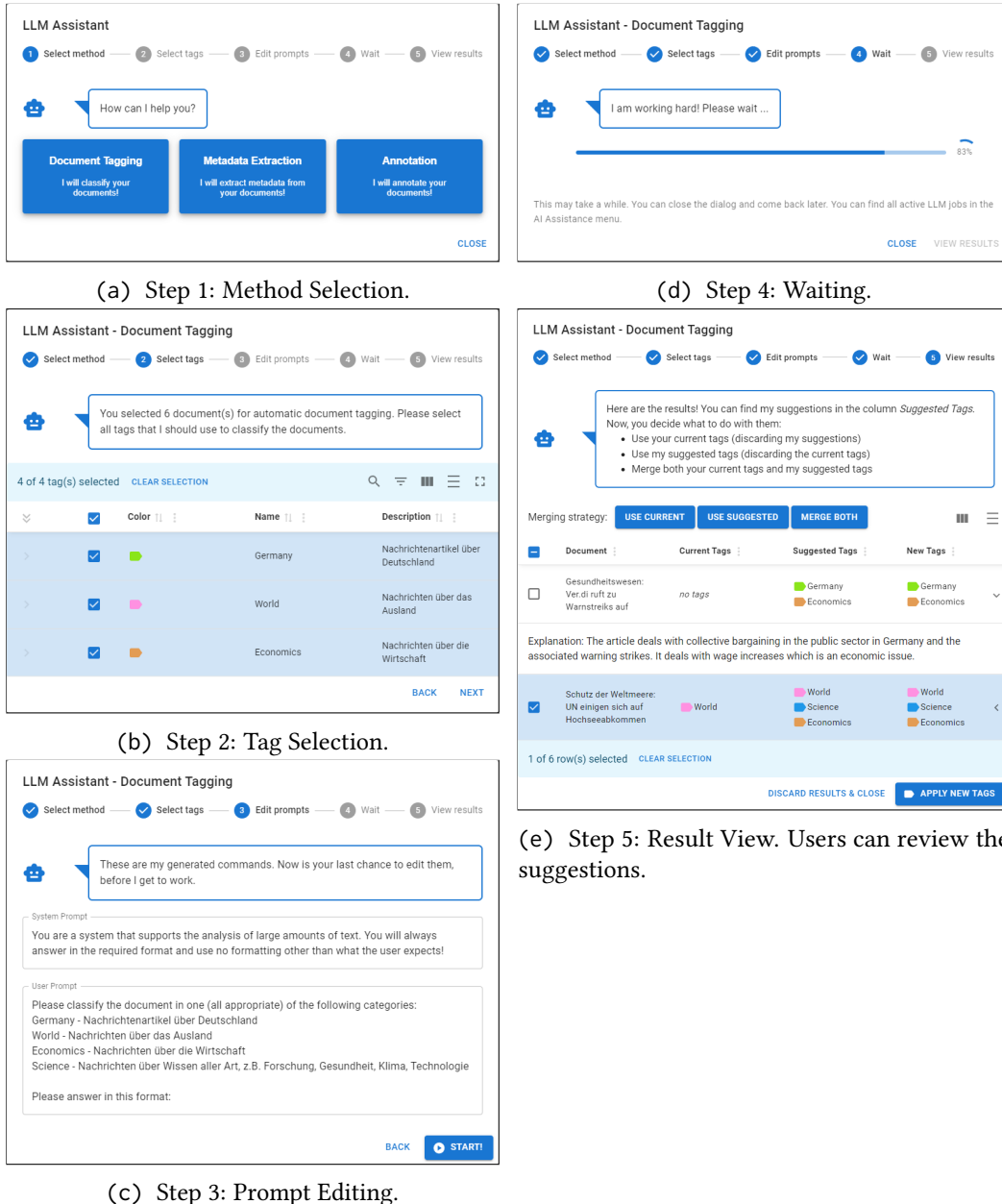


Figure 5.12: LLM Assistant 5-step user interface for document tagging.

(Step 5). This step displays the LLM’s suggested tags (e.g., “AstraZeneca”) alongside its reasoning. Alice quickly reviews and then uses merge functions to apply the LLM’s classification across the entire dataset.

Since Alice plans to conduct temporal analysis later in her project, she needs the publication dates of each news article. Similarly, identifying the article’s author could be relevant to her analysis of power and legitimacy.

Therefore, Alice defines two new custom metadata fields: “*article author*” (a text field) and “*publication date*” (a date field). Importantly, Alice must describe these metadata fields, as their descriptions will serve as instructions. She relaunches the *LLM Assistant*, selects *metadata extraction*, and chooses her newly created fields. The *LLM Assistant* performs information extraction using a template-

filling approach, scanning each article and automatically populating the selected fields. As before, she reviews the suggestions and confirms that the LLM correctly extracted the date in a machine-readable format, which is a prerequisite for the *Concept Over Time Analysis* feature in Section 7.4.

This workflow demonstrated how Alice categorized her news corpus by relevant analytical tags and contains required metadata, setting the stage for coding and further analysis.

5.5.4 Benchmark Setup

This benchmark evaluates LLMs on tasks mirroring real-world QDA use cases, as outlined in the previous section. We aim to identify the most suitable model for adequate user support to be integrated into DATS. To this end, we carefully select datasets for document classification (assigning tags to documents) and document information extraction (extraction of metadata from documents). DATS mainly caters to English and German data, so we focus our evaluation and dataset selection on these two languages.

Models

Consistent with the DATS model philosophy (Section 4.5), we use resource-efficient, locally running models to ensure data privacy and deployment in resource-constrained environments. Consequently, we evaluate three open-weight decoder-only models: Llama 3.1 (Grattafiori et al. 2024; Touvron et al. 2023), Gemma 2 (G Team 2024), and Mistral NeMo (MA Team 2024; Jiang et al. 2023) with 8B, 9B, and 12B parameters, respectively. We only test instruct fine-tuned models and use half-precision (FP16) variants. Details of these models are provided in Section 2.6.

Experiment Setup

We conduct zero-shot experiments using a single, clear prompt for each task, dataset, and language, intentionally avoiding extensive prompt engineering. Models are instructed on the expected output format, and deviations are counted as errors.

All experiments were run on a single A100 GPU (80GB) and repeated three times per configuration. Reported results are averaged across runs to mitigate fluctuations.

5.5.5 Document Classification Evaluation

Document classification plays an essential role in QDA in keeping data collections organized. DATS enables researchers to create fine-grained tag sets that classify documents into one or more groups. Many further analysis features rely on tagged documents to compare and contrast groups or sub-corpora of the material, highlighting the potential of automatic tag suggestions to improve workflows.

We assess LLMs on two relevant document classification tasks. The multi-class setting involves assigning a single class from pre-defined options, further

Task	Prompt Template
Document Classification	Classify the document in one (all appropriate) of the following categories: {} Answer in this format. You are not required to provide a reason. Category: <category> Reason: <reason> e.g. Category: News Document: {}

Table 5.4: LLM evaluation – prompts used for the document classification task in both evaluation and implementation. {} are placeholders for task-dependent input.

Task	Document Classification
Dataset	Classes (coarse - fine)
BBC	UK - england, scotland, wales, ireland, politics World - africa, asia, australia, europe, latin, us, middle-east Sport - boxing, cricket, football, formula1, rugby, tennis Misc - business, education, elections, entertainment, arts, health, science, technology
Tagesschau	Inland - Deutschlandtrend, Gesellschaft, Innenpolitik, Mittendrin Ausland - Afrika, Amerika, Asien, Europa, Ozeanien Wirtschaft - Börse, Finanzen, Konjunktur, Technologie, Unternehmen, Verbraucher, Weltwirtschaft Wissen - Forschung, Gesundheit, Klima, Technologie
IMDB	Action, Adventure, Animation, Biography, Crime, Family, Fantasy, Film-Noir, History, Horror, Mystery, Romance, SciFi, Sports, Thriller, War

Table 5.5: LLM evaluation – taxonomies of the datasets used in the document classification task.

differentiated into coarse- and fine-grained settings based on the number of classes. In the multi-label setting, a document can be associated with multiple classes.

We use the prompt template shown in Table 5.4 for this task.

Datasets

Tagesschau: Tagesschau is an established German news website known for its serious, objective reporting on news from Germany and the world. We automatically extracted a taxonomy of 4 main categories (coarse, e.g. *economy*, *knowledge*) and 20 sub-categories (fine, e.g. *finance*, *technology*, *company*) from a publicly available crawl¹¹ spanning 2018 – 2023, which contains 225,202 articles.

BBC: The BBC is the United Kingdom’s public service broadcaster, publishing English-language news from the UK and around the world, and is deemed a trusted source of information. We extracted a taxonomy of 4 main categories (coarse, e.g.

11. <https://github.com/bjoernpl/tagesschau>

Model	Dataset	Language	Task	Precision	Recall	F1	Accuracy
Gemma2	Tagesschau	de	coarse	84.88	81.94	82.01	81.94
Llama3.1	Tagesschau	de	coarse	83.13	74.79	76.01	74.79
Mistral	Tagesschau	de	coarse	83.55	77.62	78.92	77.62
Gemma2	BBC	en	coarse	76.59	76.58	74.85	76.58
Llama3.1	BBC	en	coarse	71.07	70.29	65.64	70.29
Mistral	BBC	en	coarse	69.97	71.69	68.89	71.69
Gemma2	IMDb	en	coarse	40.12	37.74	37.70	37.74
Llama3.1	IMDb	en	coarse	38.78	33.00	32.57	33.00
Mistral	IMDb	en	coarse	39.66	36.70	36.04	36.70
Gemma2	Tagesschau	de	fine	68.00	59.00	60.00	59.43
Llama3.1	Tagesschau	de	fine	58.00	26.00	29.00	25.80
Mistral	Tagesschau	de	fine	65.00	43.00	47.00	42.86
Gemma2	BBC	en	fine	73.00	65.00	66.00	65.47
Llama3.1	BBC	en	fine	57.00	46.00	39.00	46.07
Mistral	BBC	en	fine	68.00	64.00	63.00	64.27
Gemma2	IMDb	en	multi-label	55.48	52.23	52.35	8.58
Llama3.1	IMDb	en	multi-label	52.12	42.38	43.95	6.88
Mistral	IMDb	en	multi-label	57.51	49.62	51.05	8.58

Table 5.6: LLM evaluation – results of the document classification task.

world, sport) and 26 subcategories (fine, e.g. *boxing, cricket, rugby*) from the BBC dataset published by Yucheng Li et al. (2024), covering 2017–2024, which contains about 85,000 articles. The dataset¹² is continually updated.

IMDb Genres: IMDb, the Internet Movie Database, contains information about movies, TV shows, and other content. The IMDb genres dataset¹³ includes about 400,000 movie descriptions and their classification into one of 16 major genres (coarse, e.g. *action, adventure, fantasy*) and 2-3 of 25 subgenres (multi-label, e.g. *crime, mystery, thriller*).

For all datasets, we sampled 10,000 documents. The taxonomies of the datasets are presented in Table 5.5. Since analyzing news articles is a typical use case within the Discourse Analysis Tool Suite, the Tagesschau and BBC datasets, which are news datasets, are a good fit for this benchmark. While not directly related to our users’ domains, the IMDb genres dataset was explicitly included for its multi-label classification task. It is unlikely that DATS will be used to analyze movies, but it is an easy-to-use, publicly available dataset for multi-label document classification.

Results

We evaluate document classification across three subtasks: coarse-grained, fine-grained, and multi-label classification. The results are presented in Table 5.6. We report weighted precision, recall, F1, and accuracy.

12. https://huggingface.co/datasets/RealTimeData/bbc_news_alltime

13. <https://kaggle.com/datasets/rajugc/imdb-movies-dataset-based-on-genre>

Task	Prompt Template
Extractive QA	Extract the answer to the following question from the context below: Context: {} Question: {} Answer in this format. If the question cannot be answered from the context, respond with 'Not answerable'. You are not required to provide any reasoning. Answer: <answer> or <not answerable> Reasoning: <reasoning> e.g. Answer: 42 Remember, the answer MUST be extracted verbatim from the text. Do not generate it!
Template Filling	I prepared a list of slots. The slots are: {} Extract the corresponding information (if any) from the context below: {} Answer in this format. If the text does not include information about a specific slot, leave it empty. <Slot>: <extracted information> e.g. Incident: Arson, Perpetrator: John Doe, Weapon: Matches Remember, you MUST extract the information verbatim from the text Do not generate it!

Table 5.7: LLM evaluation – prompts for the information extraction task. They are used in both evaluation and implementation. {} are placeholders for task-dependent input.

Gemma 2 consistently outperforms the other models. It maintains high performance (achieving F1 scores of up to 82 in the coarse setting and up to 66 in the fine-grained setting) despite a remarkable increase in the number of classes. In contrast, the other models struggle in the fine-grained scenario, which requires effectively distinguishing among numerous categories. Llama 3.1 performs the worst, achieving F1 scores of 29 and 39.

Next, we compare English and German performance. For coarse-grained classification, all models exhibit superior performance on the German Tagesschau dataset compared to the English BBC dataset. However, a noticeable performance drop of over 22 points F1 occurs across all models on the fine-grained German task. This suggests challenges in German fine-grained classification.

5.5.6 Metadata Extraction Evaluation

Similar to classifying documents, assigning additional metadata to research material helps to organize extensive data collections. Users can define project-wide metadata for their documents, e.g., "*interview date*", "*company name*", or "*position*". Or, for a project with news articles about protests, users may define metadata such as "*protest location*" or "*number of participants*". Document metadata is essential to many analysis features, including search and filtering, visualization, and quantitative analysis. Automatically extracting such meta information from provided documents could notably boost researchers' productivity, making this a relevant task for our benchmark. A user may need to provide a description in

Task	Document Information Extraction
Dataset	Slots
MUC-4	Incident: One of 'Arson', 'Attack', 'Bombing', 'Kidnapping', 'Robbery' Perpetrator: An individual perpetrator Group Perpetrator: A group or organizational perpetrator Victim: Sentient victims of the incident Target: Physical objects targeted by the incident Weapon: Weapons employed by the perpetrators

Table 5.8: LLM evaluation – taxonomy of the MUC-4 dataset used in the template filling information extraction task.

addition to the metadata name, such as *"The position of the interviewed person within the company"*, or a question like *"What is the position of the interviewed person?"*

We consequently frame metadata extraction as a document-level information extraction task, given a pre-defined set of slots. The user would define the slots by creating corresponding metadata fields. Hence, the slots are also pre-defined for this benchmark.

We explore extractive question-answering and template-filling approaches for this task. In extractive question answering, the system is presented with a context and a question and is expected to locate and extract the answer verbatim from the context or determine if it is unanswerable. Template-filling (similar to slot-filling, relation extraction, and event extraction) focuses on extracting multiple related pieces of information. For example, in relation extraction, arguments of a relation are identified; in slot-filling, information about a specific intent is extracted; in event extraction, relevant information about a particular event is extracted.

We use the prompt templates shown in Table 5.7 for these tasks.

Datasets

SQuAD: The Stanford Question Answering Dataset (Rajpurkar et al. 2016) is a benchmark dataset for extractive question answering. It consists of 100,000 questions posed by crowd workers about a set of Wikipedia articles, with each question answered by a text segment from the context.

SQuAD 2.0: The SQuAD 2.0 dataset (Rajpurkar et al. 2018) builds on the original and introduces 50,000 additional unanswerable questions, making it more challenging. Crowd workers carefully crafted the unanswerable questions to be similar to answerable ones.

GermanQuAD: The GermanQuAD dataset (Möller et al. 2021) is a German counterpart to SQuAD and contains 13,722 questions. This extractive question-answering dataset was carefully constructed by students and experts familiar with machine learning and question answering (QA), using German variants of the English Wikipedia articles used in SQuAD.

Model	Dataset	Language	Task	Exact Match	F1
Gemma2	SQUAD1	en	extractive-qa	75.73	87.19
Llama3.1	SQUAD1	en	extractive-qa	70.39	82.64
Mistral	SQUAD1	en	extractive-qa	67.48	83.35
Gemma2	SQUAD2	en	extractive-qa	63.85	69.80
Llama3.1	SQUAD2	en	extractive-qa	48.82	56.25
Mistral	SQUAD2	en	extractive-qa	50.81	60.64
Gemma2	GermanQuAD	de	extractive-qa	60.01	81.25
Llama3.1	GermanQuAD	de	extractive-qa	48.91	73.04
Mistral	GermanQuAD	de	extractive-qa	47.34	73.18
Gemma2	MUC-4	en	template-filling	36.69	36.70
Llama3.1	MUC-4	en	template-filling	40.62	40.63
Mistral	MUC-4	en	template-filling	23.86	23.87

Table 5.9: LLM evaluation – results of the information extraction task.

MUC-4: The MUC-4 dataset (Sundheim 1992), designed for template-filling, contains 1700 news articles about terrorist incidents. It requires systems to classify each incident and fill five slots of relevant information. The taxonomy of this dataset is presented in Table 5.8.

The SQuAD datasets, with multiple questions per Wikipedia passage, align with our scenario of extracting various metadata from documents. While some questions align with typical metadata fields (e.g., "*when*", "*how many*", "*where*"), others capture more complex information, making them less suitable for metadata extraction. Nonetheless, strong performance on these datasets indicates potential for successful application in DATS. The MUC-4 dataset aligns well with our use case of extracting multiple metadata from documents, particularly within the news domain, making it an ideal fit for our benchmark.

Results

We evaluate document-level information extraction across two subtasks: extractive QA and template filling. The results are shown in Table 5.9. We report exact match and F1 scores, with template-filling scores averaged across all slots.

Gemma 2 exhibits a clear advantage in extractive QA, achieving an F1 score of up to 87. Llama 3.1 performs best at template filling, with an F1 score of 40. Interestingly, template filling is a greater challenge overall, likely due to the increased complexity of extracting multiple correct answers simultaneously, as reflected in lower scores across all models (at least 56 F1 for extractive QA vs. at most 40 F1 for template filling). In extractive QA, the models exhibit comparable performance on the GermanQuAD and SQuAD datasets, indicating no significant difference in language capabilities between English and German.

5.5.7 Integrating the LLM Assistant in DATS

Across the evaluated tasks, Gemma 2 consistently performs best. While Llama 3.1 performs best at template filling, it struggles with instruction following and consistently performs worst on German tasks, highlighting potential limitations. Mistral NeMo, while generally capable, could never demonstrate superior performance, even though it has the most parameters (12B) among the benchmarked models.

Regarding language performance, our findings reveal discrepancies. While all models perform well on German data for coarse-grained document classification, a consistent decline is observed in fine-grained classification. This suggests that current open-weight LLMs still struggle to handle the complexities of German.

Still, our findings highlight the potential of LLMs, especially Gemma 2, to enhance QDA workflows regarding document classification and document information extraction. The benchmark results provide valuable insights for informing the integration of LLMs into DATS.

We designed the *LLM Assistant* as a standalone feature that can be accessed at any time. The feature is initiated from the search view by selecting the documents for analysis and clicking the *LLM Assistant* button. This launches a five-step dialog. The steps are described in the previous Alice workflow in Section 5.5.3, and are depicted in Figure 5.12:

1. **Task Selection:** Users select *document tagging* or *metadata extraction*.
2. **Focus:** Users specify which tags or metadata fields the LLM should consider.
3. **Prompt Review:** Users revise the system and user prompts that are automatically generated based on the chosen task and selections
4. **Job Execution:** A progress bar indicates that the *LLM Assistant* job runs in the background. The dialog can be closed and reopened later.
5. **Result Review:** LLM-generated suggestions and their reasoning alongside the documents' existing tags/metadata are presented. Here, the user can choose to keep the existing work, overwrite it with the LLM's suggestions, or merge both per document or in batches.

The *LLM Assistant* is integrated into the DATS platform (described in detail in Section 4.5) using existing LLM-serving (vLLM, Gemma 3) and background-processing (RQ workers) capabilities. We repurpose the prompts from our benchmark evaluation because they were intentionally designed for future implementation. We opt for the template-filling prompt style rather than extractive question-answering for metadata extraction, even though the results suggest that extractive question-answering is the better approach for document-level information extraction. Providing metadata descriptions is more intuitive for users than formulating fitting questions for each field. Consequently, DATS allows users to provide metadata descriptions (for metadata extraction) and tag descriptions (for document tagging).

Since the initial implementation of the *LLM Assistant*, LLM-serving libraries (such as vLLM) have improved drastically. While we were required to write

heuristic-based LLM parsing functions for conducting experiments, DATS uses vLLM’s constrained generation feature, which allows us to precisely define the model’s response format. When conducting these experiments, a major issue was parsing the model’s responses. Throughout all experiments, we tracked parsing errors. Gemma 2 consistently adhered to the provided instructions (1% issues), followed by Mistral NeMo (2% issues). Llama 3.1, on the other hand, struggled notably (5% issues), most notably in span classification tasks (which are evaluated in Section 6.6). With the introduction of constrained generation, all model responses can nowadays be parsed easily and correctly.

5.5.8 Conclusion

In this section, we investigated the potential of LLMs to enhance qualitative data analysis workflows, focusing on automatic metadata extraction and document tagging in the Discourse Analysis Tool Suite. We designed a benchmark reflecting real-world use cases and evaluated the performance of three prominent open-weight LLMs. Our findings demonstrate the promise of LLM integration, particularly for English-language projects. Consequently, we implemented the *LLM Assistant* within DATS, a significant step towards empowering researchers with transparent, user-controlled AI assistance that augments rather than replaces human expertise. By adopting a modular and flexible system design, this integration paves the way for future LLM-based enhancements within DATS.

While the *LLM Assistant* shows promise in enhancing QDA workflows, it has some limitations. The implementation utilizes zero-shot learning, which may not fully capture the nuances of specific QDA tasks. Fine-tuning LLMs on user-specific data can lead to more accurate, contextually relevant suggestions. Also, we restricted the *LLM Assistant* to English-language projects due to performance discrepancies observed between English and German language tasks. Expanding language support will require further research and development by model developers until similar performance is achieved across different languages. Further limitations regarding LLMs, benchmark datasets, and accessibility of DATS are discussed in the overall thesis limitations in Section 9.2.

Code for replicating the benchmark¹⁴ is available.

5.6 Conclusion & Discussion

This chapter established the necessity of digital support for exploration, search, and organization in discourse analysis (Section 5.1). Assistance with exploration is necessary during the initial corpus construction phase, as manual review of the entire dataset is infeasible. Search and filtering functionality are essential for reducing potentially large datasets and finding theoretically relevant documents, allowing sub-corpora to be created, extended, or restricted iteratively. Finally, robust organizational tools are critical for structuring the data, which eases later analytical steps, such as the historical and comparative methods of discourse analysis.

These three methodological processes are challenged by five factors (Section 5.2): scalability, data heterogeneity, linguistic ambiguity, theoretical sampling

14. <https://github.com/uhh-lt/llm4qda>

(identifying cases for minimal/maximal contrasting), and the need for flexible category systems.

DATS core functionalities include keyword and semantic search, as well as manual organization (Section 5.3). These features successfully address data scalability through filtering, manage data heterogeneity via a unified processing pipeline that maps all modalities to the text domain, and mitigate linguistic ambiguity and the need for minimal contrasting by using contextualized embeddings in its semantic search function.

We subsequently presented two key contributions to the DATS platform. First, *Perspectives* (Section 5.4) addresses the challenge of exploring unknown corpora by integrating interactive text clustering. This extension provides an interactive document map based on contextualized embeddings, allowing for a holistic overview. Further, document clustering can be tailored to the researcher's needs, accounting for the varied requirements across disciplines, projects, and research questions. Second, the *LLM Assistant* (Section 5.5) reduces the manual effort required to organize extensive document collections. By leveraging zero-shot large language models for text classification and information extraction, this feature improves researcher efficiency in structuring the corpus and extracting relevant metadata.

In conclusion, this chapter demonstrated the practical application of advanced NLP techniques, including contextualized embeddings, interactive text clustering, and LLM-based, zero-shot text classification and information extraction, to address methodological challenges in discourse analysis.

Discussion

The methodological and technical advancements presented in this chapter — the *Perspectives* interactive clustering extension (Section 5.4), the *LLM Assistant* for organization (Section 5.5), and the conceptual parallel work on configurable similarity search (Section 5.3.3) — address several requirements established for a discourse analysis tool suite supporting hermeneutic, Grounded Theory-based research in Section 1.5. This section discusses these contributions with respect to the technical requirements.

The most immediate impact of these contributions is on *scalability*. The *LLM Assistant* addresses the efficiency bottleneck of manual corpus organization by automating repetitive metadata assignment and tagging across extensive document collections. Similarly, *Perspectives* enables the exploratory structuring of vast, initially unknown corpora by grouping documents into meaningful clusters, providing the systematic overview required for the methodologically guided reduction of large data volumes. The underlying architectures for both features were designed for *responsiveness* by leveraging GPU-accelerated task processing and optimized inference techniques, thereby ensuring a smooth research process.

All main contributions prioritize the human-in-the-loop philosophy: *user control* and *transparency*. The *LLM Assistant* maintains user control by requiring explicit user approval for all AI-generated suggestions, affirming the researcher's final authority. *Transparency* is achieved through the prompt review step, which allows users to inspect and modify the underlying LLM instructions, and the result review, which displays the model's reasoning alongside its suggestions. *Perspec-*

tives operationalizes *user control* through both initial steering (via instruction-tuned embeddings and document rewriting) and post-clustering refinement operations (e.g., merge, split, add cluster). This system's *transparency* is delivered through the interactive 2D map visualization and the LLM-generated cluster names and statistics, which translate computational outputs into interpretable analytical concepts.

Furthermore, *Perspectives* addresses *adaptability* and *interactivity*. It allows the clustering process to be continuously refined via model fine-tuning with only a few user-provided labeled examples. This supports the dynamic evolution of analytical concepts. The refinement operations are interactive, providing the quick feedback loop essential for Grounded Theory's iterative approach.

The contributions also meet the *flexibility* requirement of utilizing general-purpose approaches that are easily adaptable to diverse use cases. The *LLM Assistant*'s zero-shot classification and information extraction capabilities allow it to adapt to any user-defined tag set or metadata scheme. *Perspectives* similarly promotes *flexibility* through aspect-focused analysis. By leveraging instruction-finetuned embedding models and document rewriting, the system enables users to apply arbitrary analytical lenses (e.g., topic, style, sentiment), ensuring it remains relevant across the diverse domains and materials studied by discourse analysis researchers.

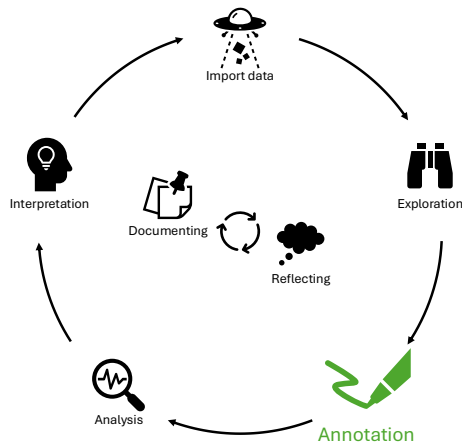
The advanced search features presented in the parallel work on configurable similarity search address another methodological challenge: *theoretical sampling*. By explicitly parameterizing the search function for similarity along one analytical dimension while simultaneously requiring dissimilarity along another (e.g., matching entities but contrasting tone), this conceptual system could locate the deviant cases as are necessary for maximal contrasting. This function improves *transparency* over traditional semantic search by making the reason for similarity explicit.

Finally, *accessibility* is considered, as the target user group often lacks deep expertise in NLP or machine learning. To address this, the user interfaces for all contributions were designed to abstract from computational complexity. For instance, *Perspectives* transforms the intricate multi-stage clustering pipeline (embeddings, dimensionality reduction, clustering) into an intuitive, interactive 2D document map. Similarly, the *LLM Assistant* simplifies LLM prompting into a streamlined dialog. This focus on intuitive design tries to guarantee that non-experts can effectively leverage state-of-the-art AI for their research.

Collectively, the implemented systems and concepts presented in this chapter demonstrate that advanced NLP techniques can be effectively integrated to augment qualitative discourse analysis.

With the corpus successfully explored and organized into theoretically relevant sub-corpora using tools like *Perspectives* and the *LLM Assistant*, the research process moves from the "macro" level of distant reading to the "micro" level of close reading. In the discourse analysis workflow, this transition to close reading and coding of the materials marks the beginning of the interpretive-analytical phase. While this chapter focused on finding and categorizing documents, the next chapter addresses the challenge of analyzing their content. We explore how

digital tools can support the labor-intensive task of qualitative coding, moving from manual span-based annotation to advanced AI-augmented systems that provide suggestions to the researcher.



6

Annotation

Contents

6.1	Motivation	138
6.2	Challenges	139
6.3	Coding vs. Annotation	140
6.4	Computational Grounded Theory	141
6.5	Annotation in DATS	144
6.5.1	Workflow with Alice	144
6.5.2	Addressed Challenges	145
6.5.3	Excursus: Annotation Scaling	145
6.6	Exploring Large Language Models for Annotation	147
6.6.1	Introduction	147
6.6.2	Workflow with Alice	148
6.6.3	Sequence Labeling Benchmark	149
6.6.4	Integrating Annotation in DATS' LLM Assistant	153
6.6.5	Conclusion	153
6.7	Semi-automatic Sequential Sentence Classification	154
6.7.1	Introduction	154
6.7.2	Workflow with Alice	156
6.7.3	Three-phase Annotation Assistance	157
6.7.4	Benchmarking Sequential Sentence Classification	159
6.7.5	Transitioning Phase 2 $\rightarrow$ 3	163
6.7.6	Integrating Sequential Sentence Annotation in DATS' LLM Assistant	164
6.7.7	Conclusion	164
6.8	Excursus: AnnoPlot - Visual Annotation Organization	165
6.9	Conclusion & Discussion	167

This chapter addresses challenges regarding qualitative coding, the process of identifying and categorizing specific text segments to reconstruct discursive formations. As the volume of relevant text grows, manual coding becomes a

significant bottleneck. This chapter explores how NLP methods can augment this interpretive process.

We begin by establishing the motivation and challenges of coding (Sections 6.1 and 6.2). Next, we describe the difference between coding in qualitative analysis and annotation in NLP (Section 6.3), and highlight the opportunities when combining both approaches (Section 6.4). Section 6.5 describes the baseline annotation features in DATS, including manual span annotation and hierarchical code management, illustrated through Alice’s continuing study of vaccine discourse.

The chapter presents two primary AI-driven advancements:

1. Building on the LLM Assistant (Fischer and Biemann 2024) introduced in Section 5.5, we evaluate the zero-shot capabilities of LLMs in Section 6.6 to support researchers with coding, identifying entities and concepts within text. This section presents a sequence labeling benchmark and details the integration of these suggestions into the DATS workflow.
2. Section 6.7 introduces a three-phase approach to annotation assistance. This *Annotation Assistant* (Fischer and Biemann 2025) dynamically adapts its strategy—shifting from zero-shot to few-shot prompting and eventually to fine-tuned models—as the researcher provides more manual annotations.

By developing these AI-augmented tools, this chapter addresses Research Questions 3 and 6, demonstrating how AI can aid coding in the discourse analysis workflow and incrementally learn from the researcher’s expertise to scale qualitative insights to larger volumes of text.

6.1 Motivation

Coding in discourse analysis is the systematic process of identifying, marking, and naming the discursive elements and patterns contained in the material. It serves to isolate the elements of discourse, making them comparable and manageable for subsequent analysis and interpretation. Thus, coding represents the first analytical transformation of data from pure text into analytical categories. As an indispensable bridge between the corpus and the reconstruction of the discursive formation, systematic coding is necessary to prove the discursive formation and to distinguish it from merely coincidental statements (Keller 2011b, pp. 62–63). This analytical work thus serves the goal of reconstruction in discourse analysis (Section 3.4.2).

Elements of Coding: Coded are the content-related and formal elements of the typical statements that point to the existence of a discursive formation (Traue et al. 2022, p. 638). Thereby, the four elements of discourse (the rules of formation) are isolated across statements: objects, concepts, enunciative modalities, and strategies (Section 3.4.1). For example:

- **Objects:** It is coded what is thematized, named, described, or paraphrased at all. It is clarified which problems or facts come to expression (and which are systematically omitted).

- **Concepts:** The keywords and domain-specific terms that are used within the discourse are coded. The focus is on which other concepts a given concept corresponds to (e.g., "*efficiency*" in connection with "*automation*") and how the meanings of these concepts change over time.
- **Enunciative Modalities:** It is coded how the objects are constructed. These include descriptions, classifications, narratives, depictions, metaphors, argumentative structures, and interpretive figures that appear in the text.
- **Strategies:** Strategies capture the actions or intentions that the text, the speaker, or the institution pursues. It is coded which solutions or norms are proposed and which functions the text serves in the social field (e.g., legitimization, critique, normalization).

Coding Techniques of Grounded Theory: The Sociology of Knowledge Approach to Discourse (SKAD) draws on the coding strategies of Grounded Theory (GT) and adapts them to a discourse-analytical epistemic interest (Section 3.6). The procedure of discourse analysis is iterative, and the elements of discourse are not determined a priori but are deduced from empirical data, following Strauss et al. (1996), in three phases: open coding, axial coding, and selective coding (Section 3.2.2). The exemplary questions are taken from Keller (2011a, p. 106):

- **Open Coding:** This creates thematic accesses to the material and begins with a case-related analysis of the text passages to capture the elementary building blocks of the discourse. The goal is the initial, bottom-up identification of all relevant discursive elements without fixed specifications. Example questions: *What actually happens in the text? Which category does the text passage point to?*
- **Axial Coding:** Subsequently, the open codes are compared, summarized, and bundled into central categories (e.g., patterns of interpretation or classifications). The central question is oriented towards the causes, circumstances, and consequences of a concept, to create networks of meaning and to work out the relations between the coded elements. Example questions: *What conditions exist for the events combined in the concept? How can the interaction between the actors be described? What strategies and tactics can be identified? What are the consequences of the events?*
- **Selective Coding:** Finally, the central categories are placed in relation to a key category. This serves theory building. Example questions: *What is most striking about the area of investigation? What do I consider to be the main problem? What is the common thread of the story? Which phenomena are reflected again and again in the data?*

6.2 Challenges

The transfer of interpretive coding procedures from discourse analysis into digital tools poses several challenges, primarily due to the qualitative methodology and the need to scale to large data volumes.

Large Data Volumes: A central difficulty arises from handling large data volumes. Discourses manifest in a multitude of documents (including newspaper articles, interviews, official papers, and social media entries), whose manual coding becomes increasingly problematic given the enormous, thematically relevant data available online. Manual coding is a labor-intensive process, especially with large datasets. It limits the scope of investigation, especially for small teams.

