Unsupervised Induction of Domain Dependency Graphs

Extracting, Understanding and Visualizing Domain Knowledge

Dissertation

Eingereicht beim Fachbereich Informatik von
Sarah Kohail

an der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Universität Hamburg

zur Erlangung des akademischen Grades
Doctor rerum naturalium
(Dr. rer. nat.)

Hamburg
2019

Declaration

Eidesstattliche Erklärung/Declaration on Oath

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I hereby declare, on oath, that I have written the present dissertation by my own and have not used other than the acknowledged resources and aids.

Hamburg, den 14.08.2019 Sarah Kohail

Gutachter;
Prof. Dr. Chris Biemann
Prof. Dr. Walid Maalej
Tag der Disputation: 10.12.2019

Acknowledgements

First and foremost, I am grateful to my "Doktorvater" Prof. Dr. Chris Biemann for giving me the opportunity to pursue my doctoral studies under his supervision. This work would not have been possible without your limitless support and encouragement. I learned alot from you and I appreciate all your contributions of ideas, time, guidance, and funding.

I would like to thank the members of the examination committee: Prof. Dr. Walid Maalej, the second evaluator, for his valuable comments and discussion, and Prof. Dr. Mathias Fischer, the committee chair, who gave me additional input during the oral defense.

I would like to express my deep and sincere gratitude to the Deutscher Akademischer Austauschdienst (DAAD), the funding source that allowed me to pursue my studies. Thank you for your generous support.

My time at the LT group was made enjoyable in large part due to the amazingly nice members: Alex, Benjamin, Dirk, Eugen, Fabian, Gregor, Katrin, Meriem, Özge S., Saba, Seid, Steffen, Timo. Not forgetting our floor neighbors from the NATS group. You have been a source of friendship as well as good advice and contribution. My experience around you has been greatly enhanced. I would not forget my former colleague Dr. Martin Riedl. Thanks for contributing to reviewing many of my papers and for teaching me many magical shell scripting tricks. Special thanks to Daniel Brilz, who translated the dissertation's abstract to German and to my hero proofreaders: Seid Muhie Yimam, Saba Anwar, Steffen Remus, and Heba Lahham.

Let me also say 'thank you' to the Department of Linguistics and Philology at Uppsala University for their insightful guidance, discussions and kind hospitality during my research visits.

Finally, but not least, to my family and friends for all their love and support. To my brothers for the humor over the years.

Mama, thank you for being a constant source of inspiration, encouragement, friendship, and support. I love you.

Zusammenfassung

Die Abwesenheit von Struktur in Texten führt bei der Verarbeitung und beim Verstehen durch Maschinen zu besonderen Herausforderungen und die Transformation von Text in eine strukturierte Repräsentation verlangt dringend nach einer Lösung. Klassische auf Bag-of-Words-basierende Vector Space Modelle (BoW-based VSM) betrachten ein Dokument lediglich als Histogramm von Worthäufigkeiten,was die strukturellen und semantischen Eigenschaften des Textinhalts ignoriert. Diese Dissertation erforscht den Nutzen von graphbasierten Textrepräsentationen als eine Alternative zu klassischen Textrepräsentationsmodellen. Im Besonderen stellen wir einen neuen datengetriebenen graphentheoretischen Ansatz zur Textrepräsentation durch Graphen mit dem Namen Domain Dependency Graphs (DDGs) vor. DDGs vereinen die Mächtigkeit der Graphenrepräsentation, als Möglichkeit zum Erhalt der Abhängigkeitsstruktur eines Texts, mit Topic Modeling, als Möglichkeit versteckte thematische Strukturen in Texten aufzudecken. Der Generierungsprozess von DDGs kann folgendermaßen zusammengefasst werden: Unter Verwendung von Topic Modeling extrahieren wir dominante Themen innerhalb eines Dokumentenkorpus. Dann werden Abhängigkeitsstrukturen (Dependenzen) der Dokumente für jedes Thema als kohärenter DDG modelliert, was den inter-thematischen Zusammenhalt mit der strukturellen Komponente des Texts erhält. Später wird auf einer zusätzlichen Ebene ein Vorgehen zur Ausdrucks- und Abhängigkeitsgewichtung angewendet, um die Extraktion von hoch domänenspezifischen Begriffen und Beziehungen sicherzustellen. Unser Ansatz ist komplett unüberwacht und benötigt keine gekennzeichneten Trainingsdaten oder vorheriges Wissen über die Domäne. In dem Bestreben weiteres Verständnis für die extrahierten DDGs zu schaffen, haben wir *DDG*$_{viz}$ entwickelt, welches ein open-source Web-basiertes Visualisierungswerkzeug ist, das einem Nutzer eine einfache Interaktion sowie mit Filtern, Analysieren und Durchsuchen von generierten DDGs durch Anpassung von verschiedenen Parametern und Konfigurationen erlaubt.