Multimodal Data: In addition to the sheer volume of data, handling multimodal data is challenging and requires specialized approaches. Discourse analysis often involves analyzing multimodal data, including video, audio, and images. The development of annotation tools that can adequately process these diverse data types represents a complex requirement for qualitative data analysis software.

Code Organization: Further, specific problems arise from the open coding techniques in Grounded Theory. In open coding, codes should be developed closely from the material, often in the form of in-vivo codes. In-vivo codes utilize direct quotes from the material as labels. The challenge in dealing with in-vivo codes is that they quickly lead to an unmanageable number of codes that are difficult to categorize. This significantly complicates the subsequent consolidation. Furthermore, maintaining consistency over time remains a central difficulty. When coding large datasets over extended time periods, there is a risk that researchers unintentionally shift their interpretation of codes, leading to inconsistent coding.

Transparency: Finally, there are the challenges of transparency and subjectivity. Coding and subsequent interpretation are highly interpretive processes. The evolution of the codebook and the coding of text passages must be understandable to third parties. Hence, it is necessary to make the rationale for the coding decision (e.g., "*Why was this segment coded with code X?*") transparent to ensure the traceability of the research process.

Natural language processing offers an interesting opportunity to automate repetitive coding tasks, such as automatically recognizing actors, named entities, or sentiment. However, all automatically identified annotations must be understood by the researcher, and interpretation remains complex. A tool could make the uncertainty of automated annotation transparent and streamline the researcher's correction and interpretation of automated results.

6.3 Coding vs. Annotation

Notable discrepancies exist between discourse-analytical practices and typical natural language processing workflows. This section elaborates on the differences between the interpretive coding process in the digital humanities and the functional annotation practice in computer science.

In the digital humanities, coding is the process of identifying, naming, and categorizing segments of data to allow for theory to emerge directly from the text. Within the Grounded Theory framework, this is an inductive process of meaning-making that moves from specific data points to general conceptualizations. Here,

the researcher approaches the material bottom-up, intending to discover rules of formation and structures that emerge from the data itself. The codebook remains dynamic and flexible, as codes are constantly refined, discarded, or newly formulated in response to reflection and deepening understanding. In this context, the researcher's subjectivity is not a source of error to be eliminated but an elementary part of the analytical process. Consequently, the primary metric of success is theoretical saturation. The point at which no new insights or categories are being generated from the analysis of additional data. High inter-annotator reliability is often a secondary objective, as the primary goal is the researcher's deep interpretive engagement with the material.

Conversely, annotation in the context of NLP generally follows a deductive, top-down approach. The primary objective is functional: to produce a high-quality "ground truth" dataset for training or evaluating machine learning models. This requires a stable, clear, a priori-defined annotation schema (annotation guidelines) in which labels must remain static to ensure consistency across large volumes of data. Because the goal is to train a model to replicate human patterns, high inter-annotator agreement is mandatory as evidence of the dataset's objectivity. In this paradigm, annotator subjectivity is considered a source of noise that must be minimized through strict protocols and unambiguous definitions.

In summary, the distinction lies in the objective and the role of the human actor. Coding is an open-ended, interpretive act of discovery (bottom-up), whereas annotation is a rule-governed act of categorization (top-down). While coding focuses on the evolution of a conceptual framework until saturation is reached, annotation focuses on the reliable application of an existing one.

Beyond these conceptual differences, this thesis also employs "annotation" as a neutral technical term within the context of the Discourse Analysis Tool Suite. As established in the DATS Terminology (Section 4.4), annotation refers to the technical practice of storing the relevant material, user, and code. In this sense, the term encompasses the technical implementation of the labeling process, regardless of whether it resulted from inductive coding or deductive annotation.

6.4 Computational Grounded Theory

As outlined in the previous section, there are significant discrepancies between NLP annotation techniques and Grounded Theory (discourse analysis) coding practices. While they seem incompatible at first glance, there are actually great opportunities when combining both approaches. The Computational Grounded Theory (CGT) by Nelson (2020) methodically combines both worlds, bridging the gap between human interpretation and machine scale.

CGT allows researchers to leverage the strengths of both paradigms through a three-step process that combines the "close reading" of the humanities with the "distant reading" of computer science (see Figure 6.1).

1. **Pattern Detection:** The process begins with the inductive use of computational techniques to reduce complicated, messy text into simpler, more interpretable lists or networks of words. By utilizing unsupervised machine learning (e.g., topic modeling), the researcher can see patterns in the data that might be obscured by preconceived notions or the sheer volume of the

corpus. This step allows the computer to help the researcher "see" latent structures.

2. **Pattern Refinement:** Following the computational exploration, the researcher returns to an interpretive engagement with the data. Through guided "deep reading" of text segments that are representative of the computationally identified patterns, the researcher adds hermeneutic depth. This step confirms the plausibility of the patterns and adds the necessary qualitative interpretation that machines cannot provide.
3. **Pattern Confirmation:** In the final step, the inductively identified and refined patterns are assessed using further computational and natural language processing techniques. This serves to confirm that the patterns hold throughout the entire corpus and are not merely artifacts of the algorithm or researcher bias. For example, Nelson demonstrates this by using WordNet (Miller 1995) specificity scores to confirm that certain discourses are more abstract than others, and by using named entity recognition to quantify whether specific actors (individuals vs. organizations) appear consistently across a corpus, as predicted by the qualitative phase.

This synthesis offers opportunities for digital discourse analysis. A primary benefit is the ability to overcome the constraints of labor-intensive manual analysis, which often restricts researchers to limited datasets. By incorporating computation, researchers can integrate massive amounts of data into theory-generating research, achieving both "breadth and depth". Furthermore, it mitigates individual researcher bias by using computational pattern detection to surface insights that might otherwise be overlooked, while the subsequent interpretive steps ensure that the results remain sociologically meaningful and grounded in the data.

Extending this synthesis, this thesis develops the concept of *Annotation Scaling* as a bridge between the interpretive and computational steps outlined by Nelson. This process represents the practical transition from human interpretation to computational classification. It begins with in-depth, qualitative coding, in which the researcher interprets patterns and develops a theory through close reading. Once the categories have stabilized, these codes are operationalized into a formal annotation schema. By applying this schema to an extended document set, the researcher provides the necessary "gold standard" examples required for machine learning. Finally, a supervised model (such as a BERT-based (Devlin et al. 2019) classifier) can be trained on these human-validated annotations to scale the analysis, annotating the remainder of the corpus with the same logic established during the qualitative phase.

The development of the Discourse Analysis Tool Suite specifically aims to operationalize this bridge. By providing a working environment where the transition from exploratory coding to large-scale annotation is fluid, DATS enables the kind of scaling described in this framework. Later in this chapter, we present a two-stage method combining semantic search and classification for *Annotation Scaling* (Section 6.5.3) and introduce a three-stage *Annotation Assistant* (Section 6.7). This assistant dynamically adapts to the researcher's progress—moving from zero- or few-shot prompting to training full machine-learning models—to provide increasingly precise suggestions as qualitative insights increase.

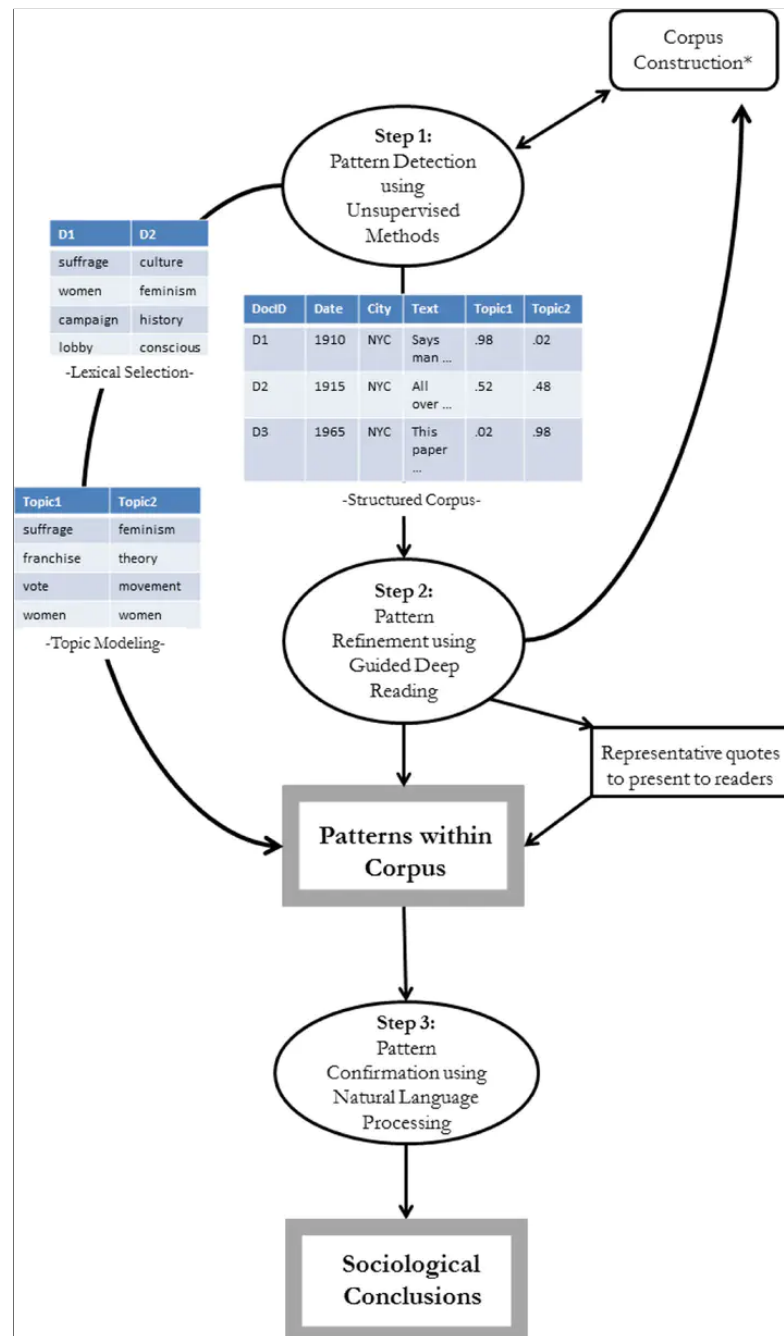


Figure 6.1: Three-step Computational Grounded Theory framework: from dataframe to conclusion: "Step 1 serves two purposes: It outputs interpretable lists of words, as shown on the left side of the figure, and produces a structured corpus which can guide the deep reading step (step 2) as shown in the center of the figure. The first two steps, including potentially collecting more data, are iterative and can be done as many times as needed to identify useful patterns in the data. Step 3 is the final confirmation step that formalizes the patterns identified in steps 1 and 2 and allows the researcher to draw reliable conclusions from the data". Quotation and figure from Nelson (2020).

6.5 Annotation in DATS

This section demonstrates how to use the base annotation features in DATS. The technical implementation of these features is explained in Section 4.3. Alice will continue to guide us through her research workflow.

6.5.1 Workflow with Alice

After organizing her corpus into relevant topics and extracting key metadata using *Perspectives* (Section 5.4.3) and the *LLM Assistant* (Section 5.5.3), Alice shifts focus to close reading. This coding phase is the first interpretive step in the SKAD methodology, in which she begins the manual process of identifying the discursive elements needed to reconstruct the discursive formation.

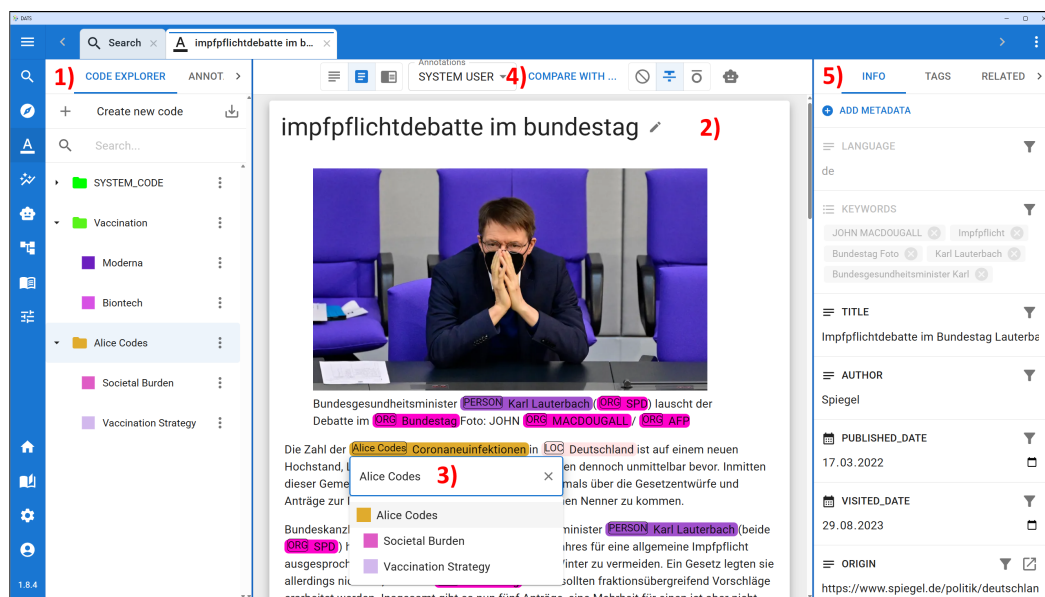


Figure 6.2: DATS annotation interface. It is designed for open coding, featuring the hierarchical code explorer (1), the document display for span annotation (2), coding on-the-fly with a pop-up for code selection (3), the annotator selector to view collaborators' annotations, and an informative sidebar (5) that displays all document metadata, memos, or related documents.

Alice navigates to the annotation view (see Figure 6.2) to begin the iterative coding process. This interface is designed to support the flexibility required by open coding in Grounded Theory. The code explorer in the left sidebar (1) allows Alice to view and manage her hierarchical code system (taxonomy). She can create, rename, delete, and merge codes, supporting the iterative refinement of categories inherent in axial and selective coding. She can also switch this sidebar view to see an overview of all coded text passages.

In the middle of the screen, the document is rendered, complete with any associated image (2). Here, Alice performs the typical qualitative work: selecting text passages (spans) to identify discursive elements like "*Societal Burden*" or "*Vaccination Strategy*" and applying the relevant codes (3). For image documents, she can select image regions to code visual rhetoric. The annotator selector at the top (4) lets her choose which annotations to show, enabling her to compare her

own annotations with those of her teammates, which is helpful for collaborative projects. When selecting the *SYSTEM USER*, Alice can see all automatically annotated entities (e.g., persons, organizations, locations) that were extracted during preprocessing (Section 4.6.3).

The right sidebar (5) provides access to all document information. By default, this shows metadata assigned to the document (e.g., the publication date and author extracted by the *LLM Assistant*). Further, this sidebar allows Alice to switch between views to see either related documents (such as pages from the same book or images embedded in the article) or a memo overview. Here, Alice can write a memo or view all contextual notes to document her interpretive rationale for this document.

By integrating annotation, taxonomy management, and contextual information into a single, cohesive view, DATS supports the coding phase and ensures that Alice's coded material is grounded and traceable to the source material and to her emergent theory. At any time, Alice can export the automatically and manually created annotations, the created taxonomy, and the raw documents in standard formats for further analysis with other tools.

6.5.2 Addressed Challenges

DATS provides basic annotation functionality. Users can create a hierarchical taxonomy, add annotations, and compare their annotations with those of other users.

First, it supports annotating two modalities: text with span annotations and images with bounding-box annotations. Time-span annotations for audio and video annotation are not implemented.

Furthermore, DATS includes automatic named entity recognition, primarily to enable more detailed search and filtering. However, it provides no effective means to correct or interpret automated annotations, nor does it make the model's certainty transparent. In practice, automatic entity annotations often overwhelm users, as each annotation must be analyzed and understood. Most of the time, users disabled all automatic annotations, as they obstructed their own annotations.

Consequently, DATS' core features address only a few of the methodological challenges identified in Section 6.2, leaving room for further enhancements.

6.5.3 Excursus: Annotation Scaling

DATS' basic annotation functionalities do not address the challenge of handling large data volumes during the labor-intensive coding process. In our first step towards this, we developed the concept of *Annotation Scaling* in parallel work (Petersen-Frey et al. 2023).

Qualitative research approaches, such as discourse analysis, rely on intensive, iterative, and manual coding. However, these methods cannot be easily extended to the large-scale corpora available today, which often contain thousands of documents. While NLP methods offer opportunities for scaling, they typically require vast amounts of labeled training data (though current approaches such as few-shot training and in-context learning for LLMs reduce this requirement) and expertise in machine learning, making them unsuitable for most humanities

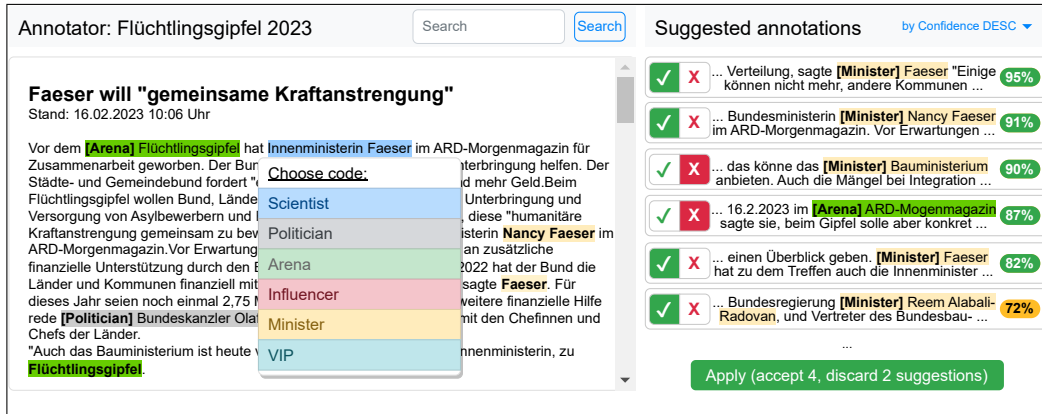


Figure 6.3: Annotation Scaling mockup. Left: Document with automatic suggestions enabled. Highlighted texts prefixed with [code] are manually created annotations; other highlights are system suggestions. Right: Batch approving/discarding suggestions to semi-automatically scale annotations. Figure from Petersen-Frey et al. (2023).

projects. This disparity creates a notable gap between in-depth qualitative analysis and the utilization of large datasets.

In this work, we approached this by conceptualizing a human-in-the-loop system in which the machine learning component serves as an assistant to the human researcher, requiring neither programming nor NLP expertise. This framework proposes a two-stage method combining semantic search and classification:

Stage 1: Interactive Scaling. This stage focuses on accelerating the coding workflow. After a researcher manually codes a few text spans, the system uses contextualized embeddings (e.g., models from the SentenceTransformers library by Reimers and Gurevych (2019)) to represent the meaning of those spans. These embeddings are then used to compute dynamic representatives for each category. Leveraging an approximate nearest neighbor index for high-speed retrieval (e.g., FAISS¹), the system suggests new, semantically similar text passages from the entire corpus. Suggestions are filtered using a negative support set (previously rejected samples) to ensure high precision. The suggestions are shown to the researcher. By accepting or rejecting these suggestions, the system adapts instantly to the researcher’s interpretation, accelerating the coding process considerably.

Stage 2: Fully-automatic Scaling. Once a sufficient number of manually verified examples have been collected, the conceptual system will automatically train a classifier on all examples. To this end, we use transformer adapters (Section 2.4.5), which are parameter-efficient layers inserted into pre-trained language models. They enable the creation of high-quality span- or sentence-classification models with few training instances and reduced computational effort, which is beneficial in resource-constrained settings.

To validate this two-stage approach, we simulated the interactive coding workflow across multiple datasets and compared different candidate retrieval strategies. The advanced retrieval strategy, which utilized the negative support

1. <https://faiss.ai/>

set, demonstrated a strong performance. Furthermore, we demonstrated that adapters are effective for training the classifier, achieving strong performance on named entity recognition tasks with only 30-100 training instances per class, thereby confirming the technical feasibility and efficiency of the two-stage scaling framework.

In conclusion, this work envisioned a system that would allow qualitative researchers to scale their annotation-based research semi-automatically to extensive document collections (see Figure 6.3). Our approach is designed to bridge the gap between qualitative and quantitative results by enabling systematic annotation of thousands of documents, thereby supporting the quantitative grounding of hypotheses derived from prior qualitative findings. Core ideas of this work are reused in the *Concept Over Time Analysis* feature presented in Section 7.4.

6.6 Exploring Large Language Models for Annotation

6.6.1 Introduction

Coding, one of the core steps in DATS' discourse analysis workflow, remains resource-intensive and time-consuming. Users of DATS engage in a dynamic process of structuring data and conducting in-depth qualitative analysis, which involves coding documents with a custom, hierarchical code taxonomy. Although DATS provides basic support for coding, executing this task, particularly with large corpora, can be very demanding. The rise of large language models (LLMs) and their demonstrated capabilities for complex reasoning and zero-shot inference present new opportunities to streamline and enhance the coding workflow.

This section is based on the second part of the work presented in Fischer and Biemann (2024), which focuses on using LLMs to support the coding task and boost researcher productivity within the DATS platform. While the first part of this work (Section 5.5) addressed document organization through document tagging and metadata extraction, this section details the span-annotation capabilities. Together, these features constitute the *LLM Assistant*, providing support across multiple levels of the qualitative research process.

The evaluation of LLMs for annotation assistance and the implementation of the *LLM Assistant* address the following research question of this thesis:

Research Question 3

How can language models be used to assist researchers during the qualitative coding process and enable the scaling of these insights to large datasets?

The practical application of LLMs for span annotation primarily concerns the NLP task of sequence labeling (Section 2.5.4). Traditionally, this is defined as a supervised learning problem where a model is trained on a pre-labeled dataset to assign categories—such as person, organization, or location—to each token in a sequence. However, LLMs enable zero-shot classification, allowing the model to label tokens without task-specific labeled examples by relying on its pre-trained

knowledge. This flexibility enables LLMs to assist with annotating categories that emerge dynamically from the data.

The key contributions are:

1. We articulate the envisioned AI-assisted coding workflow in DATS, highlighting user requirements.
2. We evaluate three open-weight LLMs (Llama 3.1, Gemma 2, and Mistral NeMo) on a custom-tailored benchmark for sequence labeling, demonstrating the promising potential of LLM integration within the Discourse Analysis Tool Suite.

Since the *LLM Assistant* was introduced in Section 5.5, we only briefly discuss how annotation with LLMs is integrated in DATS.

6.6.2 Workflow with Alice

This section illustrates an idealized workflow for span annotation using the *LLM Assistant*. This scenario highlights potential areas where AI-powered automation could enhance productivity. The technical implementation is explained in the previous Section 5.5.7. We will continue to follow Alice as she guides us through her research workflow.

Having established her initial code taxonomy through close reading and identified the manual burden of coding every single actor (Section 6.5), Alice now uses the *LLM Assistant* to continue her analysis of power and legitimacy by performing large-scale span annotation on her German news corpus.

First, Alice ensures her code taxonomy includes the necessary codes for her power analysis, such as "*Politician*" or "*Robert Koch Institute*"². Importantly, she provides detailed descriptions for each code, which serve as instructions. Then, she selects a relevant sub-corpus, e.g., by filtering by a previously applied document tag, and initiates the *LLM Assistant* (see Figure 6.4).

She chooses *annotation* (Step 1) and selects her target codes "*Politician*" and "*Robert Koch Institute*" in the subsequent step (Step 2). Next, Alice verifies the automatically generated zero-shot prompt (Step 3), which instructs the LLM to perform sequence labeling on the news text. By providing the LLM with code definitions, she leverages its zero-shot inference capabilities to recognize these concepts in her news articles. The job runs in the background (Step 4) and applies the selected codes to fitting text passages of the selected documents. Finally, Alice is presented with the result review screen (Step 5). It displays the LLM's suggested text spans (e.g., "*Karl Lauterbach*" tagged as "*Politician*" or "*RKI*" tagged as "*Robert Koch Institute*"). Alice uses this screen to accept, reject, or modify these suggestions.

This process creates a comprehensive, large-scale annotation of key actors in the discourse, providing Alice with a foundation for visualizing and analyzing power dynamics.

2. The Robert Koch Institute is Germany's central federal government agency and research institute for disease control and prevention. During the COVID-19 pandemic, it served as the primary scientific authority providing data, risk assessments, and policy recommendations to the German government.

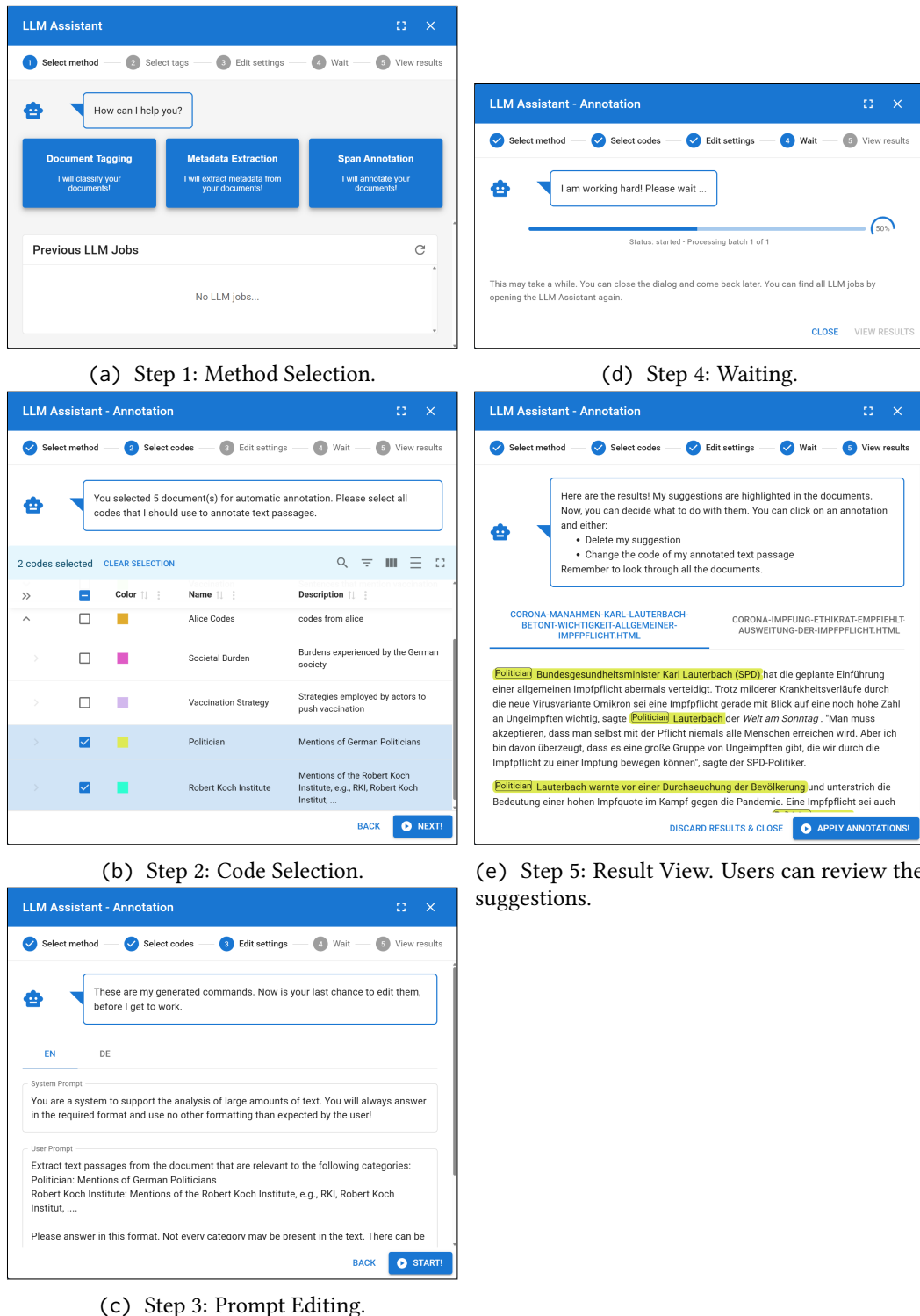


Figure 6.4: Steps 1-5 of the *LLM Assistant* dialog for span annotation.

6.6.3 Sequence Labeling Benchmark

As elaborated in this chapter, coding of relevant text passages (spans) is an important analytical step in discourse analysis projects. The workflow described in the previous section highlighted how automating parts of the coding process could streamline researchers' workflows. With this benchmark, we aim to identify

Task	Prompt Template
Sequence Labeling	I prepared a list of categories/information. The categories are: {} Extract fitting text spans from the following text: {} Respond in the following format: <category>: <extracted text> e.g. Art: Mona Lisa, Building: Eiffel Tower Remember, you MUST extract the information verbatim from the text. Do not generate it!

Table 6.1: LLM evaluation – prompts used for the sequence labeling task in both evaluation and implementation. {} are placeholders for task-dependent input.

the most suitable model for effectively supporting users like Alice in annotating large datasets. It evaluates LLMs on the annotation task, which translates to the NLP task of sequence labeling.

Models

Since this section covers the second part of our work, it utilizes the same models as in Section 5.5. In alignment with the DATS model philosophy (Section 4.5), we evaluate three open-weight decoder-only models, all chosen for their balance of performance and size: Llama 3.1 (Grattafiori et al. 2024; Touvron et al. 2023), Gemma 2 (G Team 2024), and Mistral NeMo (MA Team 2024; Jiang et al. 2023) with 8B, 9B, and 12B parameters, respectively. We only test instruct fine-tuned models and use half-precision (FP16) variants. Details of these models are provided in Section 2.6.

Experiment Setup

The experiment setup is consistent with Section 5.5: We ran zero-shot experiments utilizing a single, clear prompt for each dataset and language, intentionally avoiding resource-intensive prompt engineering. To ensure reliable results and mitigate fluctuations, all experiments were run on a single A100 GPU (80GB) and repeated three times per configuration, with the reported results averaged across these runs. Furthermore, all models were instructed on the expected output format, and any deviations were counted as an error. Because users can create fine-grained code systems to code text passages in DATS, we formulate the automatic annotation of text passages as coarse- and fine-grained sequence labeling tasks. We use the prompt template shown in Table 6.1 for this task.

Datasets

FewNERD: FewNERD (Ding et al. 2021) is a fine-grained, large-scale named entity recognition dataset consisting of 8 coarse-grained (e.g., *person*, *location*, *event*) and 66 fine-grained categories (e.g., *person-actor*, *location-gpe*, *event-election*). Over 180,000 sentences of Wikipedia articles were carefully annotated by experienced annotators.

Task	Sequence Labeling
Dataset	Classes (coarse - fine)
FewNERD	Art - broadcastprogram, film, music, other, painting, writtenart
	Building - airport, hospital, hotel, library, other, restaurant, sportsfacility, theater
	Event - attack, disaster, election, other, protest, sportsevent
	Location - GPE, bodiesofwater, island, mountain, other, park, road
	Organization - company, education, government, media, other, party, religion, show organization, sportsleague, sportsteam
	Other - astronomy, award, biology, chemical, currency, disease, educational degree, god, language, law, living thing, medical
	Person - actor, artist, athlete, director, other, politician, scholar, soldier
	Product - airplane, car, food, game, other, ship, software, train, weapon
GermanLER	Person - Anwalt, Richter
	Ort - Land, Stadt, Straße, Landschaft
	Organisation - Unternehmen, Institution, Gericht, Marke
	Norm - Gesetz, Verordnung, EU Norm
	Regulierung - Vorschrift, Vertrag
	Rechtsprechung Literatur
Quotations	Sprecher
	Direkte Rede

Table 6.2: LLM evaluation – taxonomies of the datasets used in the sequence labeling task.

GermanLER: GermanLER (Leitner et al. 2020), the German legal entity recognition dataset, consists of German legal documents and a typology relevant to court decisions with seven coarse-grained and 19 fine-grained types (e.g., *court*, *institution*, *law*, *court decision*, *legal literature*). About 66,000 sentences were annotated by two domain experts.

Quotation Attribution: The dataset for quotation attribution in German news articles (Petersen-Frey and Biemann 2024) comprises 1,000 annotated German news articles from Wikinews³. It includes information about *who* said *what* to *whom* as well as *how* and *in which context*. Quotations are categorized into direct, indirect, free, and reported speech.

While categories such as persons, events, and organizations are arguably interesting for discourse analysis projects (e.g., for analyzing networks and relationships among such entities), other categories are probably less relevant. We often see more lengthy annotations spanning multiple sentences in such projects (which is addressed in Section 6.7). Still, we decided to include FewNERD (as a representative for English) and GermanLER (as a representative for German) in this benchmark,

3. <https://www.wikinews.org/>

Model	Dataset	Language	Task	Precision	Recall	F1	Accuracy
Gemma2	FewNERD	en	coarse	48.53	55.72	51.15	83.73
Llama3.1	FewNERD	en	coarse	40.60	7.72	12.87	79.46
Mistral	FewNERD	en	coarse	49.95	18.97	26.32	81.12
Gemma2	GermanLER	de	coarse	21.85	32.42	24.82	86.32
Llama3.1	GermanLER	de	coarse	11.84	1.67	2.78	81.21
Mistral	GermanLER	de	coarse	28.60	11.79	16.11	82.41
Gemma2	FewNERD	en	fine	47.92	40.78	42.31	83.23
Llama3.1	FewNERD	en	fine	38.97	6.53	10.54	79.20
Mistral	FewNERD	en	fine	39.53	12.68	17.64	79.86
Gemma2	GermanLER	de	fine	32.58	22.49	21.37	84.48
Llama3.1	GermanLER	de	fine	14.36	1.78	3.02	81.54
Mistral	GermanLER	de	fine	39.91	8.64	13.13	82.22
Gemma2	Quotations	de	quotations	36.40	25.61	29.81	90.29
Llama3.1	Quotations	de	quotations	33.16	10.88	15.18	88.17
Mistral	Quotations	de	quotations	23.03	10.69	13.71	86.24

Table 6.3: LLM evaluation – results of the sequence labeling task.

as NER is a prominent sequence labeling task, and we argue that understanding the concepts of these datasets is likely required to perform well on more complex annotation tasks. GermanLER is included in our benchmark because it is the only German NER dataset with a relatively large and complex taxonomy. Furthermore, as DATS is expected to be used by researchers in political science and law, it is interesting to analyze the capabilities of LLMs on German legal texts. Quotation attribution is part of the benchmark, as the identification and annotation of utterances were common to multiple projects we conducted with colleagues from the digital humanities. However, we simplify this task for the benchmark by considering only *speaker* and *direct speech* annotations. The taxonomies of the datasets are presented in Table 6.2.

Results

We evaluate sequence labeling across three subtasks: coarse- and fine-grained NER (coarse, fine), and quotations (quot). The evaluation results are listed in Table 6.3. We report weighted precision, recall, F1 score, and accuracy. Accuracy includes the classification of outside tokens.

Gemma 2 consistently outperforms the other models across all sub-tasks (21–51 F1). Llama 3.1’s performance is notably poor (3–15 F1), primarily due to frequent failures to adhere to output formatting instructions. The model often reverses the positions of NER labels and their corresponding text spans, leading to significant parsing errors. To maintain a fair comparison across all models, we retain our original parsing algorithm and prompt rather than fix such errors. Consequently, our experiments also evaluate the models’ instruction-following capabilities. Increasing the number of classes from coarse-grained to fine-grained NER results in a notable performance drop across all models. Comparing English and German results reveals significantly lower scores for German datasets. Even the best performing model, Gemma 2, achieves at most half the F1 score in German

compared to English. This could suggest challenges in handling German text or reflect the increased difficulty of GermanLLER.

6.6.4 Integrating Annotation in DATS' LLM Assistant

Our findings suggest that open-weight LLMs struggle to handle the complexities of German. Still, at least for English projects, we see the potential of LLMs to enhance coding workflows. Hence, we also integrated an annotation functionality into the *LLM Assistant*.

The LLM-based annotation feature is accessed the same way as document tagging or metadata extraction: users select the documents for analysis and click the *LLM Assistant* button. This launches the five-step dialog shown in the previous Alice workflow in Section 6.6.2 and is depicted in Figure 6.4:

1. **Task Selection:** Users select *annotation*.
2. **Focus:** Users specify which codes the LLM should consider.
3. **Prompt Review:** Users revise the default system and user prompts.
4. **Job Execution:** A progress bar indicates that the *LLM Assistant* job runs in the background. The dialog can be closed and reopened later.
5. **Result Review:** LLM-generated annotation suggestions and their reasoning are presented. Here, the user can choose which ones to keep or discard.

The *LLM Assistant* is integrated into the DATS platform (described in detail in Section 4.5) using existing LLM-serving (vLLM, Gemma 3) and background-processing (RQ workers) capabilities. We repurpose the benchmark prompts.

6.6.5 Conclusion

In this section, we investigated the potential of LLMs to enhance coding, a fundamental step in qualitative data analysis projects. We designed a benchmark consisting of German and English datasets and evaluated the performance of three prominent open-weight LLMs. Our findings demonstrate the promise of LLM integration, at least for English-language projects. Consequently, we also integrated span-annotation functionality into the *LLM Assistant*. This assistant represents an important first step towards empowering researchers with AI assistance, and by adopting a modular, flexible system design, this integration also paved the way for the LLM-based feature presented in the next section.

Limitations regarding performance discrepancies between English and German language tasks, as well as zero-shot prompting and fine-tuning of LLMs, were mentioned in Section 5.5.8. Limitations regarding LLMs are discussed in the overall thesis limitations in Section 9.2.

Code for replicating the benchmark⁴ is available.

4. <https://github.com/uhh-lt/llm4qda>

6.7 Semi-automatic Sequential Sentence Classification

6.7.1 Introduction

One of the core QDA tasks on the DATS platform is in-depth qualitative analysis through the manual coding of text documents. This process involves searching for documents relevant to a research question, creating and expanding a category system (taxonomy), and reviewing the material to code relevant passages. While DATS provides span-level annotation functionality (e.g., annotations of arbitrary length, from a single word to multiple sentences), our observations during collaborations with researchers across various disciplines indicate that coding occurs primarily at the sentence or paragraph level (e.g., coding a sequence of sentences). Moreover, a common requirement is the analysis of taxonomy distribution across the entire dataset—for example, to validate a theory, to see how a particular category typically occurs, or to see how categories relate to one another. However, obtaining such quantitative insights requires coding a large amount of data, which is often impractical to do manually.

This section is based on Fischer and Biemann (2025), who addressed these requirements by introducing a framework for sequential sentence annotation into DATS and by providing researchers with tools for efficient, practical analysis. We developed a user interface specifically designed for sequential sentence annotation. Here, a single sentence is the smallest possible unit to annotate (as opposed to a single word in the span annotation interface). This interface includes functionalities for comparing annotations between users and, most importantly, an *Annotation Assistant* that learns from existing annotations to provide suggestions for unseen data.

Central to our approach is a three-phase *Annotation Assistant* that provides progressively refined suggestions as users code data. Key requirements drive this design, including user control over the annotation process, optimal suggestions at each stage, and fast inference for a smooth user experience. The three phases are:

1. **Zero-shot Prompting:** When no annotations are available, the assistant leverages the taxonomy and definitions to generate initial suggestions via zero-shot prompting of large language models.
2. **Few-shot Prompting:** As users annotate data, the assistant transitions to few-shot prompting, refining its suggestions by incorporating user-provided examples.
3. **Fine-tuned Model:** With sufficient annotations, a sequence tagger can be fine-tuned to this data, further improving the accuracy of suggestions.

To evaluate the effectiveness of our three-phase approach and select the best model to integrate into the DATS, we construct a benchmark comprising five datasets for sequential sentence classification. We assess the zero- and few-shot performance of three prominent open-weight LLMs: LLaMA 3.1, Gemma 2, and Mistral NeMo. Further, we fully fine-tune and evaluate sequence tagging models

on this benchmark. Our approach works best with Mistral NeMo and steadily improves as the number of annotated samples increases.

The development and evaluation of the three-phase *Annotation Assistant* addresses two research questions of this thesis:

Research Question 3

How can language models be used to assist researchers during the qualitative coding process and enable the scaling of these insights to large datasets?

Research Question 6

How can AI-augmented methods—including visualization, annotation, and semantic search—be effectively designed within a human-in-the-loop framework to incrementally and rapidly adapt to evolving concepts and domain-specific inquiries in qualitative discourse analysis projects despite the scarcity of labeled data?

The underlying task addressed by this work is sequential sentence classification, a special case of sequence labeling in NLP (Section 2.5.4). Sequential sentence classification is a task in which the basic unit of analysis is the sentence rather than the token. The algorithm must assign a category (or code) to every sentence in a document. It is similar to, for example, named entity recognition, but instead of tagging tokens, it tags sentences. Hence, every sample of a typical sequential sentence classification task consists of a list of sentences and corresponding tags assigned to each sentence.

The contributions are:

1. We formulate an AI-assisted sequential sentence-annotation workflow, as envisioned by our project partners and us, highlighting user needs and requirements.
2. We propose a three-phase *Annotation Assistant* that, depending on the number of annotated examples, utilizes zero-shot, few-shot prompting, or a fine-tuned model.
3. We construct a benchmark on sequential sentence classification to evaluate the approach.
4. We report on the integration of AI-assisted sequential sentence annotation into DATS.

With these contributions, this section extends the *LLM Assistant* with sentence-level annotation capabilities. While document organization (tagging and metadata extraction) and span-level annotation assistance are detailed in Sections 5.5 and 6.6, respectively, this section describes the sentence-level annotation assistance, the sentence-level annotator component, and the integrated comparison view, which facilitates collaboration.

6.7.2 Workflow with Alice

This section illustrates an idealized workflow for sentence annotation using the three-phase *Annotation Assistant*, highlighting potential avenues to improve productivity. The technical implementation is explained in the following sections. We will continue to follow Alice as she guides us through her research workflow.

Having used the *LLM Assistant* to automate span annotation of actors (Section 6.6.2), Alice turns her attention back to the discourse's rhetorical framing. To efficiently code the subtle differences in risk perception, she employs the three-phase *Annotation Assistant*, designed explicitly for sequential sentence classification.

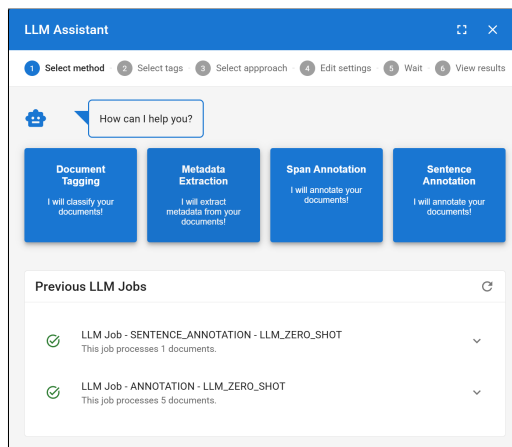
Before coding, Alice and her project team brainstorm a taxonomy of frames relevant to risk perception. The taxonomy includes frames such as "*Individual Risk*", "*Societal Burden*", and "*Economic Cost*", among others. They carefully define each frame and formulate detailed descriptions.

With the taxonomy in place, Alice opens the three-phase *Annotation Assistance* (see Figure 6.5), which is an enhanced version of the *LLM Assistant*. She selects the *sentence annotation* task and her framing codes (Step 1 + 2). The system presents her with the choice of annotation strategy (Step 3), offering three options based on the amount of labeled data currently available (zero-shot, few-shot, or fine-tuning). Since she has no prior labels for framing, the assistant recommends the zero-shot approach. Next, a prompt is displayed to Alice for review (Step 4). It was generated by the system and includes her taxonomy and definitions. Alice confirms the prompts. Finally, the job runs (Step 5), and Alice needs to wait.

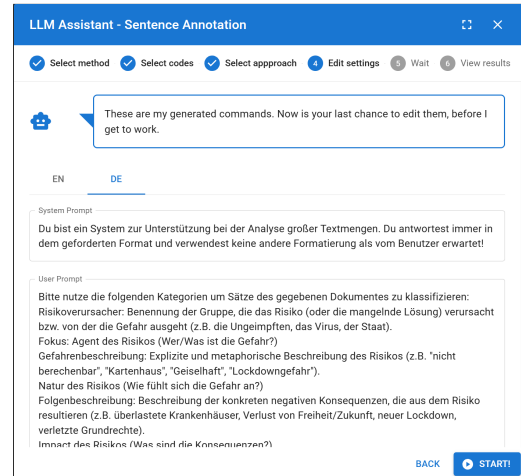
Once complete, Alice is directed to the sentence annotator (Figure 6.6). This side-by-side view displays the document aligned sentence by sentence, with Alice's annotations on the left and the assistant's suggestions on the right. She reviews them carefully, using the control buttons in the middle (2) to instantly accept or copy suggested annotations from the LLM to her own workspace. Hovering over an annotation highlights the corresponding text (3). Hovering over the category name in the taxonomy highlights all annotated texts (4). In case Alice notices further interesting sentences, she can easily create annotations by clicking and dragging sentences where needed.

After manually labeling a minimum required number of sentences per frame, Alice repeats the process. This time, the three-phase *Annotation Assistant* leverages her examples to either perform few-shot prompting (by passing examples into the prompt) or, once sufficient data is available, automatically fine-tune a sequence tagger. This iterative process tries to provide the best suggestions at any stage. The improved suggestions accelerate Alice's progress, enabling her and her team to annotate a significant portion of the dataset efficiently. After a couple of iterations and with a larger pool of annotated data, the assistant fine-tunes a specialized model, improving the quality of its suggestions.

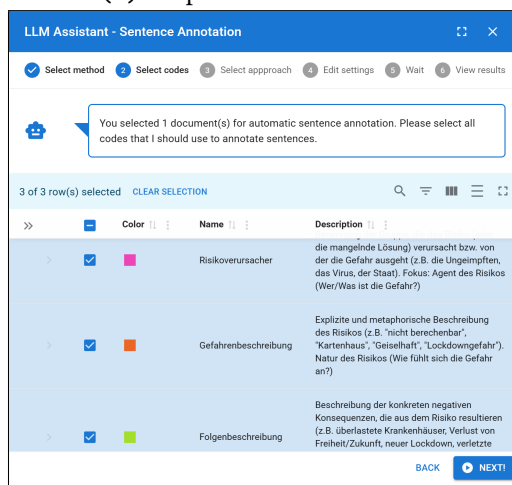
Confident in the assistant's capabilities, Alice can run the fine-tuned model to automatically annotate the remaining documents. This process takes time. Later, Alice reviews some of the assistant's annotations. Satisfied with the quality, she proceeds to quantitative analysis. For example, she can utilize DATS' built-in features to visualize how the framing evolves (Section 7.4) or export the annotated data for further analysis with her other preferred tools.



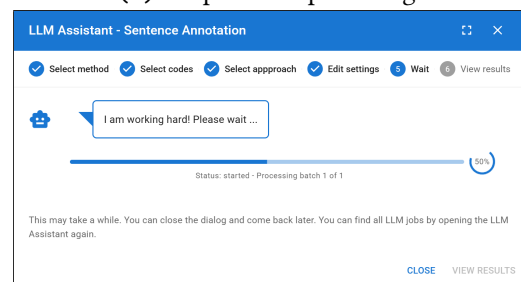
(a) Step 1: Method Selection.



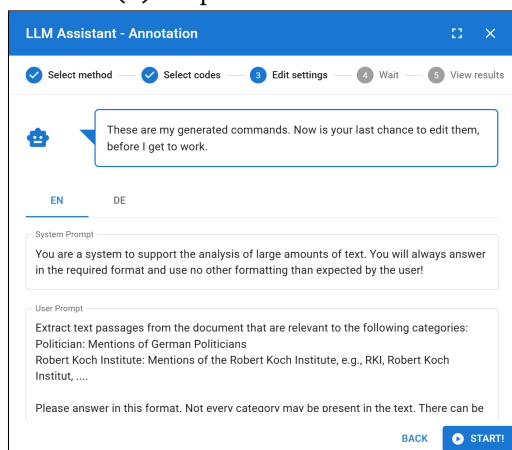
(d) Step 4: Prompt Editing.



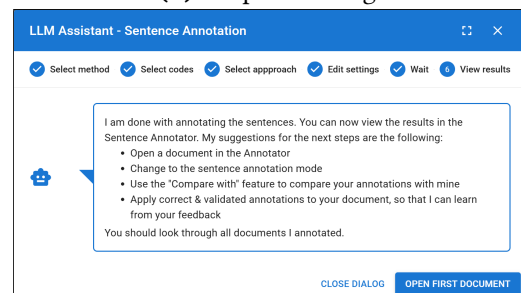
(b) Step 2: Code Selection.



(e) Step 5: Waiting.



(c) Step 3: Approach Selection.



(f) Step 6: Result View. Users can review the suggestions.

Figure 6.5: Steps 1-6 of the enhanced three-stage *LLM Assistant* dialog for sentence annotation.

6.7.3 Three-phase Annotation Assistance

The workflow described in the previous section highlighted how the *Annotation Assistant* could empower researchers like Alice to analyze large textual datasets

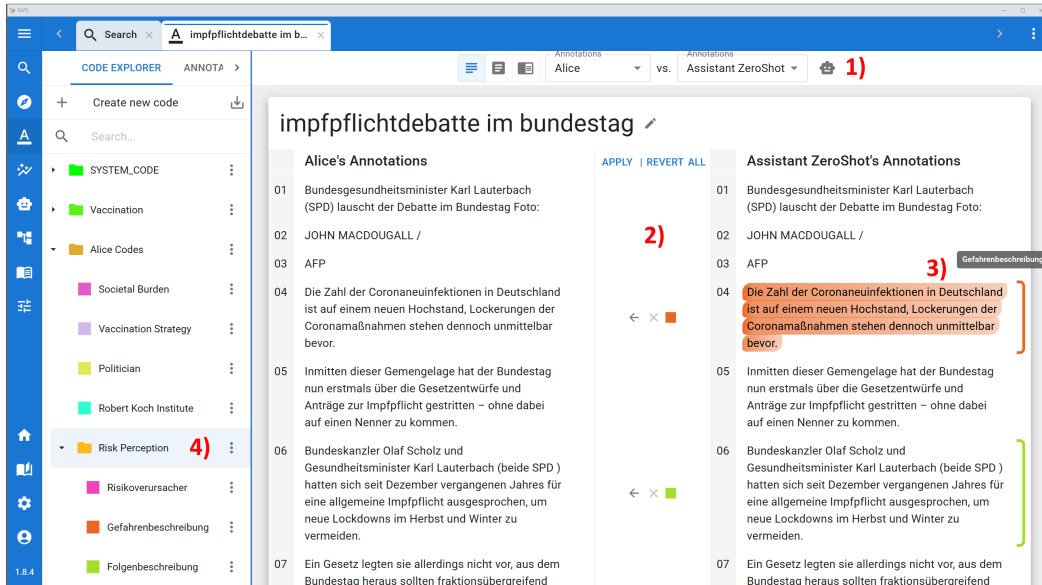


Figure 6.6: DATS comparative sentence annotator. It provides a comparison view between two users' annotations. Here, it is used to compare user and assistant annotations. The button (1) opens the *LLM Assistant* dialog. The central control panel (2) facilitates rapid review and acceptance of suggested annotations. Hovering over an annotation (3) highlights the corresponding text. Hovering over a category highlights all corresponding passages (4).

effectively. To achieve this, we designed the assistant with a focus on user centricity. Most importantly, the assistant should provide the best possible suggestions at each stage of the coding process. This requirement motivated our three-phase approach to AI-assisted coding, which leverages increasing amounts of user-provided data to refine its suggestions:

Phase 1: Zero-shot Prompting. When no annotations are available, the assistant leverages the provided taxonomy and definitions to generate initial suggestions through zero-shot prompting of LLMs. Please note that the user provides the taxonomy and category definitions, which are fully customizable.

Phase 2: Few-shot Prompting. After the user annotates a few documents and meets a certain threshold of labeled examples per category, the assistant transitions to few-shot prompting, refining its suggestions by incorporating the user-provided examples.

Phase 3: Fine-tuned Model. Once sufficient annotations are available, a sequence tagger is fine-tuned on this data, further improving the accuracy of the assistant's suggestions.

Notably, the thresholds for transitioning between phases (e.g., what constitutes "few" or "sufficient" annotations) are configurable. We conduct experiments for determining such thresholds in Section 6.7.5. They can be adjusted based on the dataset's specific characteristics and the user's preferences. Further, the reasoning

is currently supported only during Phases 1 and 2, as the LLM is instructed to provide a reason alongside its prediction.

6.7.4 Benchmarking Sequential Sentence Classification

This benchmark evaluates LLMs and sequence taggers on sequential sentence classification tasks. As outlined in the previous section, both are crucial components of our proposed three-phase approach. We aim to identify the most suitable configuration for integrating practical annotation assistance into DATS. To this end, we carefully select datasets for sequential sentence classification with varying domains and tasks. In this benchmark, we focus only on the single-label classification task, e.g., each sentence is classified into one of many classes.

Models

This benchmark can be understood as an extension of the LLM benchmark conducted in Sections 5.5 and 6.6. The previous benchmark evaluated LLMs for document classification, metadata assignment, and sequence labeling. Consequently, consistent with the DATS model philosophy (Section 4.5), it utilizes the same models: Llama 3.1 (Grattafiori et al. 2024; Touvron et al. 2023), Gemma 2 (G Team 2024), and Mistral NeMo (MA Team 2024; Jiang et al. 2023) with 8B, 9B, and 12B parameters, respectively. We only test instruct fine-tuned models and use half-precision (FP16) variants. Details of these models are provided in Section 2.6.

For fine-tuning experiments, we evaluate a sentence transformer-based sequence tagger. We employ an established model architecture (Huang et al. 2015; Panchendrarajan and Amarasen 2018) consisting of four layers: (1) sentence transformer to compute sentence embeddings, (2) BiLSTM (Hochreiter and Schmidhuber 1997; Graves and Schmidhuber 2005) to capture dependencies between sentences, (3) a linear layer to project from the BiLSTM’s hidden dimension to the number of tags, and (4) a conditional random fields (Lafferty et al. 2001) layer to model dependencies between tags. We compare two embedding models: The default *all-mpnet-base-v2* (Song et al. 2020) with NV-Embed (7B parameters) (Lee et al. 2025), the best-performing model on the English MTEB (Muennighoff et al. 2023) as of December 2024.

Experiment Setup

As in the previous benchmark, we conduct zero- and few-shot experiments with a single, clear prompt across all datasets, avoiding extensive prompt engineering.

Table 6.4 lists the system and user prompt templates used in our benchmark and implementation. The system prompt includes details about the LLM’s role and instructions for formatting the output. The user prompt template briefly explains the task, lists the annotation guidelines (including categories and definitions), and presents the document as a numbered list of sentences. Finally, the key constraint of using categories of the annotation guideline is reiterated. We use structured generation enforcing models to generate a JSON response containing *sentence id*, *classification*, and optional *reasoning*. This drastically improves the parsing of model responses but may slightly impact performance.

System Prompt

You are a professional annotator specialized in annotating sentences of a document with the help of annotation guidelines. You strictly adhere to the guidelines and follow the desired output format.

Output Format:

You MUST answer in this JSON format, but the reason is optional:

```
[
  {
    "text_id": 1,
    "reason": "The sentence provides context for the research.",
    "category": "background"
  },
  {
    "text_id": 2,
    "reason": "The sentence presents the research findings.",
    "category": "result"
  },
  ...
]
```

User Prompt

Please annotate each sentence of the following document.

Annotation Guidelines:

<annotation_guidelines>

Document:

<document>

Remember to annotate every provided sentence.

You MUST use the categories provided in the Annotation Guidelines!

Answer Template

<text_id>: <classification>

<text_id>: <classification>

<text_id>: <classification>

...

Table 6.4: Three-stage *Annotation Assistant* evaluation – prompt template for sequential sentence classification. The prompts are used in the benchmark and implementation. < > are placeholders for dataset-dependent input.

In few-shot prompting, we provide the model with a series of examples. This is done by presenting the following sequence:

1. System prompt
2. Few-shot examples
 - (a) Modified user prompt: a version of the user prompt without the annotation guidelines.
 - (b) Correct answer: the correct labels for the sentences, formatted according to the answer template.
3. User prompt: the full user prompt, including the annotation guidelines

Few-shot examples are taken from the training splits and are computed in advance using a $k - 2k$ sampler similar to the implementation of Ding et al. (2021). For example, in $k = 2$, we find the minimum number of training samples required to satisfy the following constraint: every category must occur at least two but no more than four times. This means that k does not correspond to a fixed number of documents or sentences. Instead, the number of fully annotated training documents used as few-shot examples varies across datasets.

We conduct fine-tuning experiments by training the sequence tagger for up to 100 epochs on the training split of the datasets. We use the AdamW optimizer, gradient clipping, early stopping with a three-epoch patience, and freeze the embedding model’s weights. The BiLSTM has one layer with a hidden dimension of 256.

All experiments were conducted on a single A100 GPU (80GB) and repeated three times per configuration. The results are averaged across runs to mitigate fluctuations. We only report results on the test sets.

Datasets

This benchmark encompasses various domains, including online discussions, conversational dialogues, and scientific and medical abstracts. It covers the tasks of discourse act classification, emotion recognition, and rhetorical role labeling. The dataset taxonomies are listed in the Appendix Tables A.4, A.5, and A.6.

Coarse Discourse: The Coarse Discourse dataset (Zhang et al. 2017) comprises approximately 9,000 Reddit threads with over 100,000 comments annotated for discourse acts. Each comment is classified into one of ten categories, such as *question*, *answer*, or *disagreement*, enabling analysis of conversational flow and argumentation in online discussions.

CSAbstract: The CSAbstract dataset (Cohan et al. 2019) consists of 2,000 computer science research abstracts from Semantic Scholar. Each sentence is annotated with one of five rhetorical roles, like *background*, *objective*, and *method*.

PubMed 200k: The PubMed 200k (Dernoncourt and Lee 2017) dataset comprises approximately 200,000 abstracts from randomized controlled trials in the medical domain. Each sentence is annotated with one of five rhetorical roles similar to CSAbstract.

Dataset		CSAbstract		PubMed200k		CoarseDisc.		EmotionLines		DailyDialog	
Model	Shot	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Llama 3.1	0	35.68	49.67	33.91	60.27	27.64	23.59	23.19	28.61	26.16	30.60
Gemma 2	0	40.02	55.08	52.88	73.32	33.28	32.36	33.26	46.92	31.32	40.83
Mistral	0	39.39	56.63	45.85	70.98	31.64	30.73	18.68	27.64	26.25	31.14
Llama 3.1	2 ↑	37.71	51.96	48.43	71.54	27.54	23.95	23.53	28.94	27.86	33.39
Gemma 2	2 ↓	45.82	60.79	56.93	72.02	18.21	11.77	29.86	33.06	30.85	37.51
Mistral	2 ↑	41.35	57.60	53.95	76.06	31.94	31.47	22.49	28.61	27.46	36.59
Llama 3.1	4 ↓	35.09	50.63	49.22	69.93	26.79	23.11	22.19	27.98	25.83	30.02
Gemma 2	4 ↓	45.78	57.89	50.43	51.93	13.59	08.16	26.85	29.55	22.60	26.44
Mistral	4 ↑	44.57	60.27	54.04	76.95	32.85	31.96	23.99	32.61	27.80	37.86
allmpnetv2	All	37.43	61.23	72.89	85.78	29.15	41.48	26.06	47.14	22.03	51.81
NV-Embed	All	55.65	71.98	81.72	90.20	42.40	51.05	34.86	52.89	31.77	57.67

Table 6.5: Three-stage *Annotation Assistant* evaluation – results of prominent LLMs in zero-shot (0), few-shot (2, 4), and fully fine-tuned (All) settings on five sequential sentence classification datasets. F1-score and accuracy (Acc) are reported. ↑ indicates overall improvement compared to the previous step, ↓ an overall decrease.

EmotionLines: The EmotionLines (Hsu et al. 2018) dataset comprises 2,000 dialogues and 29,245 utterances from Friends TV scripts and Facebook Messenger conversations. Each utterance is annotated with one of eight emotions, including *surprise*, *sadness*, and *joy*.

DailyDialog: The DailyDialog dataset (Yanran Li et al. 2017) comprises 13,118 multi-turn dialogues totaling approximately 100,000 utterances and is designed to reflect everyday conversations across diverse topics. Each utterance is annotated with emotion labels, similar to those in EmotionLines.

Results

The benchmark results are presented in Table 6.5. Gemma 2 exhibits excellent zero-shot performance, but its performance generally decreases with the addition of few-shot examples. Llama 3.1 consistently performs the worst in zero- and few-shot settings, with few-shot examples having a negligible impact. Mistral demonstrates consistently good performance, improving with more few-shot examples. However, even in the 4-shot scenario, it often fails to surpass Gemma 2’s zero-shot performance. Exploratory experiments with more examples in the few-shot learning scenario were inconclusive and could not consistently improve over the 4-shot setting.

As expected, the fine-tuned sequence tagger based on NV-Embed achieves the best results across all datasets, with improvements of up to 15 points in F1-score (PubMed) and 19 points in accuracy (CoarseDiscourse) compared to the best few-shot results. Notably, rhetorical role labeling appears to be an easier task than emotion detection and discourse act classification. The relative simplicity of rhetorical role labeling stems partly from the lower number of classes compared to other tasks. The taxonomies of CSAbstract and PubMed use 5 classes, while

those of DailyDialog, EmotionLines, and CoarseDiscourse use 8, 9, and 10 classes, respectively.

While Gemma 2’s zero-shot performance and the fine-tuned models’ overall performance are encouraging, the few-shot results are less satisfactory. Future work includes investigating alternative approaches to few-shot learning.

6.7.5 Transitioning Phase 2 $\rightarrow$ 3

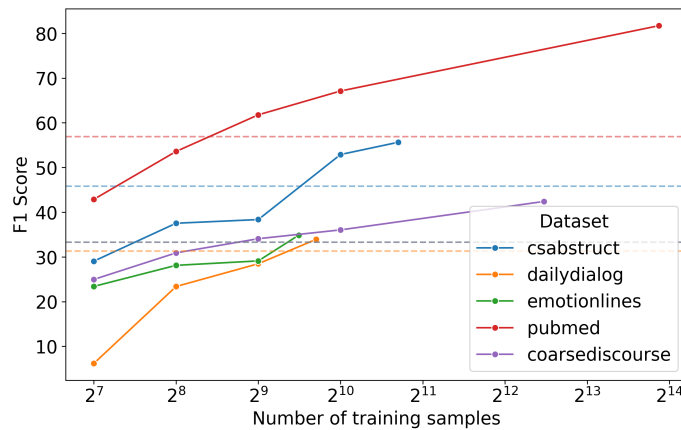


Figure 6.7: Evaluation of NV-embedd-based sequence tagger with increasing number of training examples. The dashed line denotes the highest F1-score achieved by LLMs in few-shot settings. Figure from Fischer and Biemann (2025).

We aim to identify thresholds for transitioning from Phase 2 (few-shot prompting) to Phase 3 (fine-tuning a sequence tagger). The optimal threshold is the number of training examples the sequence tagger requires to surpass LLMs in the few-shot setting. For this, we train the NV-embed-based sequence tagger on an increasing number of fully annotated training examples: 128, 256, 512, 1024, and the whole training split. Figure 6.7 shows the performance improvements of the sequence tagger as the number of training examples increases. The dashed lines correspond to the best F1 Scores achieved by LLMs in few-shot settings, which were evaluated earlier as part of this benchmark. Hence, the minimum number of fully annotated documents needed to train a sequence tagger superior to LLMs is given by the intersection of the dashed line with the corresponding solid line. For example, approximately 380 fully annotated documents are needed to surpass LLMs in the PubMed dataset, about 770 in CSAbstract, 500 in CoarseDiscourse, and 700 in DailyDialog and EmotionLines. As this number is highly dataset-dependent, we decided to make the threshold for transitioning between phases configurable in the user interface, allowing users to adjust it based on their project and preferences. Still, if 1000 documents are annotated, the user interface actively recommends training a sequence tagger.

6.7.6 Integrating Sequential Sentence Annotation in DATS' LLM Assistant

The sequential sentence annotation feature is incorporated into the *LLM Assistant* (Section 5.5 and 6.6) within the DATS platform (Section 4.5) using the existing technologies of LLM-serving (vLLM, Gemma 3) and background-processing (RQ workers). We build on the *LLM Assistant*'s flexible, modular architecture and adapt its frontend to support the three-stage annotation assistance. An additional step is introduced that recommends a zero-shot, few-shot, or model fine-tuning strategy to the user, depending on the number of available annotated examples. This recommendation can be overwritten. This six-step dialog is shown in Alice's workflow in Section 6.7.2, depicted in Figure 6.5. The sentence annotator component is built from scratch in React, also shown in the Alice workflow in Figure 6.6.

DATS offers a semantic-similarity search based on sentence embeddings computed during import. We reuse these pre-computed sentence embeddings to drastically reduce the training and inference time of the sequence tagger model.

6.7.7 Conclusion

This section explored a three-phase approach to AI-assisted coding to enhance qualitative data analysis workflows, focusing on sequential sentence classification within the Discourse Analysis Tool Suite. We designed a benchmark and evaluated the performance of three prominent open-weight LLMs and sequence tagging models based on sentence embedding models. Our findings demonstrate the effectiveness of our proposed three-phase *Annotation Assistant*. Performance steadily improves as more annotated data becomes available when using Mistral. However, few-shot prompting does not scale as well as expected and may require a different approach. We integrated this assistant into DATS, providing researchers with a valuable semi-automatic, sequential sentence annotation tool. We see this integration as a notable step toward democratizing access to AI for data analysis, empowering researchers with a transparent, user-controlled system that augments, rather than replaces, their expertise.

While the proposed *Annotation Assistant* represents a step toward enhancing qualitative data analysis workflows, it is important to acknowledge its current limitations. One limitation is the performance gap between few-shot prompting (e.g., 2, 4, 8, 16 examples) and full fine-tuning (hundreds of examples). This gap can lead to a period during which the assistant's suggestions do not improve significantly, despite ongoing user annotations. Another limitation is the focus on English datasets in our benchmark. It does not cater to the needs of all DATS users, particularly those working with German-language data. Furthermore, the implementation of the *Annotation Assistant* offers limited configurability. While we benchmarked and integrated the best-performing model, users might benefit from the ability to select and configure different models based on their specific needs and preferences. Further limitations regarding LLMs are discussed in the overall thesis limitations in Section 9.2.

Code for replicating the benchmark⁵ is available.

6.8 Excursus: AnnoPlot - Visual Annotation Organization

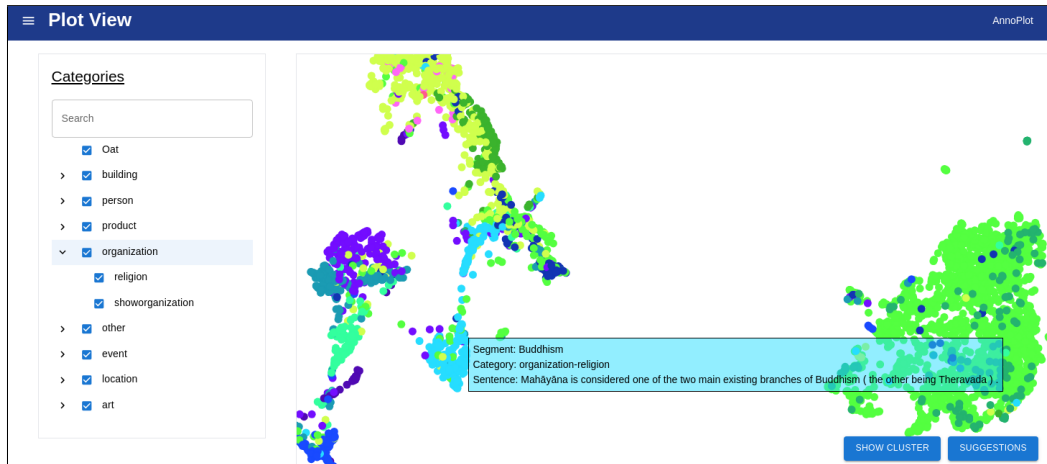


Figure 6.8: *AnnoPlot* Plot View. Left: Overview of the hierarchical category system for filtering the annotations. A tooltip displays related information about the hovered annotation, including the context sentence. Figure from Fittschen et al. (2024).



Figure 6.9: *AnnoPlot* Category View, corresponding to the data seen in Figure 6.8. Notably, person, scholar (right two points), and location, building, library (top three points) cluster together. Figure from Fittschen et al. (2024).