Zur Demonstration der Effektivität der generierten DDGs führen wir eine extrinsische Evaluation unter Integration von verschiedenen DDG-basierenden Merkmalen, sowie Graph Mining und Abgleichansätzen durch, um die Leistung bei relevanten sprachtechnologischen

Aufgaben, namentlich Aspect-based Sentiment Analysis (ABSA) und Semantic Textual Similarity (STS), wie folgt zu verbessern:

- Wir erforschen die Effektivität von DDG-basierten Merkmalen wie die von DDGs identifizierten Top-Domänenbegriffe und Aspekte, zusätzlich zu Merkmalen der distributionellen Semantik zur Verbesserung der Leistung von überwachten Modellen verschiedener Aspekt-basierter Teilaufgaben der Stimmungsanalyse (Sentiment Analysis). Wir schlagen zudem ein neuartiges unüberwachtes Graph-Rule Mining Verfahren vor, welches linguistische Strukturinformation zur genauen Identifikation der überzeugendsten Aspekte unterschiedlicher Entitäten (Aspect Identification) sowie Opinion Target Expressions (OTE-sentiment Extraction) aus unstrukturierten nutzergenerierten Rezensionen beinhaltet.

- Wir schlagen eine unüberwachte STS Lösung zum Aufspüren von Ähnlichkeiten zwischen zwei Texten basierend auf DDG-Abgleichen vor. Wir führen ein Verfahren zum approximativen Abgleich von Subgraphen ein, um einen Abhängigkeitsteilgraph im Abhängigkeitsgraphen des Kandidatentexts zu finden, welcher ähnlich zu einem gegebenen Abhängigkeitsgraphen eines Abfragetexts ist. Dies erlaubt das Auftreten von Knotenlücken und Nichtübereinstimmungen, bei denen ein bestimmtes Wort in einem Abhängigkeitsgraphen nicht auf ein Wort im Graphen des Abfragetexts zugeordnet werden kann, als auch Strukturunterschiede in den Graphen. Wir prüfen zudem den Einfluss der Verwendung von ähnlichkeitsbasierten und abdeckungsbasierten DDG-Merkmalen zur Verbesserung der Identifikation und Vorhersage von überwachten STS Modellen.

Experimente auf unterschiedlichen Benchmark-Datensätzen für unterschiedliche Teilaufgaben zeigten, dass die Integration von DDG-basierten Merkmalen zu besseren Ergebnissen im Vergleich zu zu aktuellen Ansätzen führt.

Abstract

The unstructured nature of text documents makes the task of processing and understanding it by machines very challenging, and transforming it into structured representation has become a pressing. Classical Bag-of-Words-based Vector Space Model (BoW-based VSM) represents documents as independent terms and only considers the document as a histogram of word occurrences, ignoring structural and semantic aspects of textual contents. This dissertation explores the utility of graph-based text representations as an alternative to classical text representation models. Specifically, we propose a new data-driven graph-theoretic approach to representing text by means of graphs, called Domain Dependency Graphs (DDGs). DDGs integrate the power of graph representation, as a way to preserve the dependency structure of a text, with topic modeling, as a way to uncover the hidden topical semantic structure of a text. In summary, DDGs generation process goes as follows: using topic modeling, we extract dominant topics from a corpus of documents. Then, source-side dependency structures of documents per topic are modeled as one coherent DDG, which maintains the inter-topic cohesiveness together with the structural aspect of a text. Later, an extra level of term and dependency weighting approach is applied to ensure the extraction of highly domain-specific words and relations. Our approach is completely unsupervised and needs no labeled training data or previous knowledge about the domains. In an effort to provide further understanding of the extracted DDGs, we develop DDG_{viz}, an interactive open-source web-based visualization tool, which enables users to filter, analyze, search and easily interact with generated DDGs by adjusting various parameters and configurations.