In this chapter, we explored three different variants for generating automatic annotations: a two-stage approach of retrieval and few-shot model training (Section 6.5.3), zero-shot LLMs (Section 6.6), and a three-stage approach of zero-shot prompting, few-shot prompting, and model training (Section 6.7). However,

5. <https://github.com/uhh-lt/seq-sentence-classification>

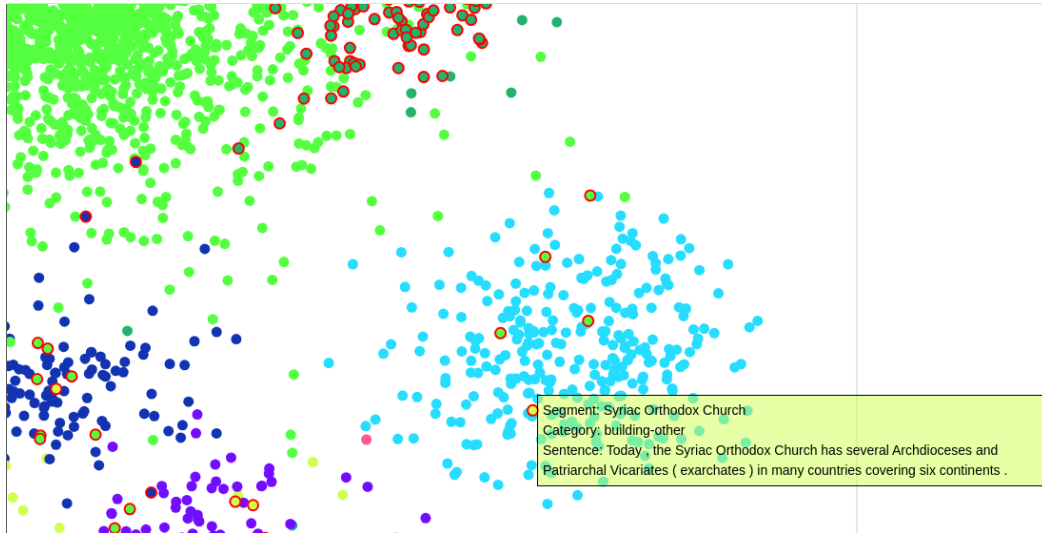


Figure 6.10: *AnnoPlot* Error Identification. The light blue dots are a cluster of religious organizations. A user must now decide if this annotation is correct or incorrect. Figure from Fittschen et al. (2024).

obtaining an overview to manage and, if required, correct annotations becomes necessary when dealing with such large numbers of semi-automatically generated annotations.

In parallel work (Fittschen et al. 2024), we addressed the challenge of data integrity and quality assurance arising from dynamic category systems and the introduction of thousands of semi-automatic annotations.

Coding projects in qualitative research are inherently dynamic, as category definitions evolve and coding strategies are constantly revised. This iterative process, especially when combined with automated scaling, makes it difficult for researchers to maintain a clear overview of the entire annotated dataset and identify potential errors or inconsistencies.

Our work addresses this by introducing *AnnoPlot*, an interactive web application for analyzing, managing, and visualizing annotated text data. The tool provides a novel two-dimensional view of annotation data and its category system. It features:

1. **Annotation Overview:** Annotations (spans) are first encoded into high-dimensional embeddings (BERT), and then reduced to two dimensions using a dimensionality reduction algorithm (UMAP, McInnes et al. (2020)). This scatter plot visually clusters semantically similar annotations, enabling researchers to assess the distribution and consistency of their categories (see Figure 6.8).
2. **Category Management:** In parallel, the tool offers a category view, which visualizes the hierarchical code system as a bubble chart. This allows researchers to manage, organize, and merge codes based on their semantic proximity, supporting the conceptual consolidation phases of selective coding (see Figure 6.9).

3. **Domain Adaptation:** The tool's main feature is a dynamic UMAP implementation. It uses a small neural network trained interactively with user input via triplet loss (to push apart clusters of different categories) or mean squared error loss (to adjust the positions of individual data points). This allows the researcher to fine-tune the visualization in real time, adapting it to the specific domain and their evolving interpretative framework.
4. **Support Error Detection:** The application integrates the HDBSCAN (Campello et al. 2013) clustering algorithm to detect outliers and inconsistent annotations automatically. This feature flags annotations that fall into clusters belonging predominantly to a different category, providing the researcher with a guided mechanism for quality control and correction (see Figure 6.10).

With these features, *AnnoPlot* offers a solution for the management, overview, and error correction of annotated datasets, particularly those generated through the semi-automatic scaling methods proposed in this chapter. It empowers the discourse researcher to continuously refine their annotations using advanced visualization and quality control techniques, which are another step towards bridging qualitative and quantitative analysis.

6.9 Conclusion & Discussion

This chapter established the necessity of digital support for coding in discourse analysis (Section 6.1). Coding serves as the bridge between raw text corpora and the reconstruction of discursive formations. However, in an era of massive digital data volumes, assistance with coding through digital tools is important for researchers to identify, code, and manage discursive knowledge effectively, while maintaining the iterative, interpretive nature of qualitative methodologies.

The inherent scalability limitations of manual coding for large data volumes constrain the coding process (Section 6.2). Further practical challenges include the difficulty of managing the volume of annotations (whether derived from semi-automatic methods or in-vivo coding), and the need for transparency of the interpretive coding and the underlying algorithms to make the decisions traceable. There are conflicts (Section 6.3), but also opportunities (Section 6.4), between the dynamic, bottom-up coding process of Grounded Theory and the requirements of scalable computing methods.

DATS provides basic annotation capabilities, including hierarchical taxonomy creation, collaborative span-level text annotation, and automatic named entity recognition (Section 6.5). However, DATS' core functionalities address neither scalability nor user control over automated processes. In practice, the density of automated NER suggestions often disrupts the user's workflow, leading researchers to turn off the feature frequently and limiting the platform's potential for efficient analysis.

We subsequently presented two key contributions to the DATS platform. We first described the span-annotation capabilities of the *LLM Assistant* (Section 6.6) and subsequently examined its functionality for sequential sentence classification (Section 6.7). Specifically, we benchmarked small, efficient, open-weight LLMs

for sequence labeling and sequential sentence classification, then integrated the best-performing models into the *LLM Assistant*. For the task of sequential sentence classification, we evaluated a three-phase annotation approach. It dynamically transitions from zero-shot prompting (for initial suggestions) to few-shot prompting (for refining suggestions with user examples) and, finally, to a fine-tuned sequence tagger (for high-accuracy corpus-wide application). We subsequently integrated this three-phase approach into the *LLM Assistant* and implemented a sentence annotator component with a compare view to facilitate collaboration and discussion.

In parallel work, we introduced two approaches to address scalability and annotation quality challenges. The *Annotation Scaling* excursus (Section 6.5.3) detailed a system that uses contextualized embeddings, indexing, and transformer adapters to scale annotations from qualitative findings to quantitative results efficiently. The *AnnoPlot* excursus (Section 6.8) then introduced a visualization tool for the management and quality assurance of annotated datasets that uses a dynamic UMAP to adapt to the researcher’s interpretive framework. It further provides an overview via both the annotation view and the category bubble chart, and employs HDBSCAN for automatic error detection in the annotated data.

In conclusion, this chapter demonstrated the successful application of advanced NLP techniques to address the scalability and efficiency challenges in qualitative discourse analysis. These include contextualized embeddings for a semantic search retrieval approach, various LLM prompting strategies (zero-shot, few-shot), and model training with and without adapters. However, major challenges remain, particularly in making the annotation process transparent and effectively supporting the handling of multimodal data.

Discussion

The main contributions of this chapter — the *LLM Assistant* (Section 6.6) and the *Annotation Assistant* (Section 6.7) — and the parallel work of *Annotation Scaling* (Section 6.5.3) and *AnnoPlot* (Section 6.8) are designed to enhance the annotation workflow of digital discourse analysis. This section discusses these contributions with respect to the technical requirements of a discourse analysis tool suite introduced in Section 1.5.

The two mechanisms of this work target the central technical challenge of *scalability*. First, the *Annotation Scaling* framework proposes using approximate nearest neighbor indexing for high-speed semantic search over massive corpora to enable semi-automatic annotation by accepting or rejecting suggestions. Furthermore, it employs transformer adapters to enable model training with minimal data and resources. In addition, the three-phase *Annotation Assistant* ensures scalability throughout the workflow: while the LLM-based phases (zero- and few-shot) provide immediate assistance on individual documents, the final phase (a fine-tuned sequence tagger) provides the necessary machine learning model for high-speed, corpus-wide application.

Here, selecting small, efficient open-weight LLMs and using optimized libraries (such as vLLM with KV caching) satisfies the *responsiveness* requirement by minimizing inference latency, thereby ensuring a smooth, interactive user experience.

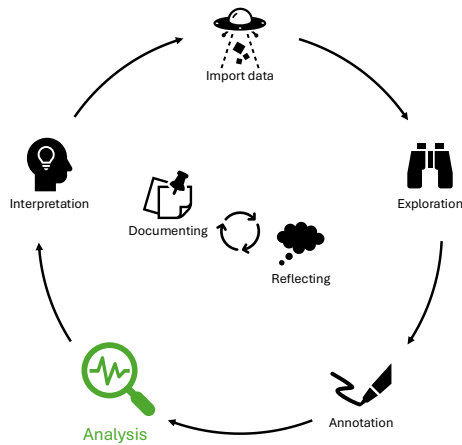
The iterative, data-driven nature of Grounded Theory demands high *adaptability*. This is mainly achieved through few-shot learning. First, the *Annotation Scaling* framework uses rejected suggestions from Stage 1 to adapt quickly, then trains transformer adapters with few labeled examples in Stage 2. Second, the dynamic phase transition in the three-phase *Annotation Assistant* is inherently adaptive, shifting from zero-shot to few-shot to fine-tuned modes as the user's annotated example dataset grows, ensuring the model's performance continuously adapts to the researcher's evolving understanding. In particular, the few-shot in-context learning in Stage 2 leverages a small number of user examples to adapt quickly. Finally, *AnnoPlot* enables users to actively manipulate the visualization to suit their needs by updating a dynamic UMAP model with minimal feedback.

The contributions adhere to the human-in-the-loop principle, prioritizing *user control* over automation. Thus, all automated annotations and suggestions provided by the *LLM Assistant* and *Annotation Assistant* require explicit user acceptance or rejection, ensuring that the researcher retains full authority over all analytical decisions. *Transparency* is partially addressed by providing insights into the system's decision-making logic. The *LLM Assistant* and *Annotation Assistant* allow users to inspect and modify the underlying prompts and display the LLM's reasoning (memos) alongside its suggestions. The envisioned *Annotation Scaling* system also relies on manual review, allowing users to accept or reject the system's suggestions and ensuring they remain in control.

The approaches presented address *flexibility* by moving away from specialized models. First, the assistants rely on LLMs' general-purpose reasoning capabilities, allowing users to apply the annotation function to any new domain-specific code system immediately, without training. Both *AnnoPlot* and *Annotation Scaling* use generic encoder models (e.g., BERT embeddings), which are broadly applicable and can be easily replaced with other language-specific models.

Finally, all interfaces presented in this chapter were designed with *accessibility* in mind. We designed easy-to-use interfaces that abstract away the complex NLP logic underlying these systems, aiming to be highly usable for users without machine learning knowledge.

While the coding process provides the foundational "micro-level" evidence by labeling text segments, the research goal of a discourse analysis extends beyond individual documents to broader societal patterns. Once a sufficient base of qualitative codes has been established, the researcher faces the challenge of scaling these insights to visualize long-term trends and comparative shifts across the entire corpus. This transition from close reading to distant reading moves the research into the phase of quantitative and mixed-methods analysis. The following chapter presents methods to bridge this gap, specifically exploring how annotated concepts can be tracked over time to reveal the evolution and changes of a discourse.



7

Analysis

Contents

7.1	Motivation	172
7.2	Challenges & Risks	173
7.3	Analysis in DATS	174
7.3.1	Workflow with Alice	175
7.3.2	Addressed Challenges	177
7.4	Concept Over Time Analysis	177
7.4.1	Introduction	177
7.4.2	Related Work	179
7.4.3	Workflow with Alice	180
7.4.4	Experiments	182
7.4.5	Implementation	185
7.5	Conclusion & Discussion	187

This chapter explores the analysis phase of the discourse analysis workflow, focusing on quantitative and mixed-methods procedures that allow for the computer-aided evaluation of large-scale research materials. We begin by establishing the motivation, challenges, and risks of quantitative scaling (Sections 7.1 and 7.2). Section 7.3 outlines the baseline analysis features in DATS, such as word and code frequency statistics, as demonstrated through Alice’s ongoing pandemic discourse project.

The core of this chapter (Section 7.4) introduces the *Concept Over Time Analysis* (COTA). COTA (Fischer, Schneider, Geislinger, et al. 2024) facilitates an iterative, human-in-the-loop process of defining concepts, refining them, and visualizing their trends.

By implementing COTA, this chapter addresses Research Questions 4 and 6, demonstrating how semantic search and manual annotation can be tightly coupled to uncover temporal patterns, thereby bridging individual text passages and overarching discursive trends.

7.1 Motivation

In this chapter, the term analysis refers to quantitative and mixed-methods procedures (Section 3.1.2) that encompass the (semi-) automatic, computer-aided evaluation of research materials. This is to be distinguished from the qualitative fine-grained analysis of the interpretation phase of discourse analysis. Interpretation will be treated separately in Chapter 8.

While qualitative approaches aim for in-depth understanding, the quantitative paradigm in empirical social research primarily focuses on measurement, hypothesis testing, and generalizing results. Within this framework, social reality is captured through standardized numerical data that enable statistical evaluation and the identification of broader patterns across a representative sample (Baur and Blasius 2022b, p. 13).

Discourse analysis is fundamentally an interpretative and qualitative procedure. Nevertheless, quantitative analyses are part of many research projects. Discourse analyses investigate either extended periods of time (the historical method) or differences between orders (the comparative method). The historical method aims to elaborate and reconstruct the historical emergence of phenomena (Traue et al. 2022, p. 628). Through the analysis of word or document frequencies, discourse continuities and discontinuities can be recognized, which manifest themselves in the phenomena under investigation over extended periods or differences. Keller (2010) states that qualitatively significant transformations often arise from a sum of quantitative deviations, which can be made visible through statistical methods. Quantitative methods may be used for preparatory work, to complement qualitative analyses, or as an independent method to combine the strengths of both paradigms.

The necessity of quantitative approaches is particularly highlighted by Lexicometric Discourse Analysis (following Pêcheux (1995)). This method investigates changes in the frequency of specific words or word fields in large corpora and is particularly suited to analyzing mass media meaning production. Such procedures are increasingly important in the era of open data and big data, as extensive digital corpora become more accessible.

The fundamentally interpretive-qualitative research program of the Sociology of Knowledge Approach to Discourse (SKAD) also recognizes the usefulness of quantitative procedures as supplementary tools for guiding qualitative work. Quantitative steps can be used to select and structure material for screening extensive discourse materials. They serve to justify selecting documents for subsequent qualitative analysis by using the frequency of topics, actors, or linguistic elements to estimate the reach/scope of a discourse. This way, research materials can be reduced from unmanageable complexities to manageable excerpts.

Keller (2019, p. 38) gives a concise example of the mixed-methods application of frame analysis: Here, fundamental framing strategies and elements were initially qualitatively determined in exploratory preliminary work. Building on this, subsequent quantitative studies of mass media reporting required a high coding effort. In these studies, the frequencies of occurrence of specific frames and their temporal progression were analyzed. Although such analysis procedures are not at the core of SKAD, this demonstrates Keller's openness to practice-

relevant analysis procedures that combine qualitative and quantitative methods. We will see in Section 7.4 that this approach to frame analysis highly motivated the main contribution of this chapter.

7.2 Challenges & Risks

Given the size and complexity of contemporary datasets, manual analysis methods may reach their limits. It becomes difficult to recognize meaningful patterns in the vast amounts of text data available. One goal of discourse analysis is to analyze regularities and structural connections within discourses. To meet this challenge, quantitative analyses and visualizations can serve as a solution by making patterns visible, for example, in the following exemplary dimensions:

- **Temporal Trends and Frequencies:** The analysis of concepts, topics, sentiments, opinions, etc., over time can be relevant for discourse analysis to uncover discourse continuities and discontinuities. The historical method aims to elaborate on and reconstruct the emergence and development of objects. Here, a temporal perspective can help identify patterns, trends, and shifts, or reveal how ideas, narratives, or cultural practices evolve, providing insights into the dynamics of discourse.
- **Co-occurrences of Codes:** The analysis of co-occurrences (e.g., *Which classifications appear together with which rules of formation?*) is interesting and insightful. Researchers want to elaborate on the boundaries and transitions between discourses as well as the power relationships between interpretative patterns. Quantitative procedures (e.g., correlation or co-occurrence analyses) can help to test the empirical plausibility of these relationships (e.g., *"How often does interpretive pattern X appear together with speaker position Y?"*).
- **Network Visualizations:** The visualization of relationships between actors, topics, and positions across the entire corpus can be interesting for identifying relationships. These visualizations can help map the boundaries of the discourse arena and reveal which actors or institutions play a central role in articulating specific positions within it.

The central methodological challenge for discourse researchers, however, lies in the qualitative recourse: the statistical evaluation must be interpreted through a discourse-analytical lens. The purely numerical information is not a discourse finding. Therefore, a tool that, for example, outputs co-occurrence matrices must necessarily lead researchers back to qualitative interpretation.

Closely linked to the necessity of qualitative recourse is the challenge of *traceability* (Section 1.5). The transparency and traceability between the generated visualizations and the underlying data must be ensured. This is an important technical challenge that quantitative analysis functions must address to enable qualitative interpretations.

Suppose the visualizations are not directly linked to the specific data segments they represent. In that case, it is difficult for researchers to trace the patterns they

observe back to the source material, hindering the verification of findings and potentially leading to misinterpretation. For example, a network diagram might show a strong connection between two concepts. Still, without a clear link back to the specific instances in the text where those concepts co-occur, it becomes difficult to assess the validity of that connection. Therefore, ensuring that visualizations are directly and transparently linked to the underlying data material is key to maintaining the integrity and trustworthiness of discourse analysis research.

Risks

The potential of quantitative analysis, however, comes with significant methodological risks that discourse researchers must be aware of, and that integrated tools should ideally address. These risks primarily relate to misinterpretation arising from the difference between the qualitative goals of discourse analysis and the quantitative reality of the data.

The interest of SKAD lies not in enumeration or statistics, but in the analysis of knowledge (*What is constructed as reality?*), the rules (*How is something made sayable?*), and the effects of discourses (the power/knowledge relationship) (Keller 2011b). A simple count of words, concepts, or codes (e.g., *The term "climate catastrophe" appears 500 times*) misses the qualitative content and structural formation of the discourse. Discourse analysis is not primarily a logic of frequency, but rather a logic of formation (the analysis of structural contexts and their interconnections). For instance, the interpretive pattern of *"climate resilience"* does not need to be 100 times more frequent than another (e.g., *"climate catastrophe"*) to be the dominant one, especially if the former consistently appears within all highly institutionalized texts, such as official government policy documents. This poses a challenge for tools designed to visualize discourse: how can they display frequencies, statistics, or trends without misleading users to equate frequency with relevance or power?

A second risk involves selecting the source material. SKAD, like other qualitative methodologies, uses theoretical sampling (Section 3.2.2), in which the developing theory guides the selection of documents and focuses on specific, insightful cases. Consequently, the resulting corpus is typically not representative in the statistical sense. This qualitative approach conflicts with the demands of quantitative inference. Any quantitative statement intended to generalize beyond the specific, analyzed corpus (e.g., a claim that *"the use of a specific metaphor increased across mass media between 2010 and 2020"*) requires a corpus that was initially constructed systematically and is demonstrably representative of the population or discourse field under investigation. Thus, to answer quantitative questions, a representative corpus must be created initially, which is a significant methodological difference.

7.3 Analysis in DATS

This section illustrates how to use the basic quantitative analysis features in DATS. The technical implementation of these features is explained in Section 4.3. The features are primarily intended for overview, exploration, and basic

evaluations of the corpus and the coding effort. Alice will continue to guide us through her research workflow.

7.3.1 Workflow with Alice

Throughout her iterative research process (which includes importing new documents, organizing, searching, and filtering these materials, and qualitative coding), Alice can make use of the quantitative analysis features of DATS. These analysis functions may help Alice gain an initial overview of the material and validate her coding framework.

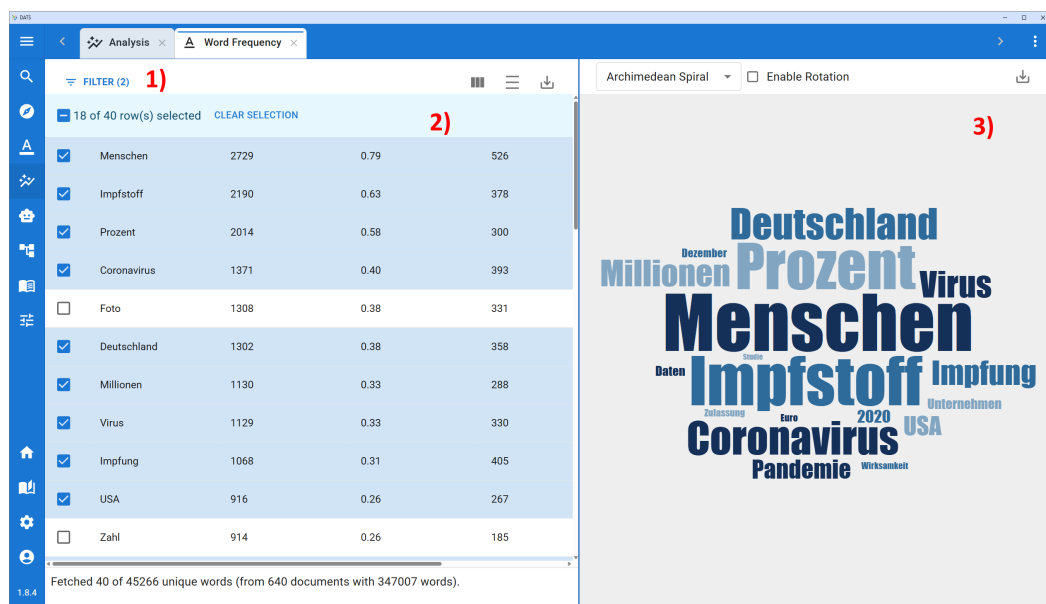


Figure 7.1: DATS word frequency analysis interface. Search and filter functions (1) restrict the analysis to a certain sub-corpora. The table (2) shows absolute, relative, and document frequencies. The word cloud (3) is automatically generated upon selecting words.

Alice can use the word frequency analysis feature for an initial exploration of the corpus and to check for general shifts in key terminology. This feature (see Figure 7.1) filters stop words and aggregates term frequencies across the entire corpus or a filtered subset. The results can be visualized using a word cloud.

Alice can use the search and filter functions (1) of DATS to define a relevant sub-corpus (e.g., *"all documents from 2020 to 2021"*) and then run the analysis. The table (2) lists all words, their absolute frequencies, their relative frequency, and their document frequency. The word cloud (3) can be generated by selecting words from the table and highlights the most salient terms being discussed in that period. By comparing the word clouds for 2020 and 2023, she can detect continuities and discontinuities (e.g., the term *"risk"* being replaced by *"pandemic"*).

To validate her previous coding effort, Alice then utilizes the code frequency analysis function. This feature (see Figure 7.2) aggregates and visualizes annotation counts across the entire corpus or a selected sub-corpus using interactive bar and pie charts.

This analysis allows Alice to assess the distribution of her codes. Alice can use the filtering options (1) to segment the results by modality (e.g., only codes

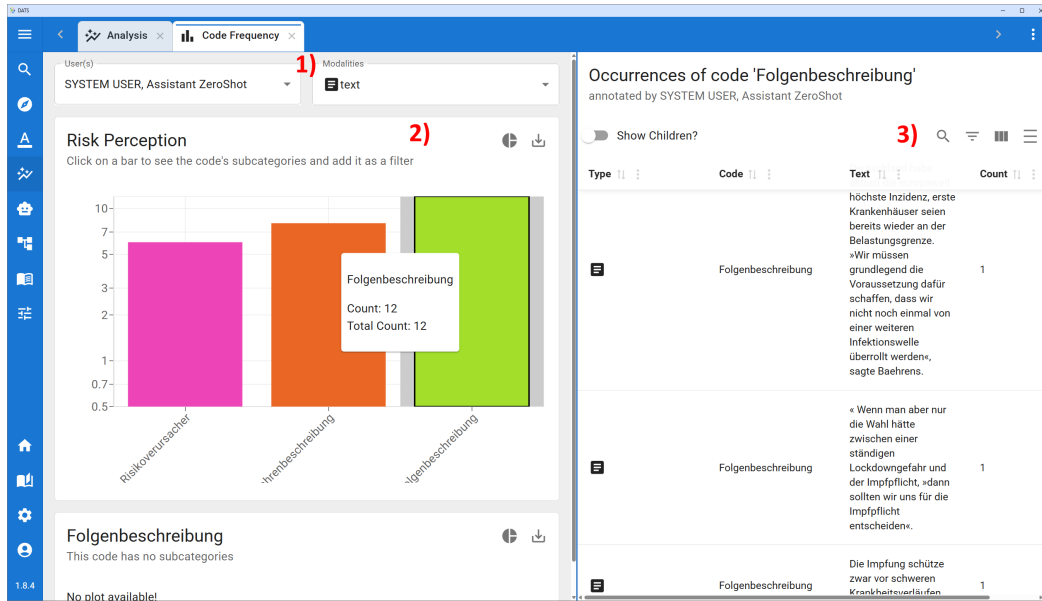


Figure 7.2: DATS code frequency analysis interface. Filter functionalities (1) restrict the analysis to certain users and modalities. The bar chart (2) shows absolute code frequencies. The table (3) lists the text passages assigned to the currently selected code.

applied to text documents) or by annotator, allowing her to compare collaborative codings. The charts on the left (2) show the frequency of code assignments (how often each code was used). The table on the right (3) shows the occurrences of the currently selected code (to which text passages a specific code was applied).

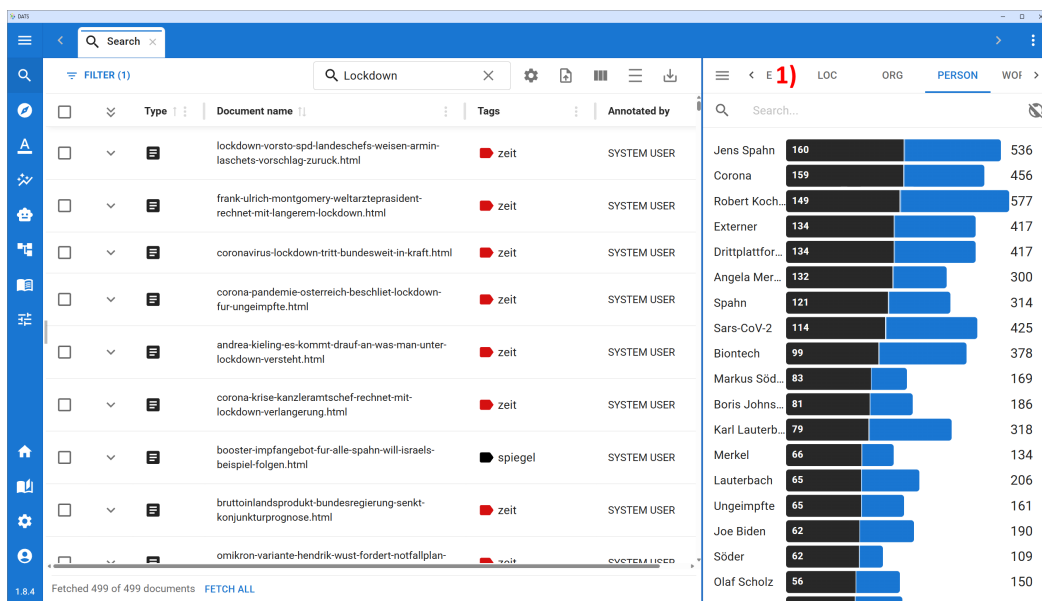


Figure 7.3: DATS search interface with contextual statistics expanded on the right (1). The blue bar (the right count) denotes the total number of occurrences of a certain entity, while the black bar (and the middle count) denotes the number of occurrences in the current search result.

Alice can also use the contextual statistics to support her search and filtering process (see Figure 7.3). DATS provides quantitative analyses that are applied dynamically to the current set of search results.

This is useful when Alice filters her documents to focus on a specific sub-discourse (e.g., all documents that mention a certain vaccination and were published by a particular author). Within the search interface in the right sidebar (1), DATS displays a view of named entities, keywords, tags, and codes for that set of documents. This allows her to get a quick overview of key statistics, such as the most frequently mentioned entities (e.g., persons, organizations) within the current filtered search set.

7.3.2 Addressed Challenges

The analytical functions in DATS (word frequency analysis, code frequency analysis, and contextual statistics) address some methodological challenges identified in Section 7.2, particularly those related to data overview and traceability.

All analysis features provide basic support for *traceability*. For example, the dynamic link between the contextual statistics and the currently filtered documents ensures that quantitative observations can be traced back to the specific source material. Furthermore, all features support a distant reading approach. The word frequency analysis provides a basic mechanism for visualization and pattern recognition, enabling lexicometric shifts to be made visible. The code frequency analysis also helps researchers check the distribution of their interpretive coding framework.

However, the analysis features in DATS are intentionally basic, enabling mostly exploratory quantitative analysis limited to simple frequency counts. Specifically, DATS' core analysis features do include any of the sophisticated quantitative analysis features outlined in Section 7.2 such as temporal trends, co-occurrence analysis, or network visualizations. Furthermore, while mixed-methods analysis offers significant potential, the tool's basic features do not include such capabilities. Addressing these limitations is a primary motivation for the feature developed and presented in the next section.

7.4 Concept Over Time Analysis

7.4.1 Introduction

This section introduces the *Concept Over Time Analysis* (COTA), a machine-learning-based feature within DATS that identifies concepts of interest in large datasets and visualizes their temporal occurrences. The section is based on Fischer, Schneider, Geislinger, et al. (2024), who first detailed the system. COTA allows users to define concepts, helps them identify relevant sentences, refines concept representations, and finally visualizes the results over time. The interactive user interface follows the human-in-the-loop paradigm (Holzinger 2016), encouraging users to provide feedback by annotating samples, thereby improving concept representations and analysis results.

The proposed system performs few-shot sentence classification, mapping each user-defined concept to a class. We use SentenceTransformers (Reimers and Gurevych 2019) to compute representations of sentences and concept descriptions. Initially, a similarity search returns semantically relevant sentences for each concept, which the user can review and accept, effectively labeling the sentences and providing feedback to the system. With a few such labeled examples, a contrastive dataset is generated to fine-tune a sentence transformer model. This is employed to update concept representations and similarity search results, providing the user with better suggestions. This iterative refinement step is repeated until the user is content. Finally, sentences classified to concepts with confidence above a specified threshold are visualized in a timeline.

COTA facilitates a mixed-methods approach to large-scale datasets by mapping user-defined concepts—such as events, topics, and practices—onto a timeline. This temporal perspective moves beyond static observation, allowing researchers to identify the evolutionary dynamics of discourse and the shifts in narratives as they unfold. By visualizing these concepts, *COTA* may trace the lineage of ideas, illustrating how past discourses influence contemporary ensembles. For example, it can help researchers in empirical cultural studies explore the interplay between various cultural elements, such as texts or social practices, and how the dispositif shapes them and creates the discursive ensemble. It can draw the researcher’s attention to relevant passages from a particular time period. Though inspired by discourse analysis, the proposed system is versatile, extending its utility to related analyses such as visualizing topics or sentiments over time.

While *COTA* is a powerful feature on its own, its integration within DATS enables users to merge findings with existing analyses, use memos or the logbook for documentation and reflection (Section 8.3), and identify concepts to be analyzed through exploratory (Section 5.4) or search and filter (Section 5.3) functionalities.

The development of the *Concepts Over Time Analysis* addresses two research questions of this thesis:

Research Question 4

How can semantic search and manual annotation be integrated into an interactive process to facilitate the semi-automated classification and analysis of concepts over time?

Research Question 6

How can AI-augmented methods—including visualization, annotation, and semantic search—be effectively designed within a human-in-the-loop framework to incrementally and rapidly adapt to evolving concepts and domain-specific inquiries in qualitative discourse analysis projects despite the scarcity of labeled data?

The underlying task tackled by this work is sequence classification (see Section 2.5.4), a supervised learning task in NLP. In particular, this work addresses sentence classification, aiming to automatically assign a user-defined label to a new, unseen sentence. With the emergence of pre-trained sentence embedding models (see

Section 2.5.4), where language models are pre-trained on large datasets of sentence pairs, meaningful sentence representations are learned, which facilitate interesting approaches to few-shot classification.

The contributions are:

1. We introduce a flexible approach to sentence classification for user-defined concepts that combines (a) semantic search to identify relevant sentences, (b) a feedback mechanism to enable users to review and annotate sentences quickly, and (c) few-shot fine-tuning of a sentence transformer embedding model (Tunstall et al. 2022) to increasingly better align the concept representations with the researchers' analytical goals.
2. We evaluate this approach by conducting experiments on three datasets to demonstrate its feasibility.
3. We present *Concept Over Time Analysis*, an interactive system within DATS for human-in-the-loop sentence classification with iterative model refinement. *COTA* facilitates concept identification in large datasets and provides a chronological overview.
4. We demonstrate the practical utility of *COTA* for discourse analysis by describing an envisioned workflow in the Discourse Analysis Tool Suite that outlines how to interact with the system. The workflow shows how scholars can use the system's interactive capabilities to semi-automatically annotate large text corpora and uncover trends in the materials, identifying relevant points in time for subsequent in-depth qualitative analyses.

Concepts

Within the framework of Grounded Theory, concepts are defined by Strauss et al. (1996) as the basic building blocks of a theory, identified and developed (including their properties and dimensions) through the analytical process of coding (Section 3.4.1).

In the specific context of this work, we employ a functional interpretation of the term *concept* to encompass various qualitative usage scenarios. Unlike philosophical definitions (e.g., Platonic ideas) or linguistic ones (e.g., synsets of WordNet, Miller 1995), our definition is strictly operational: a concept is any entity that semantically groups or classifies sentences. For example, in an analysis of COVID-19 discourse, a concept could represent a thematic category (e.g., "*Vaccination Policy*"), a specific sentiment (e.g., "*Skepticism*"), or a recurring opinion (e.g., "*Economic Impact of Lockdowns*"). This broad application allows the *Concept Over Time Analysis* to handle diverse analytical targets.

7.4.2 Related Work

Qualitative data analysis tools frequently used in DH include CATMA, MAXQDA, and Atlas.ti (Section 4.2). While offering various analysis tools, at the time of writing, none of them include means to identify concepts in large datasets semi-automatically or to visualize them in a chronological overview. With the proposed *Concept Over Time Analysis*, DATS aims to close that gap.

However, many tools and libraries offer useful capabilities for timeline analysis. BERTopic (Grootendorst 2022), a topic modeling tool, leverages sentence transformers to compute document or sentence embeddings, yielding easily interpretable topic representations through clustering. With dynamic topic modeling, the tool provides a set of techniques for analysing how topics develop over time. While BERTopic automatically identifies prevalent topics, *COTA* empowers users to define and analyze their own concepts or topics of interest over time. SCoT (Haase et al. 2021) is an interactive web application to analyse the sense clusters of a word and their evolution over time. SentiView (C Wang et al. 2013) is an interactive visualization system for sentiment analysis. By employing a time-varying helix in conjunction with an attribute astrolabe to depict sentiments, it visualizes changes over time in various attributes and relationships among the demographics of interest, as well as participants' sentiments on popular topics. Open Discourse¹ is a web platform for the analysis of the plenary minutes of the German federal parliament. It includes a topic analysis feature that allows investigation of political discourse over time, with filtering by speaker, gender, party, and other attributes. The existence of such tools highlights the digital humanities' interest in timeline analysis.

Other related work includes tools for few-shot classification. LabelSleuth (Shnarch et al. 2022) is a no-code platform that makes NLP accessible to a broad audience. It facilitates training integrated binary classification models and an intuitive, active-learning-powered annotation interface. In contrast, *COTA* supports training multi-class sentence classification models and enables analysis of results over time.

7.4.3 Workflow with Alice

COTA is an interactive, machine-learning-based tool that semi-automatically identifies relevant sentences for user-defined concepts, adjusts its results based on user feedback, and visualizes concept occurrences over time. Its goal is to provide users with a chronological overview of their concepts of interest, with as few steps as possible. Integrating the feedback process into the workflow was a key requirement for developing the user interface.

The *Concept Over Time Analysis* feature is applicable to a wide range of inquiries and domains. We illustrate the semi-automatic annotation and timeline analysis of the *COTA* feature through a typical, idealized workflow. The technical implementation is explained in the following sections. We will continue to follow Alice as she guides us through her research workflow.

Previously (in Section 5.5.3), Alice used the *LLM Assistant* to automatically extract metadata and identify the publication date for every article. This metadata now enables her to investigate temporal shifts and trends, or to identify key turning points in her dataset of German COVID-19 discourse—for example, pinpointing when framing shifted from *individual risk* to *societal burden*, or when debates around specific vaccines peaked. Alice launches the *Concept Over Time Analysis* to analyze such dynamic patterns.

1. <https://opendiscourse.de/>

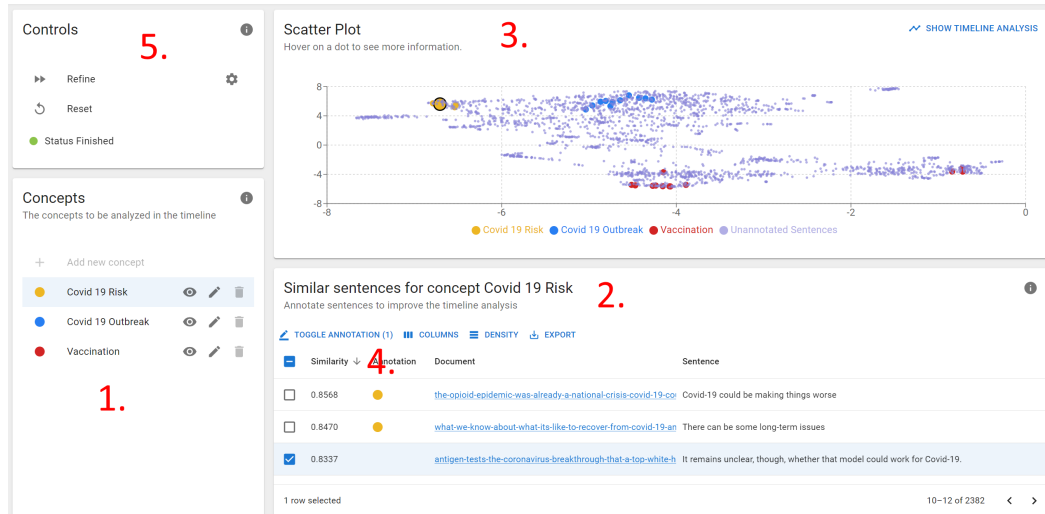


Figure 7.4: *Concept Over Time Analysis'* annotation interface: (1) Define, edit, delete & toggle visibility of concepts. (2) & (4) Sentence annotator: a ranked list of sentences similar to the selected concept. (3) 2D visualization of the search space, annotated sentences are colored & highlighted. (5) Controls to manually start, reset & configure model training. Training status is updated live. Figure from Fischer, Schneider, Geislinger, et al. (2024).

Defining Concepts: Alice is presented with the annotation view of *COTA* (see Figure 7.4). To initiate the analysis, she defines three concepts: "*Covid-19 as a risk*", "*Covid-19 Outbreaks*", and "*Vaccinations*", and provides a short description for each (1). To identify such concepts, DATS offers various features, ranging from search & filtering (Section 5.3) and statistical analysis (Section 7.3) to qualitative reading and coding of documents (Section 6.5).

Concept Annotation: After inputting concepts and descriptions, Alice starts the first iteration of the analysis. The sentence annotator (2) displays ranked sentences based on similarity to the concept, accompanied by a scatter plot (3) visualizing all sentences in the search space. Clicking on a dot in the scatter plot navigates to and highlights the corresponding sentence in the annotator. While exploring the search space, Alice realizes that the scatter plot lacks effective clustering for her concepts, primarily because, at this stage, a concept's representation relies solely on the initial description.

Alice enhances her defined concepts by providing additional information through sentence annotations. While scrolling through lists of similar sentences, she codes all fitting sentences with the corresponding concept (4), identifying relevant mentions, ideas, or paraphrases. This initial coding is recognized as the most challenging part of the analysis, and we aim to assist by providing an initial similarity ranking.

Iterative Concept Refinement: After coding the minimum required sentences per concept (the default is 5), she proceeds to the next step by clicking the refine button (5). Based on her feedback, the system improves its ability to distinguish between concepts and provides more relevant, similar sentences for each. The scatter plot updates and shows clusters of sentences centered on the annotated

ones. Annotated sentences are in Alice’s defined color, while others are visualized in purple (3). With the improved scatter plot and sentence rankings, it becomes notably easier for Alice to identify additional relevant sentences for her concepts.

After coding a few more relevant sentences, Alice can refine the analysis again. Alice can continue this process until she’s certain that the system can distinguish between concepts. Typically, this is achieved when the scatter plot shows a clear clustering and when most top-similarity sentences for a concept are deemed relevant.

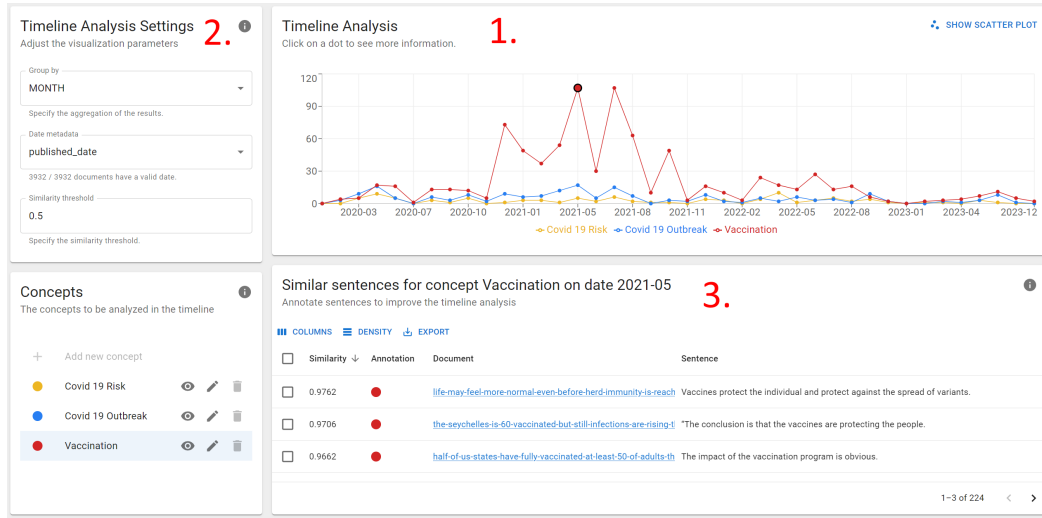


Figure 7.5: *Concept Over Time Analysis* timeline interface: (1) Aggregated occurrences of sentences similar to concepts over time. (2) Settings. (3) Sentence annotator showing similar sentences of the selected concept on the selected date. Figure from Fischer, Schneider, Geislinger, et al. (2024).

Timeline Analysis: Satisfied with the clustering and the suggested similar sentences, Alice switches to the timeline analysis visualization (see Figure 7.5). This visualization (1) depicts the date on the x-axis and the aggregated count of similar sentences per concept on the y-axis, illustrating the development of her defined concepts over time. Alice customizes the aggregation (year, month, day) and adjusts the similarity threshold to filter out irrelevant sentences (2). For example, setting the threshold to 1.0 ensures only sentences explicitly annotated by Alice are considered for aggregation and visualization. Alice explores the results by clicking on several dots, which reveal all sentences of the selected concept on the chosen date in the sentence annotator (3).

7.4.4 Experiments

Few-shot fine-tuning of sentence transformers has been shown to achieve high accuracy with only a few labeled data points by Tunstall et al. (2022). Our following experiments assess the suitability of this approach for the *Concept Over Time Analysis* and provide insights into its limitations.

In addition to the requirements established in Section 1.5, we formulate the following requirements that our system should meet: The system should (a) require

only few labeled samples to enable users to achieve satisfactory results quickly, (b) ensure short training time to facilitate an interactive user experience with fast iterations, and (c) improve performance by using more training samples rather than spending more time training, enabling iterative model refinement.

Datasets

20 Newsgroups: The 20 newsgroups dataset² (NG20), containing around 20,000 newsgroup posts across 20 topics, aligns with our users' domain. Given that concepts are commonly closely related and within the same domain, we align this experiment with the expected real-world setting by considering the classes "*misc*", "*guns*", and "*mideast*" in the politics subject.

News Categories: The News Category dataset (NCT) (Misra 2022) contains around 210,000 news headlines from 2012 to 2022. Similar to the 20 newsgroups dataset, we consider the related classes "*wellness*", "*style & beauty*", and "*healthy living*" to align the experiments with expected use cases.

Stanford Sentiment Treebank: The Stanford Sentiment Treebank dataset (SST-5) (Socher et al. 2013), containing approximately 12,000 sentences from movie reviews, involves fine-grained sentiment classification with five labels ranging from "*very negative*" to "*very positive*". We explore this dataset to assess the applicability of our proposed system to sentiment analysis, a common need in DH research.

Experiment Setup

Following Tunstall et al. (2022), we fine-tune a sentence transformer model with few-shot training data and train a logistic regression classifier on top. We use paraphrase-mpnet-base-v2³ for all experiments.

The contrastive training data comprises sentence pairs, where a pair is deemed positive (1.0) if both sentences belong to the same class and negative (0.0) otherwise. Considering C classes with N sentences per class, all combinations $(N \times |C|)^2$ are evaluated, but duplicate $(A, B) = (B, A)$ and identical (A, A) sentence pairs are removed. To balance positive and negative pairs, positive examples are oversampled, resulting in a substantial training set even with limited labeled samples.

We mimic the iterative concept-refinement step (usually performed by the user) by incrementally increasing the labeled training data. However, our experiments maintain a uniform label distribution in the labeled data, a condition that is not guaranteed in real-world settings.

The model is fine-tuned for one epoch with cosine similarity loss, utilizing the AdamW optimizer with default parameters and a batch size of 16. Experiments are conducted on a single A100 GPU (80GB).

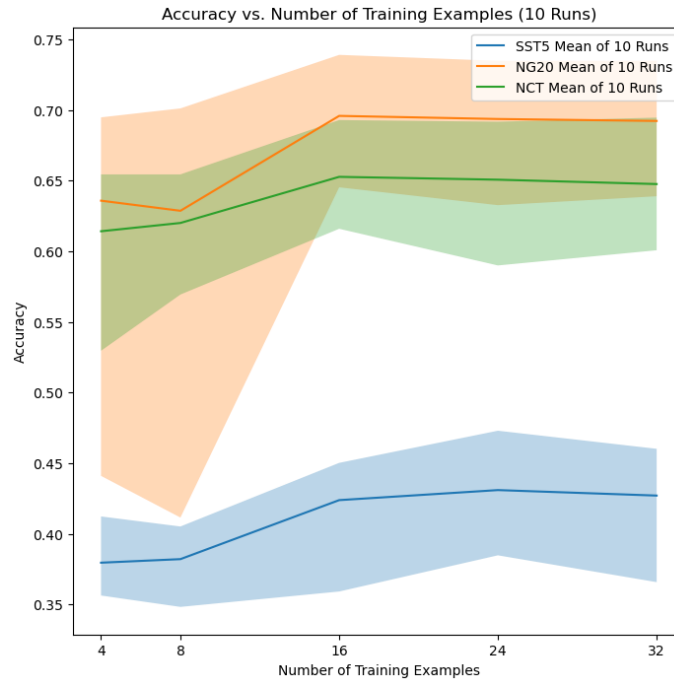


Figure 7.6: COTA evaluation – accuracies deviations over 10 fine-tuning runs. Figure from Fischer, Schneider, Geislinger, et al. (2024).

Dataset	mean	std	min	max
NG20	205.48	652.55	1.00	20082.00
SST-5	19.14	9.31	2.00	52.00
NCT	9.15	3.25	1.00	38.00

Table 7.1: COTA evaluation – number of white-space-separated words per dataset.

Results

Figure 7.6 shows the averaged accuracy from 10 runs of the SST-5, NG20, and NCT experiments. For all three experiments, we observe a steady improvement in accuracy up to 16 training examples per class. Labeling more examples does not improve the results. Additionally, we note that the quality of annotation significantly impacts our setup. In the 20 newsgroup experiment, the difference in accuracy between the best- and worst-performing runs exceeds 30%.

We acknowledge that our approach’s performance does not match state-of-the-art results on these datasets due to the few-shot nature of these experiments. To mitigate this, in the timeline analysis, users can define a similarity threshold to filter out irrelevant results. In addition, we provide an interface to assess the sentences under consideration for a given concept and time. We believe this transparency is crucial when working with such machine-learning-based applications.

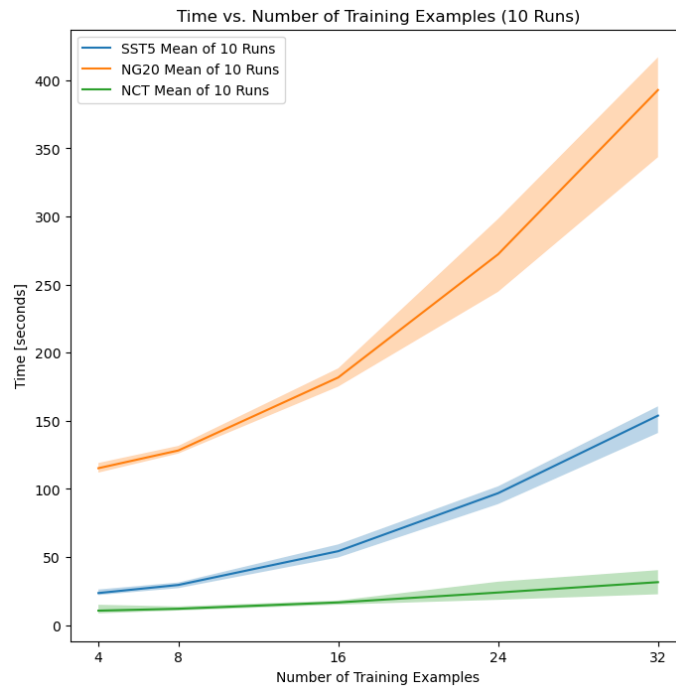


Figure 7.7: *COTA* evaluation – average runtimes over 10 fine-tuning runs. Figure from Fischer, Schneider, Geislinger, et al. (2024).

Few-Shot Training Runtime

An important requirement for implementing interactive tools is the responsiveness or latency. We measured the training times for the experiments to estimate the time users must wait before receiving the results of the *COTA* refinement in the interface.

The results are shown in Figure 7.7. Here, we see that runtime heavily depends on sentence length (see Table 7.1) and the number of training samples. Since the sentences within DATS are created using spaCy⁴, we can assume typical lengths. Hence, we expect users of our tool to wait at most two minutes when refining a *COTA* with 32 annotations per concept.

7.4.5 Implementation

Our experiments showed that a system based on this approach would meet most of our requirements (Section 7.4.4). It requires little labeled training data, training is fast, and it scales with more training samples, but only up to 16 examples per class. Hence, the *Concept Over Time Analysis* is integrated into the DATS platform (described in detail in Section 4.5) using existing background-processing (RQ workers), sentence similarity search (Weaviate), and visualization (Recharts) capabilities. Model training is implemented with SetFit⁵ (Tunstall et al. 2022).

2. <http://qwone.com/~jason/20Newsgroups/>

3. <https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2>

4. <https://spacy.io/>

5. <https://huggingface.co/docs/setfit>

The main challenges involve classifying concepts, while (1) adapting to limited user feedback, (2) achieving high performance, and (3) ensuring interactivity through responsiveness. Furthermore, concepts are often nuanced and share the same domain, resulting in high inter-concept similarity (which we also respected when conducting the experiments).

The *Concept Over Time Analysis* workflow is internally structured as a three-step pipeline, encompassing: (1) establishing the initial search space, (2) fine-tuning the sentence transformer model, and (3) computing the results for visualization. In the subsequent paragraphs, we provide a detailed explanation of each step.

The Initial Search Space: To initiate the *Concept Over Time Analysis*, users provide concepts along with proper descriptions. The descriptions are fed into a sentence transformer model to generate an initial representation for each concept. The model is configurable through configuration files and defaults to a pretrained multilingual CLIP (Radford et al. 2021; Reimers and Gurevych 2020) model.

Following this, the concept representations are employed to identify semantically similar sentences, leveraging the existing semantic similarity search feature within DATS. Each uploaded document runs through a multi-step pre-processing pipeline (Section 4.6) that includes sentence splitting and sentence embedding, with the results stored in a vector database. Consequently, the vector database is utilized to retrieve the top K similar sentences for each concept representation, along with their respective similarity scores. K is configurable in the user interface and defaults to 1000.

The set of returned sentences constitutes the search space, which remains constant throughout the subsequent steps. However, the interface allows users to reset the search space and restart the analysis process.

The initial concept descriptions play a pivotal role in the entire analysis, determining which sentences are considered. This step is executed only once at the beginning, but it is crucial to the process, as it limits sentences to the concepts' domain and speeds up the following steps.

Model Fine-Tuning: Users provide feedback by labeling sentences with concepts. This step is skipped if there are insufficient annotations for any given concept. The minimum number of labeled examples per concept can be configured in the user interface; however, our experiments suggest that 4, 8, and 16 are reasonable thresholds to start fine-tuning.

A contrastive training dataset is generated from users' annotations and used to fine-tune a pre-trained sentence transformer model (details were described in the previous Section 7.4.4). The model trains asynchronously in the background for one epoch, and the frontend is informed about status changes and displays them accordingly.

Result Visualization: In this final step, we compute a 2D representation of the search space, update concept representations, re-rank sentences, and compute the timeline analysis:

- To generate the 2D representation displayed in the annotation view, the fine-tuned model computes sentence embeddings, which are then passed

to UMAP (McInnes et al. 2020; Section 2.1.7) or t-SNE (van der Maaten and Hinton 2008) for dimensionality reduction. Initially, the embeddings stored in the vector database represent the search space sentences. Throughout iterative refinement, models with improved performance enhance the sentence representations, causing sentences to *move* toward their most similar concept and form clusters: Figure 7.8 shows the evolution of sentence embeddings produced during our experiment with the NG20 dataset. We use t-SNE (van der Maaten and Hinton 2008) to reduce the embeddings to 2D for better visualization.

- Initially, a concept is represented by its embedded description. An enhanced representation is derived by averaging the embeddings of the concept's labeled sentences.
- We calculate the cosine similarity between the concept representations and all embedded search-space sentences. The sentences shown in the sentence annotator are then ranked based on the resulting similarity scores.
- Finally, the timeline analysis is computed. For each concept, sentences below the similarity threshold are filtered out, and the remaining sentences are aggregated and counted within the specified time intervals (year, month, or day).

In summary, *COTA* introduces a machine-learning-based workflow that streamlines the identification and analysis of user-defined concepts. By integrating few-shot sentence classification into an interactive interface, the system allows researchers to create temporal visualizations of their concepts of interest.

7.5 Conclusion & Discussion

This chapter established the necessity of digital support for analysis in discourse analysis (Section 7.1). Quantitative procedures are critical for Lexicometric Discourse Analysis and for the initial material selection and structuring phase within SKAD, where quantification helps estimate the reach and scope of a discourse.

Importantly, the resulting quantitative findings must always be linked back to the qualitative source material for qualitative recourse (Section 7.2). Scaling discourse analysis to large datasets requires advanced pattern recognition techniques to capture the connections within a discourse, such as temporal trends, co-occurrence analysis, and network visualizations. However, quantitative analysis in discourse analysis projects is challenged by methodological risks, such as the quantification trap (equating frequency with power) and the conflict with corpus construction (theoretical sampling vs. representative sampling).

DATS provides basic analysis features with word frequency analysis, code frequency analysis, and contextual statistics (Section 7.3). The features support *traceability* by dynamically linking statistics to filtered documents and offer a distant reading approach to data analysis. However, they offer limited pattern recognition capabilities.

We subsequently described the *Concept Over Time Analysis* feature, a machine-learning-based tool within DATS that facilitates the identification and analysis of user-defined concepts in large datasets over time (Section 7.4). The tool empowers users to define concepts, identify relevant sentences, and visualize results in a timeline through an interactive interface. Specifically, *COTA* tightly integrates semantic search with manual annotation through an iterative human-in-the-loop interface to enable semi-automatic classification and visualization of concepts on a timeline. It leverages few-shot fine-tuning of the sentence transformer model (for sentence and concept representations) guided by user feedback, enabling the model to quickly adapt to the researcher’s specific domain and evolving concepts with a minimal number of labeled examples. This results in a powerful tool for a mixed-methods approach to analyzing large datasets. The chronological overview of custom concepts allows researchers to identify patterns, trends, and shifts in discourse over time. Beyond discourse analysis, *COTA* supports various applications, including visualizing topics or sentiments over time. Further, it can provide valuable insights into the evolution of ideas, narratives, and cultural practices, offering a dynamic understanding of discursive trends and directing the researcher’s attention to specific events at particular times

In conclusion, this chapter demonstrated how to successfully combine advanced NLP techniques (specifically contextualized embeddings, few-shot classification, and interactive visualization) into a cohesive mixed-methods approach.

Limitations regarding LLMs are discussed in the overall thesis limitations in Section 9.2.

Discussion

The contribution of this chapter — the implementation of the *Concept Over Time Analysis* feature (Section 7.4) — is to enhance the DATS platform by explicitly addressing the methodological and technical requirements of modern discourse analysis (Section 1.5).

The challenge of *scalability* is overcome by two key technical choices in *COTA*. First, to efficiently handle large, growing text corpora, *COTA* leverages a vector database (Weaviate) for initial semantic search and data retrieval. Second, the model update process relies on few-shot fine-tuning (SetFit), which uses only a small, labeled subset of the data. This drastically reduces the time required for training and the computational resources needed.

Furthermore, the system is fundamentally built to support the iterative nature of qualitative research, addressing the requirements for *adaptability*, *interactivity*, and *responsiveness*. *Adaptability* is ensured through few-shot fine-tuning, which allows the model to incrementally adapt to the researcher’s evolving, dynamic concepts with minimal effort. This process is inherently interactive, as the *COTA* workflow is explicitly designed as an iterative loop (define → refine → visualize). Finally, the experimental results confirmed that the model refinement phase—the primary latency bottleneck—runs in seconds to minutes, ensuring fast responses and a smooth, usable process.

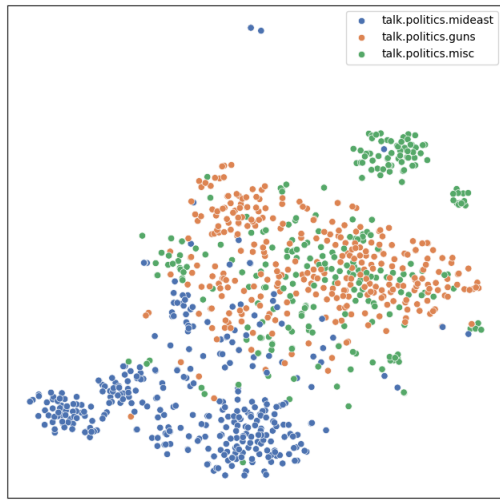
In line with the human-in-the-loop paradigm, *COTA* rigorously enforces *user control* and *transparency*. *User control* is paramount: the researcher retains full authority over analytical decisions, including concept definition, labeling, and

refinement; the AI functions purely as an augmentative tool. *Transparency* is provided through interactive visualizations, such as the scatter plot and sentence ranking in the annotation view. This visual insight into the system's clustering and ranking logic makes the decision-making transparent, aiding scientific validity.

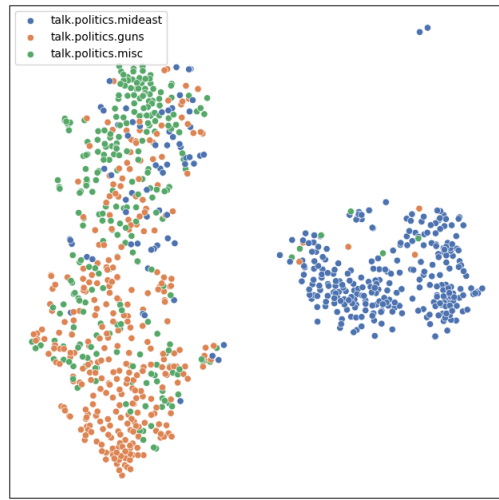
Next, *COTA* ensures *traceability*: the interactive timeline visualization allows a user to click an aggregate count and immediately reveal the specific source text passages (sentences) that contributed to that aggregation. Hence, *COTA* successfully grounds quantitative observations directly in the qualitative source material.

Finally, *COTA* meets the requirements for *flexibility* and *accessibility* relevant for digital humanities researchers. Its *flexibility* stems from using general-purpose sentence transformers and few-shot learning, as demonstrated across diverse use cases (news, sentiment, general topics), confirming the method's generic applicability across domains. Concurrently, *accessibility* is achieved because the interface abstracts complex machine learning concepts (embeddings, dimensionality reduction, fine-tuning) behind intuitive, step-by-step actions (define → refine → visualize), making this powerful mixed-methods approach usable for digital humanities researchers without a computer science background.

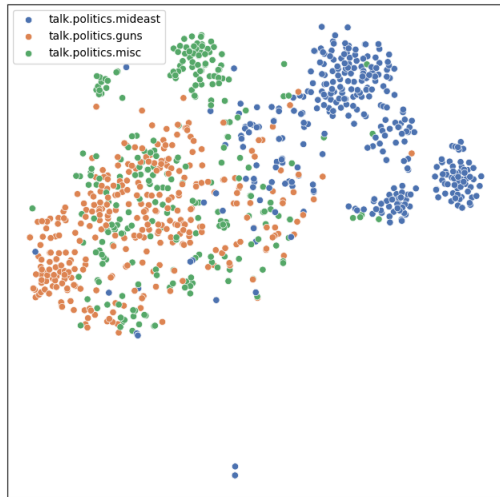
The computational methods introduced in the previous chapters—from interactive clustering to timeline analysis—offer useful ways to navigate and scale qualitative insights across large datasets. However, in discourse analysis projects, these qualitative and quantitative insights are not the final results of the research. Instead, they serve as a foundation for the core task: the hermeneutic interpretation. The researcher must synthesize these disparate artifacts (memos, codes, statistics, timelines, text passages, etc.) into a coherent argument or theory. The following chapter addresses this concluding step, presenting a visual workspace designed to support the reflective and interpretive depth required for discourse analysis.



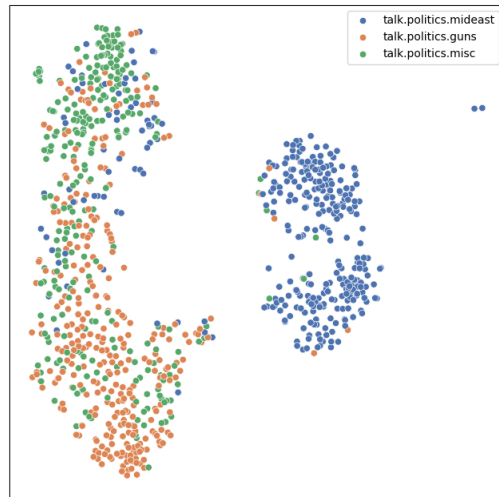
(a) Number of training samples per class: 4



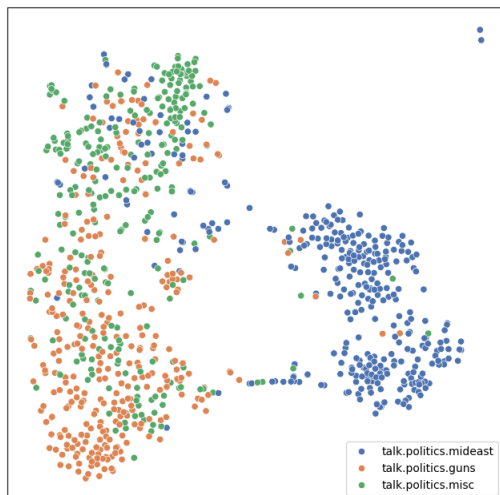
(d) Number of training samples per class: 24



(b) Number of training samples per class: 8

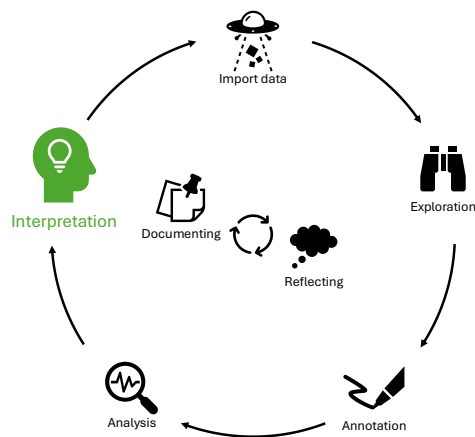


(e) Number of training samples per class: 32



(c) Number of training samples per class: 16

Figure 7.8: COTA evaluation – evolution of sentence embeddings produced by the NG20 experiment. As training samples increase, "mideast" becomes clearly separated, whereas "guns" and "misc" remain largely interspersed. Figure from Fischer, Schneider, Geislinger, et al. (2024).



8

Reflection & Interpretation

Contents

8.1	Motivation	192
8.1.1	Interpretation	192
8.1.2	Reflection	193
8.2	Challenges	194
8.3	Interpretation & Reflection in DATS	195
8.3.1	Workflow with Alice	195
8.3.2	Addressed Challenges	197
8.4	Whiteboards for Visual Qualitative Analysis	197
8.4.1	Introduction	197
8.4.2	Related Work	198
8.4.3	Implementation	199
8.4.4	Exemplary Use Cases	201
8.4.5	Workflow with Alice	205
8.5	Conclusion & Discussion	206

This chapter addresses the key phase of the iterative discourse analysis workflow: reflection and interpretation. While previous chapters focused on the computational augmentation of exploration, coding, and analysis, this chapter focuses on the human-centered process of synthesizing findings to generate object-related theories. The goal is to provide a digital environment that supports interpretation while ensuring scientific validity through traceability. We begin by establishing the motivation and challenges of interpretation (Sections 8.1 and 8.2), emphasizing the need to manage process-generated research artifacts. Section 8.3 details the baseline interpretation and reflection features in DATS, specifically the memo and logbook functionalities that allow researchers to capture interpretive thoughts at the point of discovery.

The core contribution of this chapter (Section 8.4) is the introduction of *Whiteboards* for visual qualitative analysis (Fischer, Schneider, Petersen-Frey, et al. 2024). This feature provides an interactive, graph-based interface that

enables researchers to organize research artifacts and synthesize findings while maintaining full traceability; every visual element remains linked to the underlying raw data.

By developing *Whiteboards*, this chapter addresses Research Question 5 and demonstrates how they can serve as a workspace for the complex, non-linear interpretation processes that define qualitative discourse analysis.

8.1 Motivation

The methodical core of qualitative discourse analysis lies in the reflexively guided constitution of meaning (Keller 2011b, p. 62). This encompasses two closely linked activities that shape the iterative, circular research process: interpretation for understanding and reflection to ensure scientific validity.

8.1.1 Interpretation

Interpretation is the key phase of a discourse analysis. Interpretation in Hermeneutics (Section 3.3) is not a subjective reading of data, but a systematic process of reconstructive understanding. The goal is to deduce and reconstruct the societal meaning of social actions and their products step-by-step from the data material (Kurt and Herbrink 2022, p. 611). This requires adopting a hermeneutic attitude (Section 3.3.2), a methodized reflection that moves the researcher from an everyday understanding to a scientific mode, characterized by skepticism and irritation by the data (Kurt and Herbrink 2022, pp. 613–614).

The Sociology of Knowledge Approach to Discourse Analysis (SKAD, Section 3.6) draws upon the methods of Hermeneutics and Grounded Theory for this interpretative work. In discourse analysis, interpretation is a multi-stage process that builds upon the discursive elements identified in the coding step (objects, concepts, enunciative modalities, strategies) and deliberately aims beyond the single text at the structuring aspects of discourse, such as interpretative patterns, classifications, narratives, and speaker positions (Traue et al. 2022, pp. 638–639; Keller 2011b, p. 57).

Interpretation is carried out using typical statements and coded discursive elements to address the objectives of discourse analysis. These goals (Section 3.4.2) are addressed through interpretive processes (Traue et al. 2022, p. 640):

1. **Function Analysis:** Based on the deconstructed discursive elements, the goal is to explain the functionality, the inner structure of the discourse.
2. **Power Analysis:** It is explained by which mechanisms of power the discourse is stabilized, also, by linking the form of knowledge with societal structures, such as laws, technologies, infrastructure, etc.
3. **Societal Analysis:** The reconstructed discourse is interpreted as an expression of societal power relations, and the societal effects of the discourse are demonstrated.