To demonstrate the effectiveness of the generated DDGs, we perform extrinsic evaluation by integrating several DDGs-based features, and graph mining and alignment approaches to improving the performance of relevant Natural Language Processing (NLP) tasks, namely Aspect-based Sentiment Analysis (ABSA) and Semantic Textual Similarity (STS), as follows:

- We explore the effectiveness of DDGs-based features, like DDGs top domain words and DDGs identified aspects, in addition to distributional semantics features, for improving the performance of supervised models for different aspect-based sentiment analysis subtasks. We also propose a novel unsupervised graph-rule mining approach,

which incorporates high level linguistic structural information to accurately identify the most compelling aspects of different entities (aspect identification) and extract opinion related expressions (OTE-sentiment extraction) from unstructured user-generated reviews.

- We provide an unsupervised STS solution to finding similarities between two texts based on DDGs alignment. We introduce an approximate sub-graph alignment approach to find a dependency sub-graph in the candidate text dependency graph that is similar to a given query text dependency graph, allowing for node gaps and mismatches, where a certain word in one dependency graph cannot be mapped to any word in the query text graph, as well as graph structural differences. We also examine the impact of using DDGs similarity-based and coverage-based features to improve the identification and prediction of STS supervised models.

Experiments on different benchmark datasets for different subtasks revealed that incorporating DDGs-based features show superior results compared to state-of-the-art approaches.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Unstructured text makes up the highest proportion of data on the Web. The evolution of such text has established itself as a rich source for obtaining knowledge and information about many topics and domains. Text stores information about people and events (e.g., Nelson Mandela was born on July 18, 1918), science facts (e.g., Liquid nitrogen boils at 77 kelvin), common knowledge (e.g., blood is red), political views, debates and, individuals experiences, and opinions.

Valuable as they are, dealing with unstructured text usually involves two main challenges: **First**, unstructured text lacks metadata (i.e., structured information or annotations), providing no standard means to facilitate further text analysis or improve the understanding of knowledge in a specific domain. **Second**, due to its size and heterogeneity (i.e., it covers a wide range of topics), processing unstructured text is complex and time-consuming. It often includes redundant information and ungrammatical language. The two previous challenges have received increased attention from researchers in the fields of Natural Language Processing (NLP), Machine Learning (ML), and Text Mining (TM). Since the 1950s, these research fields have put their best effort aiming to imitate human intelligence in understanding and analyzing natural language by automating many text analysis and processing tasks, including but not limited to, information extraction, sentiment analysis, automatic question answering, and text comprehension and summarization. Most of these tasks are based on text representation models, which transform textual contents into a proper representation to make it understandable to learning algorithms. Text representation models also determine the relevant features to be used for representing a text object (here, document) and its meaning. The most conventional and straightforward representation is the Bag-of-Words (BoW) model [40]. The classical BoW model represents documents as independent terms and only considers the document as a histogram of word occurrences, disregarding first-order structural properties of textual contents. Practically, a text unit is a coherent sequence of terms where adjacent terms have strong relationships between them [67]. These relationships contribute to the overall meaning of the text and maintain the inter-sentence and intra-sentence cohesive structure. Hence, deducing the meaning of a text is a hierarchical process, such as, the meaning of coarser-grained linguistic units, such as a corpora or a document, depends on the meaning of its smaller finer-grained units, such as words and sentences, the syntactic relationships that connect them, as well as how they are ordered. Building a text representation model that can preserve such relationships and maintain the structural representation of the context, will accordingly improve the performance of any learning algorithm build on top. An intuitive, widely-used representation is the graph representation.