To aid the interpretative process and conclude the analysis, Keller (2011a, p. 116) formulates the following guiding questions:

- How did the research question change over the course of the investigation?
- What results were obtained with regard to the research questions?
- In summary, what are the reconstructed elements of a discursive field, the rules of formation, discourse strategies, etc.?
- How do sociohistorical context, discursive fields, discourses, practices, and dispositifs relate to one another?
- What explanations can be formulated for the reconstructed structuring processes of and through discourse(s)?
- How do the results relate to other perspectives and statements about the same or similar objects of investigation? Are they refuted, supplemented, or confirmed?
- What insights can be gained from the research process for the discourse-theoretical perspective, the methodological approaches of discourse research, and so on?
- What do the results contribute to social scientific knowledge and discussion about the object?

8.1.2 Reflection

Reflection is closely linked to the method of interpretation in the Sociology of Knowledge Approach to Discourse Analysis. Hermeneutic reflection, a research methodology that centers on the *understanding of understanding*, is a crucial part of interpretative data analysis (Keller 2011b, p. 61). The central goal of reflection is the assurance of scientific quality and validity of the interpretations (Section 3.3):

- **Traceability:** Traceability is created through the continuous documentation of the entire research process, from the research question and data selection to the analysis and synthesis of results. Importantly, the interpretation process must also be documented. Researchers must make transparent how they understand what they believe they understand (Keller 2011b, p. 61).
- **Control of Subjectivity:** The transition from everyday understanding (1st-degree constructions) to scientific understanding (2nd-degree constructions) is accomplished through constant reflection on one's own perception and own prejudices. Reflection is thus the path to *methodically controlled* data interpretation and to establishing the validity of the interpretation results (Kurt and Herbrik 2022, p. 613).

Reflection is integrated into SKAD's work steps through two central methods: sequence analysis (Section 3.3.3, Keller (2011b, p. 63)) and memo writing (Section 3.2.2, Keller (2011a, pp. 98–99)). The sequence analysis is a methodical self-control against one's own prejudices. Researchers develop many interpretation hypotheses for individual sections, which are then checked, rejected, or refined for their appropriateness based on the immediate continuation of the text. The method

consists of developing multiple readings and then making a justifiable and justified selection (Kurt and Herbrink 2022, pp. 617–622).

Further, SKAD draws on Grounded Theory methodology to systematize reflection. In memos, further considerations, ideas, and hypotheses regarding the discourse logic are recorded throughout the investigation. For instance, the viewpoints that led to the formulation and assignment of a particular interpretative code to a text passage are documented.

Thus, what is reflected upon is one's own position and biases, one's own perception, as well as the interpretation processes and, wherever possible, the complete research process itself.

Discussion

There is a key limitation regarding the ambiguity and subjectivity of applying NLP to discourse-analytical interpretation practice that needs to be discussed.

According to SKAD, the process of the constitution of meaning can only take place in the individual consciousness of the interpreting subject (Keller 2011b, pp. 49–51). The method encourages researchers to be creative and to develop and test different interpretations. The researcher's subjectivity is therefore not eliminated but is considered necessary and made transparent. In this context, reflection is an analytical tool that ensures transparency in the interpretative steps.

This makes the application of NLP models, which strive for unambiguity and reproducibility and ignore the intentional ambiguity of interpretive sociology, almost impossible for interpretative tasks. Machine procedures often aim for unambiguous classification (e.g., sentiment or topic), which ignores intentional ambiguity and the need to justify (rather than merely determine) interpretations.

8.2 Challenges

Despite the strong theoretical foundation, the practical application of hermeneutic interpretation poses numerous methodological challenges for researchers. Digital tools may help researchers address some of these difficulties. However, while computer-supported tools facilitate the structuring and quantification of data, hermeneutic interpretation and methodical self-control remain the indispensable achievements of researchers. The process is, by definition, subjective and requires remarkable personal effort, which can be made transparent and traceable through digital functions.

Contextual Sensitivity: A fundamental challenge lies in the discourse's contextual sensitivity. Discourse is deeply embedded in social, cultural, and historical contexts, which can make the analysis and interpretation quite challenging. Researchers need to be acutely aware of these contexts to avoid misinterpreting the meaning and significance of language use (Traue et al. 2022, p. 636). However, fully grasping the nuances of context can be challenging, especially when dealing with unfamiliar cultures or historical periods.

Reliability & Validity: Establishing the reliability and validity of interpretations can be difficult. It can be challenging to demonstrate that interpretations are grounded in the data and are not simply the product of the researcher's imagination. This is a central demand of Grounded Theory and Hermeneutics, taken up in SKAD, and underlines the importance of reflection and documentation.

Traceability: Closely linked to this are the hermeneutic requirements of traceability (Section 3.3.1). To secure the credibility of the research, the entire interpretation process must be transparent and traceable, which is often difficult in practice. This requires complete documentation, primarily through memo writing.

Presenting Interpretations: However, tracing the comprehensive documentation of research results across numerous individual memos presents a significant burden to the reader. Therefore, presenting interpretations poses a challenge. Researchers must communicate their findings clearly, concisely, and persuasively, while simultaneously leaving room for alternative interpretations.

Knowledge Management: A further difficulty arises from knowledge management and context switching in research practice. Researchers often use various tools for coding, searching, analyzing, and keeping a research journal. This leads to increased effort for knowledge management and organization, and frequently requires copying and pasting data back and forth. An integrated tool suite that links all functions can offer an advantage.

Linking Data: Finally, specific challenges arise from power analysis. SKAD includes the analysis of dispositifs (non-linguistic, material, and institutional practices, see Section 3.6). Researchers must ensure that extratextual data (on institutions, architectures, practices) are considered alongside texts in the interpretation process. Support for linking codes to external data and other modalities (e.g., images, institutional organization charts, historical data) can simplify the process.

8.3 Interpretation & Reflection in DATS

This section demonstrates how to use the basic features for reflection and interpretation in DATS. The technical implementation of these features is explained in Section 4.3. Alice will continue to guide us through her research workflow.

8.3.1 Workflow with Alice

Throughout her iterative research process (which includes importing new documents, organizing, searching, and filtering these materials, and qualitative coding), Alice can make use of the interpretation and reflection features of DATS: memos and the logbook. The memo functionality enables her to attach text notes to research objects, and the logbook provides a tabular overview of all created memos, along with a text editor.

After coding and exploratory analyses, Alice decides to proceed with interpretation. She starts by reviewing documents to develop her interpretation hypotheses.

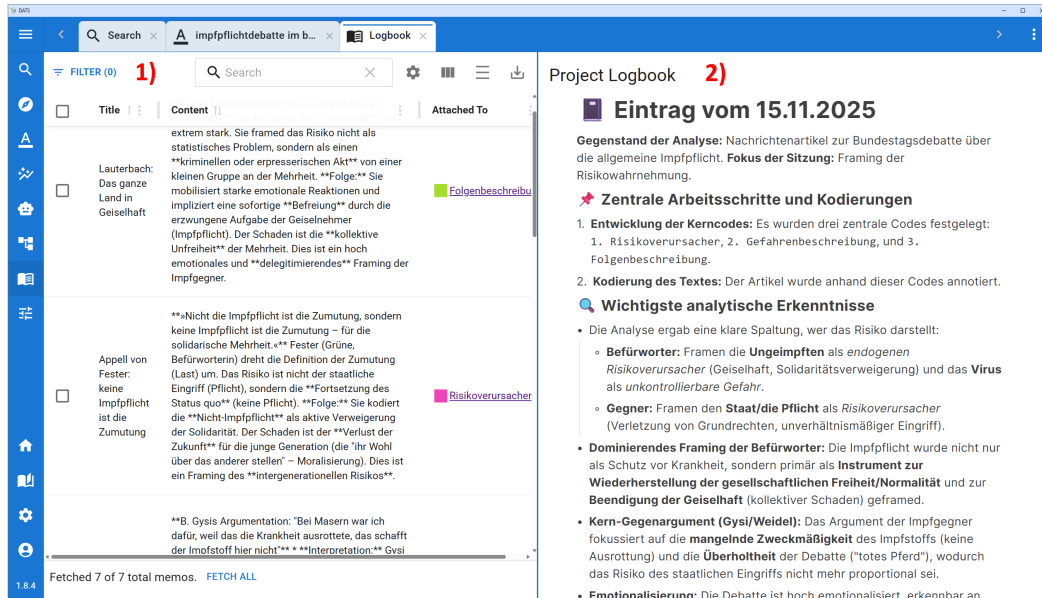


Figure 8.1: DATS logbook interface. The left table (1) shows a filtered list of memos, attached to various research materials (e.g., codes, documents, annotations). The right text editor (2) can be used to document the research process.

While reviewing her coded text passages on the vaccination debate, Alice identifies a typical statement in which the code "*Societal Burden*" co-occurs with "*Individual Freedom*". She realizes that this co-occurrence is a strong indicator of an underlying interpretive pattern: the debate is framed as a conflict between collective health responsibility and personal liberty. To document this interpretation and hypothesis, Alice uses the memo feature to attach a structured note directly to the annotated text passage, establishing the context, her hypothesis, and the reason she is pursuing this reading. She also creates a general memo on the "*Societal Burden*" code itself, reflecting on its conceptual boundaries and its relation to other codes.

At the end of her interpretive session, Alice switches to the logbook view to reflect on her current progress. This feature (see Figure 8.1) provides a filterable, tabular overview (1) of all the memos, ideas, and hypotheses she has created, allowing her to review all interpretative thoughts, ideas, and hypotheses collected throughout the day. DATS automatically timestamps and links all memos to their creation context, ensuring traceability. Alice uses the rich-text editor within the logbook view (2) to write a detailed, high-level summary of her daily findings, synthesizing the individual memos into a cohesive reflection on her overall progress, addressing potential biases, and validating her interpretations against the data.

At some point in the future, after completing her interpretation, Alice may filter the logbook for such summary memos she created. Upon completing her research, this function allows Alice to compile her main findings and arguments documented in her high-level summary memos, thereby facilitating the dissemination of the results.

8.3.2 Addressed Challenges

The basic interpretation features in DATS (memos and logbook) address some methodological challenges identified in Section 8.2, primarily related to documentation and organization. The memo-writing feature directly helps establish the research's reliability and validity. By linking memos exactly to the corresponding research material (documents, codes, annotations), researchers can ensure that the research process, and specifically the rationale for interpretive steps, is made transparent and traceable. Furthermore, by integrating documentation and reflection functionality into a single tool suite alongside coding and analysis, DATS eases knowledge management and reduces organizational friction for researchers.

However, DATS' support for the complex, synthetic stages of SKAD interpretation is limited. The memo function is limited to documentation at the level of individual research objects, which is sufficient for interpreting statements within documents but inadequate for analyzing patterns and connections in the data. To conduct a valuable, fine-grained analysis, researchers need to examine multiple, related materials simultaneously to derive interpretations. The logbook provides only a basic and static tabular overview. It lacks the functionality to provide a holistic overview of related materials, to connect or link disparate research objects spatially, or to impose effective visual order on the accumulated memos and codes. This hinders the development of complex interpretative patterns. Consequently, DATS' basic functionality does not fully meet the need for a flexible, multi-material overview for interpretation, nor does it provide a tailored environment for presenting interpretations. Addressing these specific limitations is the primary motivation for the feature presented in the next section.

8.4 Whiteboards for Visual Qualitative Analysis

8.4.1 Introduction

This section introduces *Whiteboards* for DATS, designed to facilitate and enrich the methodological procedures of Grounded Theory and Hermeneutics in qualitative discourse analysis with maps as known from situational analysis (Clarke et al. 2017). The section is based on Fischer, Schneider, Petersen-Frey, et al. (2024), who detailed the implementation of these digital canvases for the organization and generation of knowledge. By providing a flexible, spatial environment, *Whiteboards* address a critical need in qualitative research: the ability to move beyond isolated tabular views to a holistic, relational perspective of the data.

The proposed *Whiteboards* support a wide range of use cases (Section 8.4.4), including, but not limited to, discourse actor networks and code structure maps. The implementation is generic, enabling new ways of interacting with material and paving the way for novel discourse analysis practices. Specifically, the tool offers features such as creating text, post-its, various shapes, and arrows. More importantly, it allows linking all kinds of user- and system-generated data, such as annotations, codes, documents, memos, and tags, to the boards. This intertwining of typical mind-mapping features with qualitative data analysis objects creates a unique, streamlined user experience for viewing, organizing, and working with the materials.

Most relevant to interpretation processes, *Whiteboards* provide a flexible, spatial canvas that enables researchers to simultaneously review multiple, connected research objects (memos, codes, documents) that are usually isolated in tabular views. This visual and interactive environment supports the hermeneutic demands of SKAD's interpretation stages, enabling the discovery and linking of complex interpretative patterns.

At their core, *Whiteboards* are graphs of nodes and edges where nodes represent any object (e.g., shapes, post-its, documents, or annotations) while edges encode their (custom) relationships. Thus, the proposed feature can also be seen as an interactive tool for manipulating annotated, multimodal resources in a novel, graph-based way.

The development of *Whiteboards* addresses the following research question of this thesis:

Research Question 5

How can research artifacts be meaningfully collected, linked to one another, and traced back to the original sources to create a working environment for hermeneutic interpretation?

Opposed to all other previous chapters in this dissertation, this contribution does not explicitly address any natural language processing task. As discussed in Section 8.1.2, I deem NLP models are not suitable for the tasks of interpretation and reflection. Hence, the focus of this work lies mainly in developing an interactive interface that addresses the challenges of interpretation and reflection and can be used to work with results from NLP-based processes, rather than in evaluating, developing, and integrating additional NLP methods into DATS.

The contributions are:

1. We develop and describe *Whiteboards* for DATS that provide a customizable view of and interaction with user- and automatically-generated resources.
2. We demonstrate the practical utility and integration of *Whiteboards* for discourse analysis by showing real-world use cases of refining hierarchical code structures, creating actor networks, and reflecting using sampling maps, highlighting the effectiveness and usefulness of the proposed feature.

8.4.2 Related Work

Related work includes software for qualitative data visualization or for creating mind maps.

New/s/leak (Wiedemann et al. 2018) and Jigsaw (Görg et al. 2014) are visual analytics systems for investigative analysts that display documents and automatically extracted entities, such as people, places, and organizations. While New/s/leak provides a generated graph view of the entity network, Jigsaw offers "*Ttblets*" for manually creating such networks. *Whiteboards* cover all of *tablets*' functionality and extend it to other materials, such as documents, tags, and entities.

MAXQDA is a commercial but frequently used tool for mixed-methods research in DH (Section 4.2). It offers the *Creative Coding*¹ feature to manage code systems on a 2D canvas, as well as *MAXMaps* for creating visual models of various data types. While offering a creative interaction, these features are often limited to specific, predefined templates or hierarchical structures. In contrast, our proposed feature enables comprehensive, custom-generated visualization and working with any material, including codes, annotations, tags, or documents.

Miro² is a visual workspace that started as a tool for collaborative mind mapping but has since integrated tools and features to support various use cases across companies, including project planning, data visualization, diagramming, agile processes, and much more. Miro was analyzed and evaluated as part of a manual discourse analysis in a co-creation approach (Eiser et al. 2023) to software development. It was found to be helpful and motivated the full integration of *Whiteboards*, which overcomes many disadvantages of working with third-party tools, such as cumbersome data copying and pasting, losing connection to the source material, and manually applying changes across all used tools.

The developed *Whiteboards* feature differs significantly from conventional mind-mapping tools. While providing standard features such as post-its, texts, various shapes, and arrows, it further facilitates direct interaction with and manipulation of research objects, as elaborated in the following section.

8.4.3 Implementation

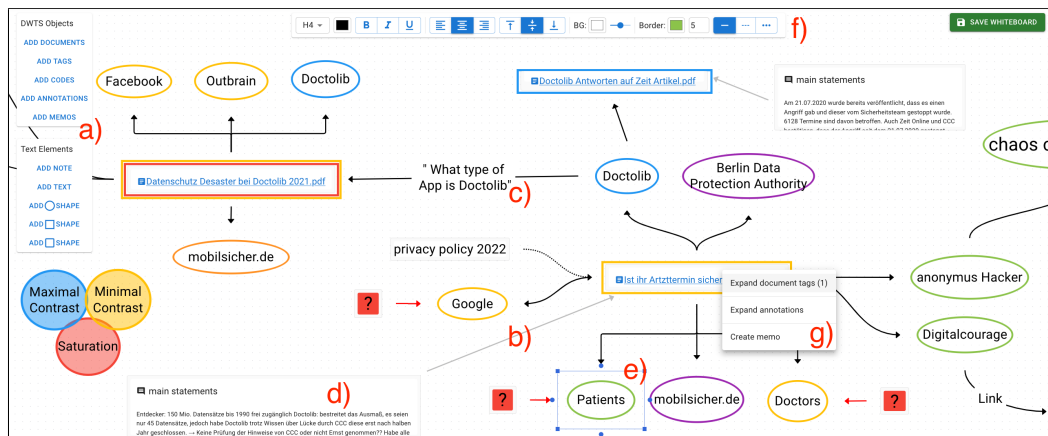


Figure 8.2: *Whiteboards* use case – sampling map. The sampling map was created with the *Whiteboards* extension. Features are described in Section 8.4.3. Figure from Fischer, Schneider, Petersen-Frey, et al. (2024).

Whiteboards allow interacting with research materials in a novel way on a 2D canvas. Such materials and objects are the outcome of typical workflows within DATS and include, for example, uploaded documents, tags, codes, annotations, and user-created memos, as well as automatically annotated named entities from the pre-processing pipeline, as already provided by DATS (Section 4.3).

1. <https://www.maxqda.com/help-mx22/06-creative-coding/idea-behind-creative-coding>

2. <https://miro.com/>

Users can create and work with multiple *Whiteboards*. In technical terms, *Whiteboards* are graphs of edges and nodes. Nodes represent automatically generated and user-created objects. There exist two conceptually different nodes:

1. Database nodes include documents, codes, tags, memos, annotated text passages, etc. They are directly linked to DATS' database model.
2. Whiteboard nodes include texts, post-its, and various shapes, and only exist in a particular whiteboard.

Similarly, there exist two different types of edges. Database edges are directly inferred from the database. For example, a database edge is automatically rendered between a document and a tag node if the document was assigned that tag. Further, database edges exist between annotation and code nodes if text passages were annotated with that code. While database edges cannot be customized, whiteboard edges are customizable arrows that exist only within a particular whiteboard.

As database nodes and edges are linked to the respective objects and their relations, modifications are applied in both directions: Changes to database nodes or edges in *Whiteboards* are reflected in all other parts of DATS and vice versa. For example, renaming codes in *Whiteboards* applies these changes globally. Connecting tags and documents with database edges, assigns these tags to the documents.

Its primary use cases are map creation and knowledge organization. Hence, *Whiteboards* are designed to be fully interactive and allow various actions (see Figure 8.2): Navigation is done by panning and zooming. Database and whiteboard nodes can be added using the sidebar (a). While database edges (b) are added automatically, whiteboard edges (c) can only be created by users. Additional database edges are created by dragging and dropping from the grey border of database nodes to valid other database nodes (d). In contrast, whiteboard edges can be created by connecting two blue dots (e). Double-clicking database nodes allows modifying the corresponding database entries, whereas whiteboard nodes and edges can be customized using the toolbar (f). The toolbar offers various options, such as changing font size, color, and formatting; positioning (for whiteboard nodes); arrowhead color, thickness, and style (for whiteboard edges); and more. The context menu offers additional actions depending on the node (g). Most importantly, the menu allows expanding related database nodes, adding them to the board with their respective database edges. For example, a code node allows the creation or expansion of the parent code, all the child codes, the attached memo, or all annotated passages. A document node enables expanding all document tags, all annotations, the attached memo, or linked documents. This expansion feature allows an exploratory way to traverse the database, finding connections in the data while interactively unfolding a network of connected nodes.

The *Whiteboards* extension is unique in intertwining typical mind-mapping functionalities with database objects (research materials) derived from qualitative data analysis. This way, it allows for a unique user experience. We want to highlight that every database node allows users to jump to the respective content in the application: For example, clicking a node representing a document will open the corresponding document in the document viewer. Similarly, clicking a node

representing an annotation will open the document and jump to that exact passage. This interconnectivity is where this feature shines and was a major motivation to fully integrate *Whiteboards* into DATS rather than relying on third-party tools.

Working with all relevant material on a board while adding custom post-its and arrows enables creative interactions. It offers a different view of the material besides the existing presentations, such as lists, tables, or plain documents. Organizing, refining, and managing data visually is a meaningful way to better understand phenomena, identify relations, track research processes, and gain insight into discourses. The use of other visualization tools, such as Miro, would be feasible in terms of technical functions, but would require a change of working environment and leaving the analysis tool. In this context, the alternating translation work of the programs, which is a double effort of digital handcraft (Franken 2020) for researchers, is no longer necessary.

The ability to spatially arrange and connect these various research materials creates an environment conducive to interpretation. The transition from linear, text-based documentation (e.g., memos in the logbook) to a flexible, graph-based workspace enables researchers to engage in the inductive reasoning required by Grounded Theory, moving beyond individual object reflection to the systematic development of complex interpretative patterns and theoretical constructs.

Integration within DATS

Whiteboards are integrated into the DATS platform's backend and frontend (described in detail in Section 4.5) with minimal architectural changes. The backend provides all the necessary functionalities to retrieve research materials and their relations. The frontend is implemented with the libraries *React Flow*³ and *MUI*⁴ to build an interactive graph-based user interface.

8.4.4 Exemplary Use Cases

This section discusses real-world use cases for the *Whiteboards* feature. All described use cases are part of a project on a discourse analysis of the debate on *e-health* in Germany⁵. We compare our proposed tool with Miro, which was used in this project previously.

Code System Management

DATS supports hierarchical code structures. It provides means to refine, merge, or delete codes, which are integral to Grounded Theory approaches. Coding in DATS is done in the annotation interface, which involves highlighting text passages and selecting the code from a list, as well as managing the code taxonomy with the code explorer, which visualizes the taxonomy as a tree view (Section 6.5).

Whiteboards allow a different, graph-based view of the code taxonomy as shown in Figure 8.3. Consequently, the interaction also differs: The visualization

3. <https://reactflow.dev/>

4. <https://mui.com/>

5. The e-health debate in Germany centers on a nationwide digital network for the healthcare system. It is characterized by the tension between digital innovation, bureaucratic regulation, and high legal standards for patient data privacy

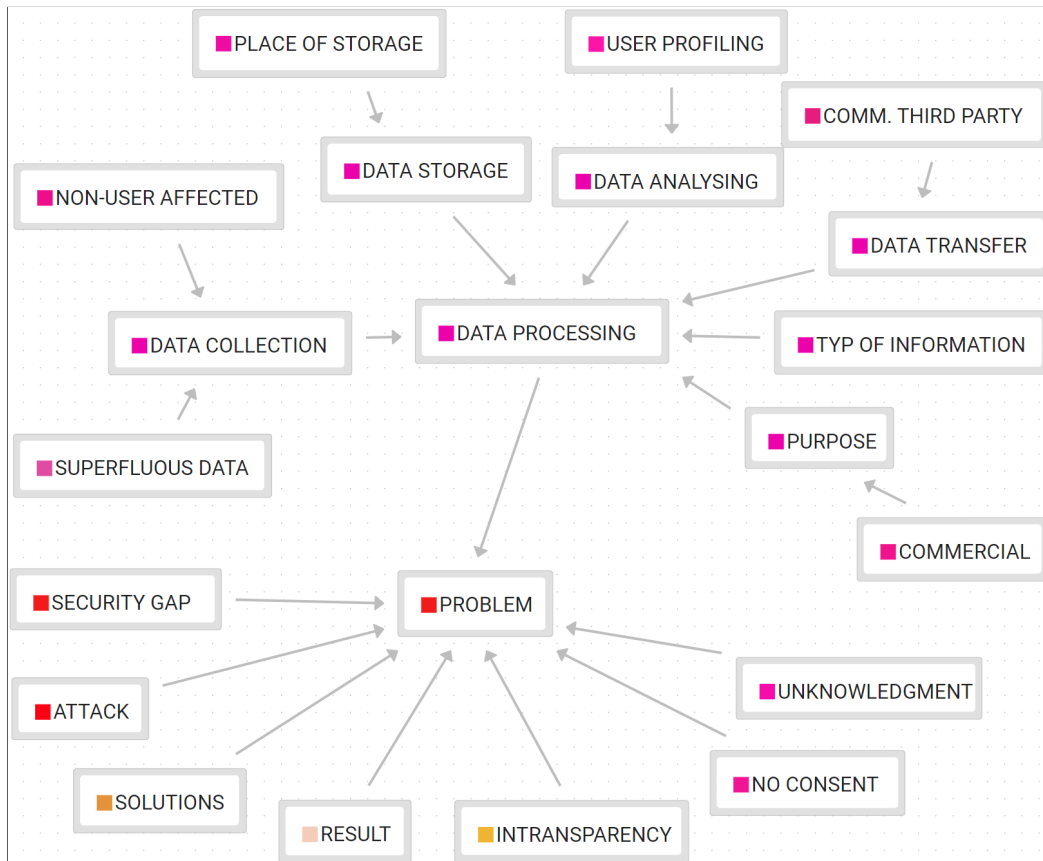


Figure 8.3: *Whiteboards* use case – taxonomy management. The *Whiteboards* extension was used to refine a complex category system on data protection. Figure from Fischer, Schneider, Petersen-Frey, et al. (2024).

of the codes can be rearranged according to the user's preference by dragging & dropping nodes. Creating and changing a code hierarchy by adding or modifying parent-child relations is done by drawing a line to add an edge. By selecting multiple nodes and then performing a context menu action, the corresponding codes can be merged, or a new top-level code can be introduced. This interaction makes *Whiteboards* a powerful tool for organizing and refining code structure. It allows for a visual, creative, and interpretable approach to coding.

After an extensive phase of open coding, the resulting category system of data protection codes was reviewed on a *Whiteboard* specifically for this task. The category system was messy, a typical outcome of open and in-vivo coding. In particular, the features for merging codes and introducing new top-level codes were frequently used to refine the taxonomy. Users preferred this graph-based interaction with their code taxonomy over the list- or tree-view-based approach offered by the code explorer. Users noted that they sometimes expand the annotated passages or corresponding memo to remember what the code describes, and possibly rename them with more descriptive names or identify code that can be merged. However, they also noted that the view can become confusing if too many passages are annotated with a code. This use case demonstrated the benefit of linked material, where codes are directly linked to annotated text passages, and highlighted a key difference from Miro.

Actor Network

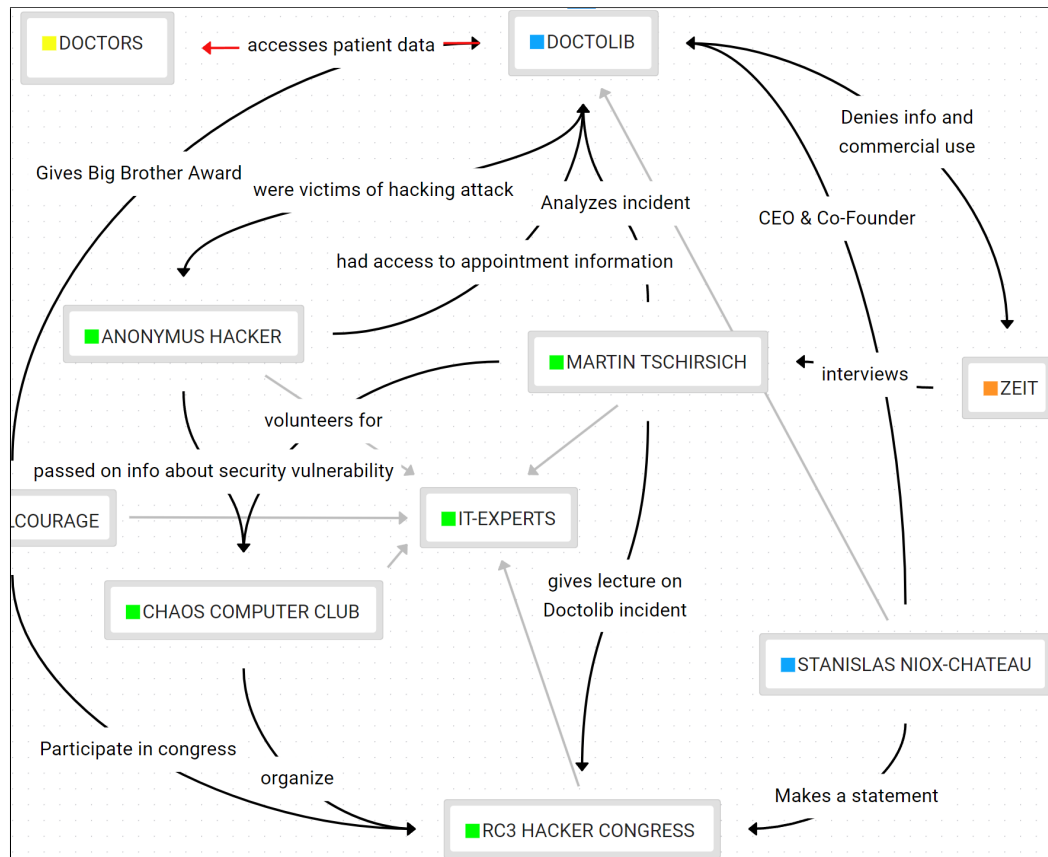


Figure 8.4: *Whiteboards* use case – actor network. An excerpt of a manually created actor-network built with the *Whiteboards* extension. Figure from Fischer, Schneider, Petersen-Frey, et al. (2024).

DATS ships with object detection and named entity recognition models that are applied during text and image pre-processing (Section 4.6.3). Automatically detected events, persons, and organizations are especially helpful in identifying actors involved in the discourse.

During the open coding phase and the reconstruction of the discursive formation, recurrent relationships among actors emerged, revealing that various actors assumed distinct roles and responsibilities. An actor-network (see Figure 8.4) was created with the *Whiteboards* extension to capture these relationships. Initially, the map was centered around a company and a health app related to various data security problems in *e-health*. Expanding on that, automatically detected entities, relevant actors, and relationships identified during close reading of the material were added to the board.

Actor networks are integral to the presentation, discussion, and reflection of results, the analysis process, and knowledge management. In this use case, *Whiteboards* helped visualize the relevant content of online articles. Furthermore, it helped structure the discourse, an essential endeavor in discourse analysis. Using *Whiteboards* instead of Miro allows quick insertion of actors from processed research material (via context menu actions); they do not have to be inserted laboriously one by one into a board.

Theoretical Sampling Map

Theoretical sampling is a commonly used method in Grounded Theory methodology. It involves theory-driven and selective data collection to refine and develop theories or concepts (Section 3.2.2).

Following a selective coding phase, a sampling map (see Figure 8.2) was created. The main objectives are to provide a comprehensive overview of the current progress and identify missing data in the corpus to achieve theoretical saturation.

The sampling map visualizes the documents selected for detailed analysis under theoretical sampling, along with their relevant actors. Starting from an initial document that appears relevant, the path through the analyzed documents can be depicted by first analyzing very similar documents with minimal contrast to achieve a more precise reconstruction. To capture the full spectrum of the discourse, documents as different as possible are used (maximum contrast). This map helps reflect on theoretical saturation and identify possible voids, such as *silent* actors mentioned in the corpus but not occupying a distinct speaker position.

By visualizing the material and associated actors with *Whiteboards*, the users could effectively identify such gaps in their corpus. New documents were sampled explicitly in the following steps to achieve better saturation. Ensuring proximity to the data and incorporating all analytical elements of the research process were particularly important in this scenario, making *Whiteboards* the preferred choice over Miro.

Image Map



Figure 8.5: *Whiteboards* use case – image map. An excerpt of an image map to analyze visually conveyed knowledge, manually built with the *Whiteboards* extension. Figure from Fischer, Schneider, Petersen-Frey, et al. (2024).

The Discourse Analysis Tool Suite supports processing all common modalities: text, images, audio, and video (Section 4.6.3). The project on the debate over *e-health* in Germany primarily used news articles as data sources, which are often accompanied by images.

Consequently, images were also coded in the phase of open coding in addition to text documents: They were analyzed with respect to attributes such as the roles

of the depicted persons, the image's focus, the image type (e.g., photography vs. illustration), and the image's use (e.g., advertising), among others.

Subsequently, the image map (see Figure 8.5) was created with the *Whiteboards* tool to provide an overview of relevant discourse images and to analyze the knowledge conveyed visually. In this map, the images were arranged by similarity, topic, and striking overlaps.

The image map enabled users to analyze which symbols are used in medical and technological fields and to find relations to other societal discourses. Further, the image map facilitated the exploration of research questions related to (a) analyzing the social codes represented in the images, (b) understanding how the digitization of health data is visually portrayed, (c) investigating the absence of specific individuals, and (d) examining the portrayal of particular individuals in different roles.

8.4.5 Workflow with Alice

This section concludes Alice's journey through DATS and her first pass through the iterative discourse analysis process. Previously (in Section 6.6.2), Alice used the *LLM Assistant* for span annotation to identify politicians (e.g., "*Karl Lauterbach*") and key institutions (e.g., "*Robert Koch Institute*") across her news article corpus. These annotations form part of the foundation for her analysis of actors and power.

Now, moving to the interpretation stage, Alice creates a new *Whiteboard* to transform her research materials into a narrative and visual argument. It serves as her dynamic workspace for synthesizing her final interpretation. To build an actor network, Alice drags and drops the key actor codes (e.g., *Politician*, *Robert Koch Institute*) and relevant documents that contain the most impactful statements onto the canvas. She then uses the node's context menu to expand related artifacts (e.g., pulling in the memo she wrote on "*Societal Burden*" or the annotated spans of the *Robert Koch Institute's* statements).

Using whiteboard edges to develop relationships, she draws connections between actors, labeling the links with concepts such as "*criticizes*", "*proposes*", or "*is cited by*" to visualize the network of influence and speaker positions. She uses post-it notes and text elements on the board to write her high-level interpretive hypotheses about the political power dynamics, creating a complete visual argument.

The true power of the integrated *Whiteboard* is its direct link to the DATS database and all research materials. Alice can present her finished actor network along with her current interpretation, knowing that (a) clicking any code or entity node (e.g., the "*Karl Lauterbach*" node) opens the underlying span annotation in the annotation view and (b) clicking any document node instantly navigates to the source news article. This connection ensures that her interpretation is entirely traceable and grounded in the qualitative data, providing a powerful medium for presenting her current research findings to her collaborators.

In summary, the *Whiteboards* extension provides a flexible, graph-based workspace that bridges the gap between individual research artifacts and cohesive interpretive arguments. By intertwining database-linked objects with custom visual elements, it enables researchers to spatially organize findings and maintain a direct connection to the source material throughout the interpretation process.