Text graph has shown a potential ability to hold linguistic information (vertices) and relationships (edges) between them (see the book by Radev and Mihalcea [137]). In this way, it transforms the non-linear, unstructured nature of text into a mathematically structured and tractable representation. This representation is powerful in a way that it can integrate several aspects of text (i.e., statistical, grammatical, topological, etc.), which in fact constitutes the solution we are looking for to address the limitations of the BoW model. A graph representation of a text brings many ready-made graph analysis solutions to text mining. There are already many existing algorithms and tools available to efficiently search, traverse, analyze, process, and deal with different problems related to the graph as a data structure. The idea of using graphs to represent text is not new to NLP, however, until recent years, graph representations have been mostly studied as a text visualization or language understanding approach, rather than a general-purpose modeling structure, to aid higher-level NLP applications. Nowadays, machines have the capacity and the processing ability to perform tasks that humans cannot, such as searching or classifying millions of documents in a fraction of a second. This advent of high-performance computing has paved the way for researchers to develop new graph-based methods alternatives to classical text representation models.

Earlier works in NLP followed the "top-down" paradigm trying to fit new information into a previously generated high-level structured representations [42]. In this case, understanding a text about a certain domain pre-requires adequate knowledge about this domain. In particular, domain entities and the relationships between them. For example, in the domain of football, possible entities are (player, fan, coach, referee, ball, match, stadium), the relationships could be (play, train, send off, injured, score, watch). Researchers who adopted the "top-down" paradigm have argued that a predefined structured domain knowledge provides learning algorithms with rich information about many aspects of meaning. However, more information requires more text annotation and hand-construction. The task of annotating text is a time-consuming process and requires many domain experts to describe the important semantics of each domain accurately, especially for text from critical domains like medical text. One way to reduce the demands of human judgment is adopting a "bottom-up" paradigm by following a data-driven approach. Data-driven approaches rely solely on the data at hand. They leverage statistical analysis and pattern/association mining approaches to learn knowledge directly from free text. For example, statistical topic models provide a general data-driven framework to automate the discovery of the hidden semantic structures of abstract topics from large collections of text documents.

In response to the challenges mentioned earlier, this dissertation proposes a new unsupervised data-driven graph-theoretic approach to representing text by-means-of graphs, called Domain Dependency Graphs (DDGs). DDGs integrate the power of dependencies, as a way to preserve the structure of the text, with topic modeling, as a way to uncover the underlying domain semantic structure. A single DDG aggregates individual dependency relations from a collection of documents belonging to the same domain. Term weighting scheme is applied to a DDG in order to derive and retain highly weighted domain-specific words and relations. We later explore the utility of DDGs by integrating several DDGs-based features, which measure various aspects of the DDGs, and by extension of the text represented by the DDGs, to improve the performance of different NLP tasks. More details about DDGs will be given in the forthcoming subsections.

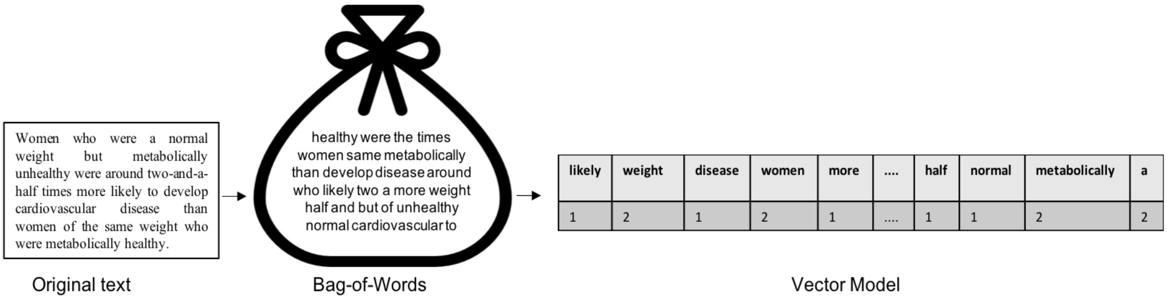
The remaining of this chapter is organized as follows: in Section 1.1, we provide some background on the Bag-of-Words-based Vector Space Model (VSM) and discuss its points of weakness. We review some alternative graph-based models in Section 1.2. Section 1.3 summarizes our research contribution, Section 1.4 outlines the essential content of this thesis, and Section 1.5 lists the publications in which this thesis are based on.