8.5 Conclusion & Discussion

This chapter established that the interpretation and reflection phases are the indispensable, subjective, and hermeneutic core of discourse analysis (Section 8.1). Interpretation is the multi-stage analytical process for reconstructing discursive formations and constituting meaning. At the same time, reflection is the methodical self-control required to ensure the scientific validity and transparency of the interpretations.

Practical difficulties challenge this process (Section 8.2). While SKAD demands complete documentation to ensure reliability and transparency, significant difficulties arise from the organizational load of knowledge management and context switching when researchers rely on disparate tools. Furthermore, achieving the necessary synthesis during the fine-grained analysis requires tools that offer flexible linking and overview capabilities.

DATS' core functionalities provide basic support for these stages through the memo feature, which enables researchers to attach contextual notes directly to individual research objects, and the logbook, which provides a filterable, tabular overview for high-level reflection (Section 8.3). However, the linear, static nature of DATS' logbook feature proved inadequate for supporting the synthetic stages of interpretation and for presenting interpretations.

To fully support the complex, non-linear, and creative reasoning required for interpretation, we developed the *Whiteboards* extension in close collaboration with cultural anthropologists for visual qualitative analysis (Section 8.4). As an interactive 2D canvas fully integrated into the DATS platform, it interleaves standard mind-mapping features with creative operations for manipulating research materials. The *Whiteboards*' working environment proved highly flexible in its application for digital discourse analysis, as demonstrated by our empirical use cases: reflecting with sampling maps, modeling actor networks, analyzing images, and organizing complex code structures. By enabling researchers to connect disparate research materials (codes, memos, documents, annotations, tags, etc.) on a spatial canvas, it provides a unique environment that supports SKAD's iterative interpretation processes and the synthesis and presentation of interpretations. Importantly, all nodes (representing research materials) and edges (representing relationships) are seamlessly linked to the database and source materials.

In conclusion, this chapter demonstrated that the challenges of interpretation and reflection are fundamentally human-centric and primarily concerned with the organization, synthesis, and presentation of knowledge, rather than being problems solvable by NLP. The main achievement here lies in building a fully integrated tool suite rather than multiple, separate tools. By integrating *Whiteboards* into DATS, they leverage the platform's unified database model, enabling comprehensive, *traceable* linking of every research artifact into a single, interactive workspace. This should ease knowledge management and reduce organizational friction.

Limitations regarding accessibility of DATS (which apply to complex tools like *Whiteboards*) are discussed in the overall thesis limitations in Section 9.2.

Discussion

The contributions of this chapter — the memo and logbook functionalities and the *Whiteboards* tool (Section 8.4) — aim to address the SKAD methodology by supporting interpretation, documentation, and reflection. This section discusses how these contributions relate to the general requirements of a discourse analysis tool suite (Section 1.5).

First, the memo and logbook features established a baseline for documentation, ensuring basic *traceability* by linking interpretive thoughts to the source material. Furthermore, *Whiteboards* reinforce *traceability* by guaranteeing that interpretations performed on the spatial canvas remain anchored to their source. Since every linked artifact points directly back to the source data, the analysis remains grounded.

Second, *Whiteboards* directly address the requirement for *interactivity*. The generic node/edge graph architecture facilitates the interactive rearrangement and free-form linking of artifacts. This interactive environment is key to supporting the flexible, iterative, and data-driven nature of Grounded Theory-based research methods.

Furthermore, *Whiteboards* align with the requirement of *flexibility*. The general-purpose 2D canvas is inherently flexible, designed to integrate and visualize diverse research artifacts, supporting a wide range of qualitative methods (e.g., creating actor-networks or code-hierarchy visualizations) and material modalities (e.g., text, code, image nodes).

Accessibility is also addressed. The *Whiteboard* feature was designed with *accessibility* and usability in mind, drawing on familiar interaction patterns from standard mind-mapping software. While it remains a complex tool, those accustomed to working with graph-based interfaces will quickly adapt.

Finally, some requirements were not addressed by this chapter's contributions. Requirements like *user control* (concerning algorithmic transparency and decision authority) and *adaptability* (relating to the incremental evolution of computational models) are primarily relevant in the context of human-AI collaboration. Since the memos, logbook, and *Whiteboards* are not AI-supported, these requirements do not apply here. Similarly, the requirement for *scalability* does not apply, as this chapter focuses on the qualitative depth of interpretation rather than on large-scale data processing.

The synthesis of research findings on the *Whiteboard* marks the culmination of the interpretive process within the DATS platform. Having navigated the entire discourse analysis workflow—from the initial import and exploration to the semi-automated scaling of annotations and the visual interpretation—the next chapter provides a concluding summary of these contributions, reflects on the limitations encountered during their development and evaluation, and offers an outlook on the future of DATS and interdisciplinary projects in the age of large language models.

9

Conclusion

Contents

9.1	Summary	209
9.2	Limitations	213
9.3	Future Work	215

9.1 Summary

This dissertation presented *Interactive Natural Language Processing for Discourse Analysis*, culminating in the design, development, and evaluation of several novel features within the Discourse Analysis Tool Suite. The main objective of this work was to address the methodological challenges faced by digital humanities—specifically those employing the Sociology of Knowledge Approach to Discourse—when confronted with the scalability challenges of modern digital corpora. Hence, this dissertation answers the leading question: How to effectively apply NLP methods to advance discourse analysis methodologies to handle large datasets?

The Discourse Analysis Tool Suite itself is the central technological artifact of this dissertation, serving as both a collaborative contribution and the foundational platform for all subsequent contributions. DATS was established as an open-source, collaborative web application that provides fundamental functionalities (multimodal data import, search, annotation, analysis, and interpretation) across the entire iterative workflow of discourse analysis.

Building on this robust foundation, this objective was continuously addressed by improving the tool suite through the development of scalable, adaptive AI-augmented features. This involved integrating advanced NLP techniques—including contextualized embeddings, few-shot fine-tuning, and large language models—into the iterative research workflow of discourse analysis. The contributions of this

dissertation are summarized below, categorized by the phases of the discourse analysis workflow they augment.

Exploration, Search, & Organization: Chapter 5 focused on tackling the central challenge of identifying and structuring relevant material within large corpora through methods for visualization and automated categorization.

The first research question (RQ 1: How can vector-based document representations be leveraged to provide researchers with a flexible, interactive visualization of large-scale text corpora that enables initial explorative structuring and classification?) was addressed in Fischer and Biemann (2026) by implementing *Perspectives*, an interactive clustering pipeline. The technical approach optionally rewrites documents with a large language model (LLM) and encodes them into high-dimensional contextual embeddings using instruction-tuned sentence transformers. These representations are then processed via dimensionality reduction (UMAP, McInnes et al. (2020)) and density-based clustering (HDBSCAN, Campello et al. (2013)), with resulting clusters enriched by LLM-generated descriptions. This work's effectiveness was validated by evaluating it on multiple datasets (Amazon Product Reviews, Spotify Songtexts, 20 Newsgroups, GVFC, Blurbs, Israel-Palestine, and News Bias), which proved that using document rewriting and instruction-tuned embeddings could effectively steer the clustering toward custom aspects, notably improving KNN accuracy by over 24 points in some tasks. This entire pipeline was successfully integrated into DATS to create an interactive dashboard and 2D map for exploratory corpus structuring and iterative cluster refinement.

The second research question (RQ 2: How can large language models be utilized to support researchers with the organization of large datasets?) was addressed in Fischer and Biemann (2024) by benchmarking LLMs for organizational tasks. The approach centered on zero-shot inference with large language models, leveraging clear prompt templates for text classification and information extraction tasks. The benchmark tested small, open-weight LLMs (Llama, Gemma, Mistral) on various datasets, including BBC, Tagesschau, and IMDb Genres for classification, and SQuAD, GermanQuAD, and MUC-4 for information extraction. The results confirmed the viability of the zero-shot approach, with Gemma 2 achieving the best performance. Consequently, these findings led to the development and integration of the *LLM Assistant* into DATS, enabling researchers to use LLMs to automate organizational tasks, thereby reducing the manual effort in corpus preparation.

Annotation: Chapter 6 was dedicated to improving the efficiency and quality of the qualitative coding process under conditions of massive, large-scale text corpora, but labeled data scarcity.

The third research question (RQ 3: How can language models be used to assist researchers during the qualitative coding process and enable the scaling of these insights to large datasets?) was addressed through two contributions. First, Fischer and Biemann (2024) addressed this by benchmarking LLMs on sequence labeling tasks. The approach used zero-shot inference, leveraging straightforward prompt templates for span classification. Evaluation on the Few-NERD, GermanLER, and Quotation Attribution datasets confirmed the potential of zero-shot LLMs for annotation assistance. The findings motivated the development

of the span-annotation functionality of the *LLM Assistant*, which aids open coding in Grounded Theory by automatically suggesting relevant text spans.

Second, a dynamic annotation approach for sequential sentence classification was proposed and evaluated by Fischer and Biemann (2025). The approach switches three phases: Phase 1 uses zero-shot prompting with LLMs for initial suggestions; Phase 2 transitions to few-shot prompting (in-context learning) once minimal examples are available; and Phase 3 fine-tunes a sequence tagger (a sentence transformer model with a BiLSTM + CRF head). Evaluation on a benchmark of five sequential sentence classification datasets (CSAbstract, Pubmed200k, Coarse Discourse Corpus, EmotionLines, and DailyDialog) demonstrated that model performance could be dynamically scaled based on data availability, with Mistral showing consistent improvement in the few-shot setting. The fine-tuned sequence tagger achieved the highest accuracy across all tasks, significantly outperforming both the zero-shot and few-shot LLM baselines and underscoring the need for dedicated model training to achieve state-of-the-art performance. Based on these findings, this approach was integrated into DATS by evolving the *LLM Assistant* into a three-stage *Annotation Assistant* with dedicated support for sequential sentence annotation, offering the best possible suggestions.

Analysis: Chapter 7 was dedicated to bridging individual text passages and overarching discursive trends by introducing a mixed-methods approach to analyzing concept evolution.

The fourth research question (RQ 4: How can semantic search and manual annotation be integrated into an interactive process to facilitate the semi-automated classification and analysis of concepts over time?) was addressed in Fischer, Schneider, Geislinger, et al. (2024) by implementing the *Concept Over Time Analysis* feature, an iterative human-in-the-loop workflow of search, annotation, and visualization. The approach couples semantic search with a manual annotation interface. Users define concepts with descriptions, retrieve similar sentences, label fitting sentences, and then iteratively refine the concept's representation via few-shot fine-tuning of a sentence transformer model. The final step aggregates the refined concept classifications chronologically. Evaluation confirmed that the few-shot fine-tuning approach improved concept representations on the 20 Newsgroups, News Categories, and Stanford Sentiment Treebank datasets, with accuracy steadily increasing up to 16 labeled examples per concept. This iterative workflow, including concept definition, refinement, and chronological aggregation, was integrated into DATS as the *COTA* feature. *COTA* visualizes the aggregated concept classifications as a timeline graph, supporting the identification of temporal trends and shifts.

Interpretation & Reflection: Chapter 8 focused on supporting the interpretation processes of discourse analysis methodology and emphasized traceability.

The fifth research question (RQ 5: How can research artifacts be meaningfully collected, linked to one another, and traced back to the original sources to create a working environment for hermeneutic interpretation?) was addressed in Fischer, Schneider, Petersen-Frey, et al. (2024), who proposed the *Whiteboards* tool, an interactive 2D canvas for visually organizing, connecting, and reflecting

on research artifacts. *Whiteboards* allow users to link disparate research artifacts (codes, memos, documents, annotations) on the canvas to discover and develop interpretative patterns and hypotheses. The utility of this tool was demonstrated through multiple qualitative use cases, including visual refinement of hierarchical code systems and creation of actor network diagrams. The *Whiteboards* tool established a spatial workspace for interpretation processes, ensuring all findings are grounded in the data by providing direct links to the source research materials.

Adaptability: The development of DATS was guided by the iterative and adaptive nature of SKAD and Grounded Theory, which led to the overarching sixth research question (RQ 6: How can AI-augmented methods—including visualization, annotation, and semantic search—be effectively designed within a human-in-the-loop framework to incrementally and rapidly adapt to evolving concepts and domain-specific inquiries in qualitative discourse analysis projects despite the scarcity of labeled data?). It was addressed across three contributions: *Perspectives* (Chapter 5), *Annotation Assistant* (Chapter 6), and the *Concept Over Time Analysis* (Chapter 7). Across these features, two adaptive workflows were implemented.

In *Perspectives* and *COTA*, adaptation is achieved through an interactive workflow in which the user provides minimal labels, the model fine-tunes, and the results are immediately updated in the visualization or the semantic search ranking. *COTA* adapts sentence-level semantic search to ensure results are relevant to the user’s emerging concepts, while *Perspectives* adapts document-level clustering to refine groupings. The technical approach centers on parameter-efficient fine-tuning of pre-trained models. This is achieved by few-shot fine-tuning of sentence transformer embedding models. This process leverages user-provided labels (validated clusters or sentence annotations) to generate a contrastive dataset and trains the model with a contrastive loss to efficiently refine the embedding space. This lightweight approach utilizes minimal GPU resources, ensuring adaptation is fast enough for interactive use.

For *Perspectives*, experiments showed that few-shot fine-tuning (2, 4, 8, 16 examples) consistently increased accuracy by 2–3 points over the zero-shot baseline, demonstrating that minimal user input could steer the clustering outcome. Similarly, for *COTA*, evaluation showed that accuracy steadily increased by 3–5 points with only a few labeled examples (4, 8, 16, 24, 32). However, the accuracy typically plateaued after 16 labeled examples per concept, limiting the effectiveness of more annotations.

In contrast to interactive fine-tuning, the *Annotation Assistant* employed a dynamic strategy to provide optimal-quality suggestions based on the total available annotated examples, rather than retraining the model after every user interaction. The dynamic three-phase annotation approach tries to provide optimal suggestions: First, initial suggestions are generated via zero-shot LLM prompting. Second, as minimal labels are gathered, suggestions improve via in-context few-shot prompting. Third, when sufficient data is annotated, the system automatically fine-tunes a sequence tagger (sentence transformer + BiLSTM + CRF head).

Evaluation across five sequential sentence classification datasets found that Mistral showed consistent, though modest (2–8 points), improvement in F1 score from zero-shot to few-shot prompting. A significant jump (11–36 points) in F1 score

was achieved by the fine-tuned sequence tagger. Analysis of the required labeled data volume found that the few-shot fine-tuned sequence tagger could surpass the performance of the LLM with ≈ 400 to ≈ 700 fully annotated documents (depending on the dataset, e.g., CSAbstract, Pubmed200k).

9.2 Limitations

The contributions of this dissertation, developed by combining NLP technologies and qualitative research methods and integrating them in DATS, have several limitations that warrant discussion.

Few-shot Learning: The integration of adaptive few-shot learning revealed constraints on performance scaling. Across *Perspectives* and *COTA*, few-shot fine-tuning consistently yielded only minor accuracy improvements over the zero-shot baseline (typically less than 5 accuracy points). While these modest gains meet the requirements of *adaptability* and *flexibility*, they do not yield the desired performance breakthroughs, highlighting the difficulty of scaling qualitative concepts with limited labels. Furthermore, the *Annotation Assistant* experiments revealed a significant performance gap between few-shot prompting and fine-tuned models. The LLM-based few-shot prompting (in-context learning) approach did not scale sufficiently to match the accuracy of specialized taggers. This fact underscores the need for dedicated model training to achieve state-of-the-art performance.

User Study: No formal user study was conducted to rigorously validate DATS' usability, methodological fitness, or overall impact on actual research outcomes. The evaluation of new features relied on simulations and proxy metrics, which only estimate real-world performance. This shortcoming is mitigated through close, continuous collaboration with digital humanities researchers within the University of Hamburg, where features and improvements were implemented iteratively based on project needs and feedback.

Benchmarks: The benchmarks conducted throughout this work informed technical implementation decisions (e.g., model selection and feature testing). The benchmarks covered a wide range of tasks: large language models were evaluated in a zero-shot setting across text classification (Section 5.5.5), information extraction (Section 5.5.6), and sequence labeling (Section 6.6.3) tasks for the *LLM Assistant*; embedding models were assessed on their learned document representations for clustering in *Perspectives* (Section 5.4.5); embedding models were evaluated on their sentence representations for *COTA* (Section 7.4.4); and models were evaluated for sequential sentence classification for the three-phase *Annotation Assistant* (Section 6.7.4). However, their value is inherently limited to a snapshot in time, as the rapid development of new models means their performance is quickly superseded. While we aimed to select datasets that closely resemble real-world QDA use cases, some discrepancies persist between the benchmark tasks and actual user workflows. In particular, some benchmarks use publicly available proxy datasets (e.g., MUC-4 for information extraction, or 20 Newsgroups for text classification), which are

not necessarily aligned with domain-specific discourse analysis projects, limiting the generalizability of the reported performance metrics. To mitigate this, future work requires monitoring and incorporating additional representative datasets to improve the benchmarks' relevance and applicability.

Accessibility: Despite aiming for intuitivity and *accessibility*, DATS remains a complex platform designed for expert users and advanced methodological use. The "*correct*" application of discourse analysis requires users to possess substantial methodological expertise in discourse analysis and Grounded Theory. Furthermore, some visualization features introduce complexity: a steep learning curve exists for *Whiteboards* for users unfamiliar with graph-based interfaces, and similarly for *Perspectives* and *COTA*, which use map-like interfaces for clustering. Effectively leveraging these advanced features may require considerable user training. Nevertheless, we believe that investing time in learning to use the tool, despite its initial complexity, will ultimately yield significant benefits for research endeavors. To mitigate this, future work requires implementing user onboarding through user guides and tutorial videos.

LLM Limitations: LLMs are prone to hallucination, generating false or factually incorrect information. Hence, all AI-augmented features (including *Perspectives*, *LLM Assistant*, and *COTA*) developed in this dissertation inherit these limitations. While LLMs can provide valuable insights, they are not perfect and may produce errors. It is crucial to understand and accept these limitations to avoid drawing false impressions or conclusions based on their outputs.

These models are known to exhibit biases due to potential biases embedded within their training data. They can occasionally generate factually incorrect outputs, favoring fluency over correctness. They may struggle with tasks that require complex reasoning, nuanced understanding of context, or common-sense knowledge. Furthermore, the accuracy and reliability of any findings depend heavily on the quality and appropriateness of the input data as well as the model's underlying assumptions and parameters. These issues can influence tasks such as document rewriting in *Perspectives* or tagging suggestions in the *LLM Assistant*.

Errors introduced during the preprocessing pipeline of DATS propagate through the system, e.g., issues in sentence splitting, which may affect the performance of clustering, suggestions, and analyses.

Given these inherent limitations, we urge users to exercise caution, careful consideration, and critical thinking when interpreting results. We consider the manual validation step, integrated into all AI-powered DATS features, very important. It allows users to retain control over the acceptance and integration of AI-generated suggestions into their analyses, ensuring human oversight throughout the workflow. We emphasize the importance of validating results through alternative methods such as manual coding or expert review. This ensures the accuracy and robustness of the findings and prevents an overreliance on automated tools that might otherwise steer the interpretive process in unintended directions.

9.3 Future Work

The development of the Discourse Analysis Tool Suite provides a strong foundation for future research, offering several exciting directions for improving AI-augmented qualitative data analysis in interdisciplinary digital humanities projects.

Adaptability: Future work could focus on exploring methods to close the performance gaps observed between few-shot learning and fine-tuned models, enhancing the adaptability. Specifically, other parameter-efficient fine-tuning techniques and adapter-based approaches could be explored for typical discourse analysis tasks to reduce training time while maintaining or improving performance. Furthermore, given the rapid rise of long-context LLMs with increased context windows of up to 1 million tokens at the time of writing (e.g., Google’s Gemini 3 Pro¹ and Meta’s Llama 4 Maverick²), future research could explore many-shot in-context learning as a potential adaptable alternative to fine-tuning. As demonstrated by Bertsch et al. (2025), such models can accommodate thousands of examples in the prompt, achieving performance competitive with fine-tuning while offering operational *flexibility* (instant task switching) without the high computational cost of weight updates.

Enhancing Tool Capabilities Beyond algorithmic refinements, future work could expand the functionality of the existing DATS extensions. For the *Concept Over Time Analysis* tool, an interesting direction would be to integrate multimodal support. By leveraging cross-modal sentence transformers powered by CLIP and adapting the user interface, *COTA* could be improved to identify and analyze concepts in image datasets, enabling visual discourse analysis. Furthermore, the analysis could be extended beyond timeline visualizations to encompass structural overviews of concept relationships.

Similarly, the *Whiteboards* feature could be enhanced to support larger datasets. Future improvements could investigate methods for hierarchical grouping, where multiple objects can be collapsed into a single node, and the creation of navigational layers. This would enable researchers to switch seamlessly between a "distant" overview of the entire project and a "close" view of specific research materials. Finally, integrating quantitative analysis results (such as dynamic plots and tables) as interactive nodes on the canvas would further aid qualitative interpretation.

Digital Discourse Analysis: The DATS framework is highly extensible, enabling the integration of new features to solve additional methodological challenges not addressed in this dissertation.

The integration of entity network analysis presents a relevant feature for discourse analysis. Entity networks can offer valuable insights into power and influence by visualizing who is talking about whom and highlighting relationships among individuals, institutions, and concepts. To facilitate this, future research

1. <https://ai.google.dev/gemini-api/docs/models>

2. <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>

could investigate integrating high-quality coreference resolution (linking pronouns to entities) and relation extraction (identifying relationships) to enable the automatic generation and visualization of accurate network diagrams.

Future work could also explore argument mining, utilizing relation extraction approaches to automatically identify claims, evidence, premises, and conclusions within text segments. This capability is highly relevant, as it enables large-scale reconstruction and visualization of the underlying argumentative and causal logic that governs a debate, thereby enhancing visibility into a discourse's structure and effects.

Finally, the automated analysis of narratives represents an interesting research direction. This could involve leveraging LLMs for zero-shot narrative extraction to identify key narrative components such as protagonist, antagonist, and resolution. Implementing such a feature would enable researchers to systematically analyze and compare the dominant narrative templates and underlying story structures used to frame complex societal issues.

AI Assistants: The evolution of user interfaces has shifted toward natural language interaction, which presents an interesting research direction.

The evolution of user interfaces has historically shifted from the *command line interface*, which requires users to master abstract commands, to *graphical user interfaces* that rely on visual metaphors and direct manipulation. This evolution continued with touch-based and gesture interfaces, and today, with the advent of large language models, the next major shift is toward *conversational interfaces* (chat interfaces). This shift is arguably the most natural way to communicate with a machine, as it allows users to state their high-level intent without having to study complex software functionality or graphical user interfaces.

This rise is reflected in general-purpose software (e.g., Microsoft Copilot³) and specialized tools (e.g., the analysis assistant in Nomic Atlas⁴). Hence, there is also potential for a *Discourse Analysis Agent* to revolutionize the DATS user experience. Such a DA agent could be designed to take a high-level interpretive goal (e.g., "*Find all documents challenging the climate debate that Alice has not reviewed*") and automatically break it down into a multi-step execution plan (e.g., filtering, semantic search, temporary tagging) using DATS' underlying API. This would lower the barrier to entry, making the complex, multi-functional tools within DATS (such as search, filtering, annotation, or *Perspectives* and *COTA*) more accessible.

Modern natural language processing — especially with large language models — has evolved from a niche, theory-focused field into a high-utility, applied technology. This shift means NLP is no longer limited to incremental self-improvement but is ready to be applied as a transformative tool in other domains. This work, rooted in interdisciplinary collaboration, successfully leveraged the technical foundations of modern NLP to address complex methodological challenges in the digital humanities.

The technological capacity to tackle societal challenges has never been greater. The current era is characterized by a promising convergence: on one side, highly

3. <https://copilot.microsoft.com/>

4. <https://atlas.nomic.ai/>

sophisticated and adaptable computational models; on the other, profound empirical problems demanding novel solutions. Therefore, I am of the firm opinion that now is the best time for interdisciplinary work! The key to unlocking this potential is not further technical refinement alone, but the establishment of bridges between disciplines. Many relevant challenges remain, and the necessary technological capabilities are available; consequently, the most critical future work lies in fostering and sustaining these promising intellectual collaborations. I encourage future computer science and digital humanities researchers to seek out and continue such interesting and productive partnerships, transforming technical potential into real-world solutions.

Appendices



Additional Material

Contents

A.1	Chapter 4 - Exploration, Search & Organization	221
A.1.1	Perspectives: Interactive Document Tagging in DATS	221
A.2	Chapter 5 - Annotation	222
A.2.1	Semi-automatic Sequential Sentence Classification	222

A.1 Chapter 4 - Exploration, Search & Organization

A.1.1 Perspectives: Interactive Document Tagging in DATS

Dataset	Instruction
emotion	Identify the main emotion expressed in the (summary keyphrases) of a song text
genre	Identify the main genre of the (summary keyphrases) of a song text
stars	Identify the sentiment of the (summary of an keyphrases of an) Amazon review
product	Identify the category of an Amazon product based on the review (summary keyphrases)
topic	Identify the topic or theme of the (summary keyphrases) of a news article
frame	Identify the framing of the (summary keyphrases) of a news article
blurb	Identifiziere das Genre, dass durch die (Zusammenfassung Schlüsselbegriffe) des Klappentextes beschrieben wird.
stance	Identify the stance towards the Israel-Palestine conflict of the (summary keyphrases) of a news article
bias	Identify the political leaning of the (summary keyphrases) of a news article

Table A.1: *Perspectives* evaluation – document embedding instructions. The instructions are provided to the embedding model. We adopted them to resemble those of the original authors.

Dataset	Prompt
emotion-summary	Write a concise summary (maximum 5 sentences) that focuses on the emotional tone of the following song lyrics. Analyze the lyrics to determine the main emotion being conveyed and describe how it is expressed. Conclude with an emotion categorization:
emotion-keyphrases	Generate keyphrases (max 5 phrases) that describe the emotional tone of the following song lyrics. Focus on phrases that reflect the emotional tone, mood, or feelings expressed in the lyrics.
genre-summary	Write a concise summary (maximum 5 sentences) that focuses on the genre of the following song lyrics. Analyze the lyrics to determine the musical genre, referencing stylistic elements, themes, or influences. Conclude with a genre categorization:
genre-keyphrases	Generate keyphrases (max 5 phrases) that describe the musical genre of the following song lyrics. Focus on phrases that reflect the genre's characteristics, style, or influences.
stars-summary	Write a concise summary (maximum 5 sentences) that focuses on the sentiment of the following Amazon review. Analyze the review to determine the main sentiment being conveyed and describe how it is expressed. Conclude with a sentiment categorization:
stars-keyphrases	Generate keyphrases (max 5 phrases) that describe the sentiment of the following Amazon review. Focus on phrases that reflect the sentiment, mood, or feelings expressed in the review.
product-summary	Write a concise summary (maximum 5 sentences) that focuses on the product categorization of the following Amazon review. Analyze the review to determine the discussed product's categorization, referencing its features, type, or purpose. Conclude with a categorization:
product-keyphrases	Generate keyphrases (max 5 phrases) that describe the product of the following Amazon review. Focus on phrases that reflect the product's type or category.
topic-summary	Write a concise summary (maximum 5 sentences) that focuses on the topic or theme of the following news article. Analyze the article to determine the main topic or theme being discussed. Conclude with a topic categorization:
topic-keyphrases	Generate keyphrases (max 5 phrases) that describe the topic of the following news article. Focus on phrases that reflect the main topic being discussed.
blurbs-summary	Schreibe eine prägnante Zusammenfassung (maximal 5 Sätze), die sich auf das Genre des folgenden Klappentextes konzentriert. Analysiere den Text, um das Genre zu bestimmen. Schließe mit einer allgemeinen Genre-Kategorisierung ab.
blurbs-keyphrases	Generiere Schlüsselbegriffe (max 5 Begriffe), die das Genre des folgenden Klappentextes beschreiben. Fokussiere dich auf Begriffe, die das Allgemeine Genre widerspiegeln.
stance-summary	Write a concise summary (maximum 5 sentences) that focuses on the stance of the following Reddit post. Analyze the post to determine whether it is Pro-Israel, Pro-Palestine, or Neutral. Conclude with a stance categorization:
stance-keyphrases	Generate keyphrases (max 5 phrases) that describe the stance of the following Reddit post. Focus on phrases that reflect the stance being discussed.
bias-summary	Write a concise summary (maximum 5 sentences) that focuses on the political leaning of the following news article. Analyze the article to determine the main political framing. Conclude with a political categorization:
bias-keyphrases	Generate keyphrases (max 5 phrases) that describe the political leaning of the following news article. Focus on phrases that reflect the political framing.

Table A.2: *Perspectives* evaluation – document rewriting prompts. The prompts are used to rewrite the original documents. We use the same prompts for topics and frames.

A.2 Chapter 5 - Annotation

A.2.1 Semi-automatic Sequential Sentence Classification

Category	Description
Background	Provides context or previous knowledge relevant to the research topic. Think of it as setting the stage for the study.
Method	Describes the procedures and techniques used in the research. This includes the study design, data collection, and analysis methods.
Objective	States the main goal or purpose of the research. This could be discussion, analysis, limitations, or concluding remarks.
Result	Presents the findings or outcomes of the research. This often includes statistical data, tables, and figures.
Other	Any sentence that doesn't fit into the above categories.

Table A.3: Three-stage *Annotation Assistant* evaluation – category system of the CSAbstract (Cohan et al. 2019) dataset.

Category	Description
Background	Provides context or previous knowledge relevant to the research topic. Think of it as setting the stage for the study.
Methods	Describes the procedures and techniques used in the research. This includes the study design, data collection, and analysis methods.
Objective	States the main goal or purpose of the research.
Results	Presents the findings or outcomes of the research. This often includes statistical data, tables, and figures.
Conclusions	Summarizes the key findings of the research and draw inferences from those findings. They provide closure to the abstract, summarizing the overall contribution of the research.

Table A.4: Three-stage *Annotation Assistant* evaluation – category system of the Pubmed200k (Dernoncourt and Lee 2017) dataset.

Category	Description
Fear	A feeling of apprehension or dread in response to a perceived threat or danger. It can range from mild anxiety to intense terror.
Disgust	A feeling of revulsion or aversion, often triggered by something perceived as unpleasant, unsanitary, or morally offensive.
Excited	A state of heightened arousal and positive anticipation. It often involves feelings of enthusiasm, eagerness, and energy.
Anger	A feeling of intense displeasure or hostility, often triggered by a perceived wrong or injustice. It can manifest as irritation, frustration, rage, or fury.
Surprise	A brief emotional state in response to an unexpected event. It can be positive, negative, or neutral, depending on the surprise.
Sadness	A feeling of sorrow, grief, or disappointment. It can range from mild melancholy to intense despair.
Joy	A feeling of happiness, contentment, or pleasure. It can manifest as amusement or love.
Neutral	A state of emotional balance or equilibrium, where no particular emotion is dominant.
Non-Neutral	Other or multiple of the above emotions are present

Table A.5: Three-stage *Annotation Assistant* evaluation – category system of the EmotionLines (Hsu et al. 2018) and DailyDialog (Yanran Li et al. 2017) datasets. EmotionLines also includes the non-neutral label.

Category	Description
Question	A comment with a question or a request seeking some form of feedback, help, or other kinds of responses. While the comment may contain a question mark, it is not required. For instance, it might be posed in the form of a statement but still soliciting a response. Also, not everything that has a question mark is automatically a Question. For instance, rhetorical questions are not seeking a response. Relation: This comment might be the first in a thread and have no relation to another comment. Or, it could be a clarifying or followup Question linking to any prior comment.
Answer	A comment that is responding to a Question by answering the question or fulfilling the request. There can be more than one Answer responding to a Question. Relation: An Answer is always linked to a Question.
Announcement	A comment that is presenting some new information to the community, such as a piece of news, a link to something, a story, an opinion, a review, or insight. Relation: This comment has no relation to a prior comment and is always the initial post in a thread.
Agreement	A comment that is expressing agreement with some information presented in a prior comment. It can be agreeing with a point made, providing supporting evidence, providing a positive example or experience, or confirming or acknowledging a point made. Relation: This comment is always linked to a prior comment to which it is agreeing.
Appreciation	A comment that is expressing thanks, appreciation, excitement, or praise in response to another comment. In contrast to Agreement, it is not evaluating the merits of the points raised. Comments of this category are more interpersonal as opposed to informational. Relation: This comment is always linked to a prior comment for which it is expressing appreciation.
Disagreement	A comment that is correcting, criticizing, contradicting, or objecting to a point made in a prior comment. It can also be providing evidence to support its disagreement, such as an example or contrary anecdote. Relation: This comment is always linked to a prior comment to which it is disagreeing.
Negative reaction	A comment that is expressing a negative reaction to a previous comment, such as attacking or mocking the commenter, or expressing emotions like disgust, derision, or anger, to the contents of the prior comment. This comment is not discussing the merits of the points made in a prior comment or trying to correct them. Relation: This comment is always linked to a prior comment to which it is negatively reacting.
Elaboration	A comment that is adding additional information on to another comment. Oftentimes, one can imagine it simply appended to the end of the comment it elaborates on. One can elaborate on many kinds of comments, for instance, a questionasker elaborating on their question to provide more context, or someone elaborating on an answer to add more information. Relation: This comment is always linked to a prior comment upon which it is elaborating.
Humor	This comment is primarily a joke, a piece of sarcasm, or a pun intended to get a laugh or be silly but not trying to add information. If a comment is sarcastic but using sarcasm to make a point or provide feedback, then it may belong in a different category. Relation: At times, this comment links to another comment but other times it may not be responding to anything.
Other	A comment that does not fit any of the previous definitions.

Table A.6: Three-stage *Annotation Assistant* evaluation – category system of the Coarse Discourse (Zhang et al. 2017) dataset.