1.1 Bag-of-Words-based Vector Space Model

The Vector Space Model (VSM) represents each text document as a vector of term weights. For a collection of N text documents D, $D = \{d_1, d_2, ..., d_N\}$, each document d_i is transformed into a vector structure $\vec{d_i} = (w_{i1}, w_{i2}, ..., w_{in})$, where the value of a dimension w_{ij} corresponds to the weight of a term t_j in a document d_i, and t_j corresponds to a single term from the whole collection vocabulary, $T = \{t_1, t_2, ..., t_n\}$. The weight reflects the importance of each term inside a document, and how much this term contributes to solving a particular task. There are many ways to define the notion of "terms" and "weights".

A "term" could represent a single word, also called a "unigram", or contiguous subsequences of n words, called "n-grams", where n refers to the size of each subsequence. So, an n-gram of size two is called "bigram" and size three n-gram is called "trigram", etc. The standard VSM using n-grams as terms is often called Bag-of-Words (BoW) model [40], see Figure 1.1.

The most often used weighting schemes are binary existence, raw term frequency (Tf) [94], and term frequency-inverse document frequency (Tf-Idf) [155]. Binary existence weighting measure encodes the absence or presence of a term in a document as a binary value of 0 or 1, respectively. Tf states the number of occurrences a term has in a document (local) without measuring the importance of that term within the whole collection of text documents (global). A more advanced term weighting scheme is Tf-Idf. It combines the local Tf measure with another global weighting measure called Idf. This measure gives higher weights to terms that frequently occur in a specific document (locally) but not in most other documents (globally). As a part of our approach, Tf-Idf weighting scheme will be defined and explained at length in Chapter 2.



Women who were a normal weight but metabolically unhealthy were around two-and-a-half times more likely to develop cardiovascular disease than women of the same weight who were metabolically healthy.

healthy were the times women same metabolically than develop disease around who likely two a more weight half and but of unhealthy normal cardiovascular to

likely	weight	disease	women	more	....	half	normal	metabolically	a
1	2	1	2	1	....	1	1	2	2

Fig. 1.1 An illustration of how text documents are represented using Bag-of-Words and Vector Space Model. In this example, a document is represented as a weighted vector of word occurrences (Tf). The order of the words and their positions in the original document are ignored.

The idea behind VSM is to have a shared feature vector between all the documents in a collection, in particular when considering the corpus as a document-term matrix, where each row is a document, and each column is a term (i.e., feature) found in at least one of the documents. From a machine learning perspective, a common representation is required as standard input to most existing algorithms and similarity measures, as such a shared representation facilitates a simple way to induce features for judging the similarity between each pair of documents in the documents collection.

For example, given two documents d_1 and d_2, the similarity between them is measured using cosine similarity, which calculates the cosine of the angle between their weighted vector representations $\vec{d_1}$ and $\vec{d_2}$ as follows:

$$cos(\vec{d_1},\vec{d_2}) = \frac{\sum_{i=1}^{n} w_{1i}w_{2i}}{\sqrt{\sum_{i=1}^{n} {w_{1i}}^2}\sqrt{\sum_{i=1}^{n} {w_{2i}}^2}} \tag{1.1}$$

Vectors are (length-) normalized by dividing by the L_2 norm $\sqrt{\sum_{i=1}^{n} {w_{1i}}^2}$ and $\sqrt{\sum_{i=1}^{n} {w_{2i}}^2}$ of w_{1i} and w_{2i} respectively. Cosine similarity result from Equation 1.1 is bounded between 0 (completely dissimilar) and 1 (identical).

Although traditional models like the BoW-based VSM have shown effectiveness in many NLP applications like information retrieval and text classification, they still have several limitations regarding their ability to capture the syntactical structure and the semantic information of text contents. We show some cases where shallow lexical surface features, like words, are not enough.