References

- Giuseppe Abrami, Manuel Stoeckel, and Alexander Mehler. 2020. TextAnnotator: A UIMA Based Tool for the Simultaneous and Collaborative Annotation of Texts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, 891–900. Marseille, France: European Language Resources Association. (Cited on page 77).
- David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1988. A learning algorithm for Boltzmann machines. In *Neurocomputing: Foundations of Research*, 635–649. Cambridge, MA, USA: MIT Press. (Cited on page 27).
- Charu C. Aggarwal and ChengXiang Zhai. 2012. A Survey of Text Clustering Algorithms. In *Mining Text Data*, edited by Charu C. Aggarwal and ChengXiang Zhai, 77–128. Boston, MA, USA: Springer US. (Cited on page 23).
- Hasin Jawad Ali, Ajwad Abrar, S. M. Hozaifa Hossain, and M. Firoz Mridha. 2025. Social media polarization during conflict: Insights from an ideological stance dataset on Israel-Palestine Reddit comments. arXiv: 2502.00414 [cs.CL]. (Cited on page 115).
- Dimo Angelov. 2020. Top2Vec: Distributed Representations of Topics. arXiv: 2008.09470 [cs.CL]. (Cited on pages 23, 106).
- Dimo Angelov and Diana Inkpen. 2024. Topic Modeling: Contextual Token Embeddings Are All You Need. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, edited by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, 13528–13539. Miami, FL, USA: Association for Computational Linguistics. (Cited on pages 105, 107, 113).
- Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2002. Semi-supervised Clustering by Seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 27–34. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (Cited on page 107).
- Baur, Nina and Blasius, Jörg, eds. 2022a. Handbuch Methoden der empirischen Sozialforschung. Wiesbaden, Germany: Springer VS. (Cited on page 46).
- . 2022b. Methoden der empirischen Sozialforschung. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 1–32. Wiesbaden, Germany: Springer Fachmedien. (Cited on pages 46 sqq., 172).
- Sebastian Bechmann. 2007. Gesundheitssemantiken der Moderne: Eine Diskursanalyse der Debatten über die Reform der Krankenversicherung. Berlin, Germany: Edition Sigma. (Cited on page 4).

- Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* (USA) 35 (8): 1798–1828. (Cited on page 21).
- Basil Bernstein. 1971. *Class, Codes and Control. Theoretical Studies Towards a Sociology of Language*. New York, USA: Routledge. (Cited on page 62).
- Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2025. In-Context Learning with Long-Context Models: An In-Depth Exploration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, edited by Luis Chiruzzo, Alan Ritter, and Lu Wang, 1:12119–12149. Albuquerque, NM, USA: Association for Computational Linguistics. (Cited on page 215).
- Timothy W. Bickmore, Ha Trinh, Stefan Olafsson, Teresa K. O’Leary, Reza Asadi, Nathaniel M. Rickles, and Ricardo Cruz. 2018. Patient and Consumer Safety Risks When Using Conversational Assistants for Medical Information: An Observational Study of Siri, Alexa, and Google Assistant. *J Med Internet Res* 20 (9): e11510. (Cited on page 29).
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research* 3:993–1022. (Cited on page 106).
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5:135–146. (Cited on page 21).
- Kevin Bönisch, Giuseppe Abrami, and Alexander Mehler. 2025. Towards Unified, Dynamic and Annotation-based Visualisations and Exploration of Annotated Big Data Corpora with the Help of Unified Corpus Explorer. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, edited by Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao, 522–534. Albuquerque, NM, USA: Association for Computational Linguistics. (Cited on page 78).
- Léon Bottou. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of COMPSTAT’2010*, edited by Yves Lechevallier and Gilbert Saporta, 177–186. Heidelberg, Germany: Physica-Verlag HD. (Cited on page 22).
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, edited by Lluís Màrquez, Chris Callison-Burch, and Jian Su, 632–642. Lisbon, Portugal: Association for Computational Linguistics. (Cited on page 39).
- Bröckling, Ulrich, Krasmann, Susanne, and Lemke, Thomas, eds. 2000. *Gouvernementalität der Gegenwart: Studien zur Ökonomisierung des Sozialen*. 1490. Frankfurt am Main, Germany: Suhrkamp. (Cited on page 62).

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 33:1877–1901. Curran Associates Inc. (Cited on pages 25 sq.).
- Claudia Brunner. 2011. Wissensobjekt Selbstmordattentat: Epistemische Gewalt und okzidentalistische Selbstvergewisserung in der Terrorismusforschung. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on page 4).
- Antony Bryant, Kathy Charmaz, and Lora Bex Lempert. 2007. Asking Questions of the Data: Memo Writing in the Grounded Theory Tradition. In *The SAGE Handbook of Grounded Theory*, 245–264. SAGE Publications Ltd. (Cited on page 53).
- Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, edited by Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, 160–172. Berlin, Heidelberg, Germany: Springer. (Cited on pages 15, 24, 106, 167, 210).
- Soumen Chakrabarti, Martin van den Berg, and Byron Dom. 1999. Focused crawling: a new approach to topic-specific Web resource discovery. *Comput. Netw. (USA)* 31 (11–16): 1623–1640. (Cited on page 87).
- Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation*, edited by Guy Emerson, Natalie Schluter, Gabriel Stanovsky, Ritesh Kumar, Alexis Palmer, Nathan Schneider, Siddharth Singh, and Shyam Ratan, 1094–1106. Seattle, WA, USA: Association for Computational Linguistics. (Cited on pages 15, 103).
- Gabriela B. Christmann. 2004. Dresdens Glanz, Stolz der Dresdner: Lokale Kommunikation, Stadtkultur und städtische Identität. Wiesbaden, Germany: Deutscher Universitäts-Verlag. (Cited on page 4).
- Adele E. Clarke, Carrie Friese, and Rachel S. Washburn. 2017. Situational analysis: Grounded theory after the interpretive turn. Sage publications. (Cited on page 197).
- Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained Language Models for Sequential Sentence Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3693–3699. Hong Kong, China: Association for Computational Linguistics. (Cited on pages 161, 223).

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 8440–8451. Online: Association for Computational Linguistics. (Cited on page 115).
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, edited by Joaquin Quiñero-Candela, Ido Dagan, Bernardo Magnini, and Florence d’Alché-Buc, 177–190. Berlin, Heidelberg: Springer Berlin Heidelberg. (Cited on page 39).
- Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, edited by Greg Kondrak and Taro Watanabe, 2:308–313. Taipei, Taiwan: Asian Federation of Natural Language Processing. (Cited on pages 161, 223).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, edited by Jill Burstein, Christy Doran, and Tamar Solorio, 4171–4186. Minneapolis, MN, USA: Association for Computational Linguistics. (Cited on pages 21, 25, 36, 142).
- Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A Few-shot Named Entity Recognition Dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 3198–3213. Online: Association for Computational Linguistics. (Cited on pages 150, 161).
- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. 2024. Xgrammar: Flexible and efficient structured generation engine for large language models. *Proceedings of Machine Learning and Systems 7*, (cited on page 84).
- Isabel Eiser, Tim Fischer, Florian Schneider, Gertraud Koch, Chris Biemann, and Fynn Petersen Frey. 2023. Open Science Prinzipien und interdisziplinäre Kollaboration in D-WISE: Zwischen Hermeneutik und Digitaler Methode in der Diskursanalyse. In *9. Tagung des Verbands Digital Humanities im deutschsprachigen Raum*, 252–257. Trier, Luxemburg: Zenodo. (Cited on page 199).
- Antke A. Engel. 2009. Bilder von Sexualität und Ökonomie: Queere kulturelle Politiken im Neoliberalismus. Bielefeld, Germany: transcript. (Cited on page 62).
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231. Portland, Oregon: AAAI Press. (Cited on page 24).

- Norman Fairclough, Jane Mulderrig, and Ruth Wodak. 2011. Critical Discourse Analysis. In *Discourse Studies: A Multidisciplinary Introduction*, 2nd, edited by Teun A. van Dijk, 357–378. London: SAGE Publications Ltd. (Cited on page 62).
- Zheng Fang, Lama Alqazlan, Du Liu, Yulan He, and Rob Procter. 2023. A User-Centered, Interactive, Human-in-the-Loop Topic Modelling System. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Andreas Vlachos and Isabelle Augenstein, 505–522. Dubrovnik, Croatia: Association for Computational Linguistics. (Cited on page 107).
- John R. Firth. 1957. A Synopsis of Linguistic Theory, 1930–1955. In *Studies in Linguistic Analysis*, 1–32. Reprinted in Palmer, F. (ed. 1968) *Selected Papers of J. R. Firth*, Longman, Harlow. Oxford. (Cited on page 21).
- Tim Fischer and Chris Biemann. 2024. Exploring Large Language Models for Qualitative Data Analysis. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, edited by Mika Hämmäläinen, Emily Öhman, So Miyagawa, Khalid Alnajjar, and Yuri Bizzoni, 423–437. Miami, FL, USA: Association for Computational Linguistics. (Cited on pages 14, 16, 96, 121, 138, 147, 210).
- . 2025. Semi-automatic Sequential Sentence Classification in the Discourse Analysis Tool Suite. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, edited by Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao, 151–162. Albuquerque, NM, USA: Association for Computational Linguistics. (Cited on pages 14, 16, 138, 154, 163, 211).
- . 2026. Perspectives - Interactive Document Clustering in the Discourse Analysis Tool Suite. Note: under review. arXiv: 2602.15540 [cs.CL]. (Cited on pages 13, 16, 96, 105, 111, 113 sq., 118, 210).
- Tim Fischer, Florian Schneider, Robert Geislinger, Florian Helfer, Gertraud Koch, and Chris Biemann. 2024. Concept Over Time Analysis: Unveiling Temporal Patterns for Qualitative Data Analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, edited by Kai-Wei Chang, Annie Lee, and Nazneen Rajani, 148–157. Mexico City, Mexico: Association for Computational Linguistics. (Cited on pages 14, 16, 171, 177, 181 sq., 184 sq., 190, 211).
- Tim Fischer, Florian Schneider, Fynn Petersen-Frey, Anja Silvia Mollah Haque, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2024. Extending the Discourse Analysis Tool Suite with Whiteboards for Visual Qualitative Analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, edited by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, 7017–7022. Torino, Italia: ELRA and ICCL. (Cited on pages 14, 16, 191, 197, 199, 202 sqq., 211).
- Elisabeth Fittschen, Tim Fischer, Daniel Brühl, Julia Spahr, Yuliia Lysa, and Phuoc Thang Le. 2024. AnnoPlot: Interactive Visualizations of Text Annotations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Nikolaos Aletras and Orphee De Clercq, 106–114. St. Julians, Malta: Association for Computational Linguistics. (Cited on pages 15, 165 sq.).

- Uwe Flick. 2022. Gütekriterien qualitativer Sozialforschung. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 533–547. Wiesbaden, Germany: Springer Fachmedien. (Cited on page 49).
- Michel Foucault. 1972. *Archaeology of Knowledge*. New York, USA: Routledge. (Cited on page 58).
- Lina Franken. 2020. Kulturwissenschaftliches digitales Arbeiten: Qualitative Forschung als digitale Handarbeit? *Berliner Blätter* 82:107–118. (Cited on page 201).
- Lina Franken, Gertraud Koch, and Heike Zinsmeister. 2020. Annotationen als Instrument der Strukturierung. In *Annotations in Scholarly Editions and Research*, 89–108. De Gruyter. (Cited on pages 87, 98, 100).
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. arXiv: 2101.00027 [cs.CL]. (Cited on page 28).
- Evelyn Gius, Jan Christoph Meister, Malte Meister, Marco Petris, Mareike Schumacher, and Dominik Gerstorfer. 2023. CATMA 7 (Version 7.0). V. 7.0. Zenodo. (Cited on page 75).
- Carsten Görg, Zhicheng Liu, and John Stasko. 2014. Reflections on the Evolution of the Jigsaw Visual Analytics System. *Information Visualization* 13 (4): 336–345. (Cited on page 198).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. 2024. The Llama 3 Herd of Models. arXiv: 2407.21783 [cs.AI]. (Cited on pages 25, 41, 125, 150, 159).
- Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18 (5): 602–610. (Cited on page 159).
- Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv: 2203.05794 [cs.CL]. (Cited on pages 23, 105 sqq., 180).
- Christian Haase, Saba Anwar, Seid Muhie Yimam, Alexander Friedrich, and Chris Biemann. 2021. SCoT: Sense Clustering over Time: a tool for the analysis of lexical change. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Dimitra Gkatzia and Djamé Seddah, 198–204. Online: Association for Computational Linguistics. (Cited on page 180).
- Zellig S. Harris. 1954. Distributional Structure. *WORD* 10 (2-3): 146–162. (Cited on page 21).
- Hans Ole Hatzel, Fynn Petersen-Frey, Tim Fischer, and Chris Biemann. 2023. Dimensions of Similarity: Towards Interpretable Dimension-Based Text Similarity. In *Frontiers in Artificial Intelligence and Applications*, edited by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Rădulescu, 1004–1011. Kraków, Poland: IOS Press. (Cited on pages 15, 103 sq.).
- Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, and Percy Liang. 2023. Foundation Models and Fair Use. *Journal of Machine Learning Research* 24 (400): 1–79. (Cited on page 30).

- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, and Mantas Mazeika et al. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*. Vienna, Austria (Online). (Cited on pages 29, 122).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* (Cambridge, MA, USA) 9 (8): 1735–1780. (Cited on page 159).
- Valentin Hofmann, Pratyusha Ria Kalluri, Dan Jurafsky, and Sharese King. 2024. AI generates covertly racist decisions about people based on their dialect. *Nature* 633 (8028): 147–154. (Cited on page 29).
- Ari Holtzman, Jan Buys, Liwei Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Generation. *International Conference on Learning Representations* (Online), (cited on page 27).
- Andreas Holzinger. 2016. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics* 3 (2): 119–131. (Cited on pages 8, 177).
- H. Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24 (6): 417–441. (Cited on page 23).
- Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. EmotionLines: An Emotion Corpus of Multi-Party Conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, edited by Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association. (Cited on pages 162, 224).
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. Online. (Cited on pages 20, 35, 117).
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv: 1508.01991 [cs.CL]. (Cited on page 159).
- Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating Lexical Priors into Topic Models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Walter Daelemans, 204–213. Avignon, France: Association for Computational Linguistics. (Cited on page 107).
- Jäger, Siegfried, ed. 1992. BrandSätze. Rassismus im Alltag. 2. durchgesehene. Duisburg, Germany: Duisburger Institut für Sprach- und Sozialforschung (DISS). (Cited on page 2).
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, and Devendra Singh Chaplot et al. 2023. Mistral 7B. arXiv: 2310.06825 [cs.CL]. (Cited on pages 42, 125, 150, 159).

- Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models. 3rd. Online manuscript released August 24, 2025. (Cited on pages 25, 30, 32, 87).
- Udo Kelle. 2022. Mixed Methods. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 163–177. Wiesbaden, Germany: Springer Fachmedien. (Cited on page 48).
- Reiner Keller. 1998. Müll – Die gesellschaftliche Konstruktion des Wertvollen. Opladen, Germany: Westdeutscher Verlag. (Cited on page 4).
- . 2010. Wissenssoziologische Diskursanalyse und Systemtheorie. In *Die Methodologien des Systems: Wie kommt man zum Fall und wie dahinter?*, edited by René John, Anna Henkel, and Jana Rückert-John, 241–272. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on page 172).
- . 2011a. Diskursforschung: Eine Einführung für SozialwissenschaftlerInnen. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on pages 2 sq., 46, 60, 97 sq., 139, 192 sq.).
- . 2011b. The Sociology of Knowledge Approach to Discourse. *Human Studies* 34:43–65. (Cited on pages 1 sq., 6, 46, 62, 65 sq., 138, 174, 192 sq.).
- . 2019. Grundlagen der Wissenssoziologischen Diskursanalyse. In *Diskursanalyse für die Kommunikationswissenschaft: Theorie, Vorgehen, Erweiterungen*, edited by Thomas Wiedemann and Christine Lohmeier, 35–60. Wiesbaden, Germany: Springer Fachmedien. (Cited on page 172).
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics*, 5–9. Santa Fe, NM, USA: Association for Computational Linguistics. (Cited on page 77).
- Gertraud Koch, Chris Biemann, Isabel Eiser, Tim Fischer, Florian Schneider, Teresa Stumpf, and Alejandra Tijerina García. 2022. D-WISE Tool Suite for the Sociology of Knowledge Approach to Discourse. In *Culture and Computing*, edited by Matthias Rauterberg, 68–83. Cham, Germany: Springer International Publishing. (Cited on pages 8, 69).
- Benjamin Krautter. 2024. The Scales of (Computational) Literary Studies: Martin Mueller’s Concept of Scalable Reading in Theory and Practice. In *Zoomland. Exploring Scale in Digital History and Humanities*, edited by Florentina Armaselu and Andreas Fickers, 261–286. Berlin, Germany: De Gruyter. (Cited on page 10).
- Dagmar Krebs and Natalja Menold. 2022. Gütekriterien quantitativer Sozialforschung. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 549–565. Wiesbaden, Germany: Springer Fachmedien. (Cited on page 49).
- Maria Kunilovskaya and Alistair Plum. 2021. Text Preprocessing and its Implications in a Digital Humanities Project. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, edited by Souhila Djabri, Dinara Gimadi, Tsvetomila Mihaylova, and Ivelina Nikolova-Koleva, 85–93. Online: INCOMA Ltd. (Cited on page 88).

- Ronald Kurt and Regine Herbrink. 2022. Sozialwissenschaftliche Hermeneutik und hermeneutische Wissenssoziologie. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 607–626. Wiesbaden, Germany: Springer Fachmedien. (Cited on pages 46, 54 sqq., 192 sqq.).
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 611–626. Koblenz, Germany: Association for Computing Machinery. (Cited on pages 36, 84).
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (Cited on pages 41, 159).
- Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, edited by Eric P. Xing and Tony Jebara, 32:1188–1196. Beijing, China: PMLR. (Cited on page 106).
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. arXiv: 2405.17428 [cs.CL]. (Cited on page 159).
- Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2020. A Dataset of German Legal Documents for Named Entity Recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, 4478–4485. Marseille, France: European Language Resources Association. (Cited on page 151).
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, edited by Greg Kondrak and Taro Watanabe, 1:986–995. Taipei, Taiwan: Asian Federation of Natural Language Processing. (Cited on pages 162, 224).
- Yucheng Li, Frank Guerin, and Chenghua Lin. 2024. LatestEval: addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and 14th Symposium on Educational Advances in Artificial Intelligence*, 38:18600–18607. Vancouver, Canada: AAAI Press. (Cited on page 127).
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, and Dilara Soylu et al. 2023. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research* 1525 (1): 140–146. (Cited on pages 29, 122).
- Yvonna S. Lincoln and Egon Guba. 1985. *Naturalistic Inquiry*. Newbury Park, Calif.: SAGE Publications, Inc. (Cited on page 50).

- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* (New York, NY, USA) 55 (9). (Cited on page 26).
- Siyi Liu, Lei Guo, Kate Mays, Margrit Betke, and Derry Tanti Wijaya. 2019. Detecting Frames in News Headlines and Its Application to Analyzing News Framing Trends Surrounding U.S. Gun Violence. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, edited by Mohit Bansal and Aline Villavicencio, 504–514. Hong Kong, China: ACL. (Cited on page 114).
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv: 1907.11692 [cs.CL]. (Cited on page 25).
- Shayne Longpre, Robert Mahari, Ariel Lee, Campbell Lund, Hamidah Oderinwale, William Brannon, Nayan Saxena, Naana Obeng-Marnu, Tobin South, Cole Hunter, Kevin Klyman, Christopher Klamm, Hailey Schoelkopf, Nikhil Singh, Manuel Cherep, Ahmad Mustafa Anis, An Dinh, Caroline Chitongo, Da Yin, Damien Sileo, Deividas Mataciunas, Diganta Misra, Emad Alghamdi, Enrico Shippole, Jianguo Zhang, Joanna Materzynska, Kun Qian, Kush Tiwary, Lester Miranda, Manan Dey, Minnie Liang, Mohammed Hamdy, Niklas Muennighoff, Seonghyeon Ye, Seungone Kim, Shrestha Mohanty, Vipul Gupta, Vivek Sharma, Vu Minh Chien, Xuhui Zhou, Yizhi Li, Caiming Xiong, Luis Villa, Stella Biderman, Hanlin Li, Daphne Ippolito, Sara Hooker, Jad Kabbara, Sandy Pentland, and Data Provenance Initiative. 2024. Consent in crisis: the rapid decline of the AI data commons. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates Inc. (Cited on page 30).
- Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A Pretrainer’s Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, edited by Kevin Duh, Helena Gomez, and Steven Bethard, 3245–3276. Mexico City, Mexico: Association for Computational Linguistics. (Cited on page 29).
- B. T. Lowerre. 1976. The Harpy speech recognition system. PhD diss., Carnegie Mellon University, Pennsylvania. (Cited on page 27).
- Maasen, Sabine, Mayerhauser, Torsten, and Renggli, Cornelia, eds. 2006. Bilder als Diskurse – Bilddiskurse. Weilerswist, Germany: Velbrück. (Cited on page 62).
- J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297. University of California Press. (Cited on page 24).
- Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv: 1802.03426 [stat.ML]. (Cited on pages 15, 23, 106, 166, 187, 210).
- MetaAI. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. Meta Blog. Accessed: 2025-02-25. (Cited on page 25).

- Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations*. Scottsdale, AZ, USA. (Cited on page 21).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, edited by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, 26:3111–3119. Lake Tahoe, Nevada: Curran Associates Inc. (Cited on pages 21 sq.).
- George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* (New York, NY, USA) 38 (11): 39–41. (Cited on pages 142, 179).
- Rishabh Misra. 2022. News Category Dataset. arXiv: 2209.11429 [cs.CL]. (Cited on page 183).
- Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, edited by Adam Fisch, Alon Talmor, Danqi Chen, Eunsol Choi, Minjoon Seo, Patrick Lewis, Robin Jia, and Sewon Min, 42–50. Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on page 129).
- Franco Moretti. 2016. Distant Reading. Konstanz, Germany: Konstanz University Press. (Cited on page 7).
- Michał Moskal, Harsha Nori, Hudson Cooper, and Loc Huynh. 2025. LLGuidance: Making Structured Outputs Go Brrr. LLGuidance Blog. Accessed: 2026-02-25. (Cited on page 84).
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Andreas Vlachos and Isabelle Augenstein, 2014–2037. Dubrovnik, Croatia: Association for Computational Linguistics. (Cited on pages 115, 122, 159).
- Laura K. Nelson. 2020. Computational Grounded Theory: A Methodological Framework. *Sociological Methods & Research* 49 (1): 3–42. (Cited on pages 78, 141, 143).
- OpenAI. 2023. GPT-4 Technical Report. arXiv: 2303.08774 [cs.CL]. (Cited on page 25).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 27730–27744. Red Hook, NY, USA: Curran Associates Inc. (Cited on pages 28 sq.).
- Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22 (10): 1345–1359. (Cited on page 36).

- Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional LSTM-CRF for Named Entity Recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, edited by Stephen Politzer-Ahles, Yu-Yin Hsu, Chu-Ren Huang, and Yao Yao. Hong Kong, China: Association for Computational Linguistics. (Cited on page 159).
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 79–86. Association for Computational Linguistics. (Cited on page 39).
- Karl Pearson. 1901. LIII. *On lines and planes of closest fit to systems of points in space*. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (11): 559–572. (Cited on page 23).
- Michel Pêcheux. 1995. Automatic Discourse Analysis. Edited by Tony Hak and Niels Helsloot. 5:264. Translation and edition of the original French text *Analyse automatique du discours* (1969). Rodopi. (Cited on pages 62, 172).
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. Doha, Qatar: Association for Computational Linguistics. (Cited on page 21).
- Fynn Petersen-Frey and Chris Biemann. 2024. Dataset of Quotation Attribution in German News Articles. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, edited by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, 4412–4422. Torino, Italia: ELRA and ICCL. (Cited on page 151).
- Fynn Petersen-Frey, Tim Fischer, Florian Schneider, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2023. From Qualitative to Quantitative Research: Semi-Automatic Annotation Scaling in the Digital Humanities. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, edited by Munir Georges, Aaricia Herygers, Annemarie Friedrich, and Benjamin Roth, 52–62. Ingolstadt, Germany: Association for Computational Linguistics. (Cited on pages 15, 145 sq.).
- Dang Pham and Tuan M. V. Le. 2021. Neural Topic Models for Hierarchical Topic Detection and Visualization. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, edited by Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano, 35–51. Cham, Germany: Springer International Publishing. (Cited on page 113).
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, edited by Marina Meila and Tong Zhang, 139:8748–8763. Online: PMLR. (Cited on pages 90, 186).
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv: 2212.04356 [eess.AS]. (Cited on page 90).

- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc. (Cited on page 29).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21 (1). (Cited on page 28).
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, edited by Iryna Gurevych and Yusuke Miyao, 784–789. Melbourne, Australia: Association for Computational Linguistics. (Cited on page 129).
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, edited by Jian Su, Kevin Duh, and Xavier Carreras, 2383–2392. Austin, TX, USA: Association for Computational Linguistics. (Cited on page 129).
- Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*. (Cited on page 41).
- Zeeshan Rasheed, Muhammad Waseem, Aakash Ahmad, Kai-Kristian Kemell, Wang Xiaofeng, Anh Nguyen Duc, and Pekka Abrahamsson. 2025. Can Large Language Models Serve as Data Analysts? A Multi-Agent Assisted Approach for Qualitative Data Analysis. arXiv: 2402.01386 [cs. SE]. (Cited on page 123).
- Jo Reichertz. 2022. Empirische Sozialforschung und soziologische Theorie. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 69–86. Wiesbaden, Germany: Springer Fachmedien. (Cited on pages 46 sq.).
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3982–3992. Hong Kong, China: Association for Computational Linguistics. (Cited on pages 39, 90, 146, 178).
- . 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 4512–4525. Online: Association for Computational Linguistics. (Cited on pages 90, 186).
- Steffen Remus, Rami Aly, and Chris Biemann. 2019. GermEval 2019 Task 1: Hierarchical Classification of Blurbs. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, 280–292. Erlangen, Germany: German Society for Computational Linguistics & Language Technology. (Cited on page 114).

- Arik Reuter, Anton Thielmann, Christoph Weisser, Sebastian Fischer, and Benjamin Säfken. 2024. GPTopic: Dynamic and Interactive Topic Representations. arXiv: 2403.03628 [cs.CL]. (Cited on pages 105, 107).
- Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22 (3): 400–407. (Cited on page 22).
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (6088): 533–536. (Cited on page 27).
- M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. 1998. A Bayesian Approach to Filtering Junk E-Mail. In *Proceedings of Annual Conference on Artificial Intelligence: Workshop on Learning for Text Categorization*. Madison, WI, USA. (Cited on page 39).
- Ina Schmied-Knittel. 2008. Satanismus und ritueller Missbrauch: Eine wissenssoziologische Diskursanalyse. Würzburg, Germany: Ergon. (Cited on page 4).
- Florian Schneider, Tim Fischer, Fynn Petersen-Frey, Isabel Eiser, Gertraud Koch, and Chris Biemann. 2023. The D-WISE Tool Suite: Multi-Modal Machine-Learning-Powered Tools Supporting and Enhancing Digital Discourse Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, edited by Danushka Bollegala, Ruihong Huang, and Alan Ritter, 328–335. Toronto, Canada: Association for Computational Linguistics. (Cited on pages 5, 13, 16, 69 sq., 82, 88 sq.).
- Florian Schrott, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 815–823. (Cited on page 40).
- Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, Hyojung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. 2025. The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. arXiv: 2406.06608 [cs.CL]. (Cited on page 26).
- Kyle Seelman, Mozhi Zhang, and Jordan Boyd-Graber. 2024. Labeled Interactive Topic Models. arXiv: 2311.09438 [cs.LG]. (Cited on page 107).
- Eyal Shnarch, Alon Halfon, Ariel Gera, Marina Danilevsky, Yannis Katsis, Leshem Choshen, Martin Santillan Cooper, Dina Epelboim, Zheng Zhang, Dakuo Wang, Lucy Yip, Liat Ein-Dor, Lena Dankin, Ilya Shnayderman, Ranit Aharonov, Yunyao Li, Naftali Liberman, Philip Levin Slesarev, Gwilym Newton, Shila Ofek-Koifman, Noam Slonim, and Yoav Katz. 2022. Label Sleuth: From Unlabeled Text to a Classifier in a Few Hours. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, edited by Wanxiang Che and Ekaterina Shutova, 159–168. Abu Dhabi, UAE: Association for Computational Linguistics. (Cited on page 180).

- Rainer Simon, Elton Barker, Leif Isaksen, and Pau de Soto Cañamares. 2017. Linked data annotation without the pointy brackets: Introducing Recogito 2. *Journal of Map & Geography Libraries* 13 (1): 111–132. (Cited on page 78).
- Tobias Singelnstein. 2009. Diskurs und Kriminalität. Berlin, Germany: Duncker & Humblot. (Cited on page 4).
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, edited by David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, 1631–1642. Seattle, WA, USA: Association for Computational Linguistics. (Cited on page 183).
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, 33:16857–16867. Curran Associates Inc. (Cited on page 159).
- Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28 (1): 11–21. (Cited on page 21).
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, and Abubakar Abid et al. 2023. Beyond the imitation game: quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research* 2023 (5): 1–95. (Cited on pages 29, 122).
- Anselm Strauss and Juliet Corbin. 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques. SAGE Publications, Inc. (Cited on page 50).
- Anselm Strauss, Juliet Corbin, Solveigh Niewiarra, and Heiner Legewie. 1996. Grounded Theory: Grundlagen Qualitativer Sozialforschung. Beltz, Psychologie-Verlag-Union Weinheim. (Cited on pages 3, 50, 52, 139, 179).
- Jörg Strübing. 2022. Grounded Theory und Theoretical Sampling. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 587–606. Wiesbaden, Germany: Springer Fachmedien. (Cited on pages 46, 50 sqq.).
- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 1102–1121. Toronto, Canada: Association for Computational Linguistics. (Cited on pages 40, 106).
- Beth M. Sundheim. 1992. Overview of the Fourth Message Understanding Evaluation and Conference. In *Proceedings of the Fourth Message Understanding Conference*. McLean, VA, USA. (Cited on page 130).
- Wilson L. Taylor. 1953. Cloze Procedure: A New Tool for Measuring Readability. *Journalism Quarterly* 30 (4): 415–433. (Cited on page 37).
- Gemma Team. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv: 2408.00118 [cs.CL]. (Cited on pages 42, 125, 150, 159).
- . 2025. Gemma 3 Technical Report. arXiv: 2503.19786 [cs.CL]. (Cited on pages 42, 85, 115).

- Kimi Team. 2025. Kimi K2: Open Agentic Intelligence. arXiv: 2507.20534 [cs.LG]. (Cited on pages 24 sq.).
- Mistral AI Team. 2024. Mistral Nemo. Mistral Blog. Accessed: 2026-02-25. (Cited on pages 42, 125, 150, 159).
- Anton F. Thielmann, Christoph Weisser, and Benjamin Säfken. 2024. Human in the Loop: How to Effectively Create Coherent Topics by Manually Labeling Only a Few Documents per Class. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, edited by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, 8395–8405. Torino, Italia: ELRA and ICCL. (Cited on pages 112, 119).
- William Timkey and Marten van Schijndel. 2021. All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 4527–4546. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on page 39).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, and Marie-Anne Lachaux et al. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv: 2302.13971 [cs.CL]. (Cited on pages 25, 41, 125, 150, 159).
- Boris Traue, Lisa Pfahl, and Lena Schürmann. 2022. Diskursanalyse. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 627–645. Wiesbaden, Germany: Springer Fachmedien. (Cited on pages 1 sq., 46, 57 sqq., 96 sqq., 138, 172, 192, 194).
- Miriam Trübner and Andreas Mühlichen. 2022. Big Data. In *Handbuch Methoden der empirischen Sozialforschung*, edited by Nina Baur and Jörg Blasius, 179–193. Wiesbaden, Germany: Springer Fachmedien. (Cited on page 1).
- Inga Truschkat. 2008. Kompetenzdiskurs und Bewerbungsgespräche: Eine Dispositivanalyse (neuer) Rationalitäten sozialer Differenzierung. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on pages 2, 4).
- Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. arXiv: 2209.11055 [cs.CL]. (Cited on pages 106, 179, 182 sq., 185).
- Peter Ullrich. 2008. Die Linke, Israel und Palästina: Nahostdiskurse in Großbritannien und Deutschland. Berlin, Germany: Karl Dietz Verlag. (Cited on page 4).
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (86): 2579–2605. (Cited on pages 23, 187).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 30:5998–6008. Long Beach, CA, USA: Curran Associates Inc. (Cited on pages 20, 25, 30, 33, 38).

- Vijay Viswanathan, Kiril Gashteovski, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large Language Models Enable Few-Shot Clustering. *Transactions of the Association for Computational Linguistics* (Cambridge, MA, USA) 12:321–333. (Cited on page 107).
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, vol. 32. Red Hook, NY, USA: Curran Associates Inc. (Cited on page 122).
- Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang. 2013. SentiView: Sentiment Analysis and Visualization for Internet Popular Topics. *IEEE Transactions on Human-Machine Systems* 43 (6): 620–630. (Cited on page 180).
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv: 2402.05672 [cs.CL]. (Cited on page 115).
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*. Online. (Cited on page 28).
- Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2018. New/s/leak 2.0 – Multilingual Information Extraction and Visualization for Investigative Journalism. In *Social Informatics*, edited by Steffen Staab, Olessia Koltsova, and Dmitry I. Ignatov, 313–322. Cham, Germany: Springer International Publishing. (Cited on page 198).
- Ruth Wodak, Peter Nowak, Johanna Pelikan, Helmut Gruber, Rudolf de Cillia, and Richard Mitten. 1990. Wir sind alle unschuldige Täter: Diskurshistorische Studien zum Nachkriegsantisemitismus. Frankfurt am Main, Germany: Suhrkamp. (Cited on page 2).
- Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024. A survey on neural topic models: methods, applications, and challenges. *Artificial Intelligence Review* 57 (2): 18. (Cited on pages 106 sq.).
- Rixta Wundrak. 2010. Die chinesische Community in Bukarest: Eine rekonstruktive diskursanalytische Fallstudie über Immigration und Transnationalismus. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on page 4).
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On Layer Normalization in the Transformer Architecture. In *Proceedings of the 37th International Conference on Machine Learning*, edited by Hal Daumé III and Aarti Singh, 119:10524–10533. Online: PMLR. (Cited on page 33).
- Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024. Benchmarking Benchmark Leakage in Large Language Models. arXiv: 2404.18824 [cs.CL]. (Cited on page 28).
- Amy Zhang, Bryan Culbertson, and Praveen Paritosh. 2017. Characterizing Online Discussion Using Coarse Discourse Sequences. *Proceedings of the International AAAI Conference on Web and Social Media* 11 (1): 357–366. (Cited on pages 161, 225).

- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*. Vienna, Austria (Online). (Cited on page 90).
- Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can Large Language Models Transform Computational Social Science? *Computational Linguistics* (Cambridge, MA) 50 (1): 237–291. (Cited on page 123).
- Jan Zienkowski. 2017. Reflexivity in the transdisciplinary field of Critical Discourse Studies. *Palgrave Communications* 3 (1): 17007. (Cited on page 57).
- Christian Zimmermann. 2010. Familie als Konfliktfeld im amerikanischen Kulturkampf: Eine Diskursanalyse. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften. (Cited on page 4).