Example 1: Negation: Consider the task of measuring the similarity between the following two sentences:

"The cat is sitting on the mat" **vs.** *"The cat is not sitting on the mat"*

Although these sentences are not semantically equivalent, measuring cosine similarity between their corresponding term-frequency weighted vectors, the pair receives a high similarity score of 0.93, and would reach 1.0 when filtering stopwords.

Example 2: Grammatical relations: Consider the following two sentences:

"Smith gave the book to Olivia" **vs.** *"Olivia gave the book to Smith"*

Both sentences share the same words. In fact, the cosine similarity between their weighted vectors is equal to 1.0. However, the two sentences carry different meanings. As both sentences are represented as an isolated set of word occurrences, the cosine similarity is neither able to capture semantic nor structural distinctions.

Example 3: Long-distance dependencies: Consider the task of supervised document-level sentiment polarity classification to judge whether the following restaurant review expresses a positive, negative or neutral opinion:

"Service is fantastic. The chicken recipe which reviewers said very delicious is just below average."

Given that the review contains the words "delicious" and "fantastic", comparing this review to a set of reviews, pre-labeled with sentiment polarities, the review is very likely to be highly

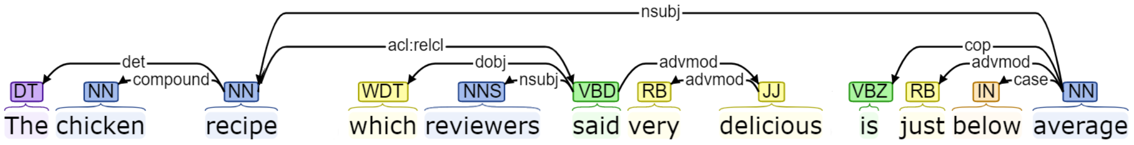


Fig. 1.2 Syntactic dependency parsing tree-view of the sentence *"The chicken recipe which reviewers said very delicious is just below average"*. The correct sentiment *"below average"* have a long-distance dependency relation to the aspect *"chicken recipe"*.

correlated with positive reviews, and therefore be assigned the "positive" polarity. However, classifying the overall sentiment as "positive" would override the fact that "chicken recipe" is "below average". Thus, it is more accurate to consider finer-grained sentiment analysis approaches.

Aspect-based Sentiment Analysis (ABSA) performs a finer-grained analysis by identifying specific sentiments toward different aspects of an entity. ABSA mainly involves two sub-tasks: aspect-polarity expression identification and sentiment polarity classification. Given the previous review, an ABSA method should be able to identify the following aspect-polarity pairs: *{SERVICE#FANTASTIC}* and *{CHICKEN_RECIPE#BELOW_AVERAGE}*, and classify each pair to the correct sentiment polarity: *{SERVICE#FANTASTIC#POSITIVE}*, *{CHICKEN_RECIPE#BELOW_AVERAGE#NEGATIVE}*. A straightforward aspect-polarity identification approach is to use a predefined lexicons/dictionaries, which include highly-frequent aspects across domain-specific reviews (e.g., for restaurant domain, aspects would be décor, food, service, etc.), and find them later in new reviews. The sentiment toward each aspect is then determined by its nearest adjacent adjective opinion word. This approach performs well when applied for the first sentence "Service is fantastic", yet it fails to determine the correct aspect-sentiment pair for the second sentence. Since linguistic structure, in this case long-distance dependency, is disregarded, the aspect "chicken recipe" will be assigned to the wrong sentiment "delicious", see Figure 1.2. We discuss ABSA in more detail in Chapter 4.

The previous examples demonstrate that conventional surface-based features, as represented in a BoW model, cannot provide sufficiently discriminative semantic features, nor convey any information about the syntactic relationship between the words. Thus, we find it necessary to develop a new text representation model that can maintain the cohesiveness aspect of a text, and better capture text semantics within a specific domain.

1.2 Graph-Based Models

To overcome the term independence assumption imposed by the classical BoW-based VSM model, encoding text as graph has been explored as an alternative. A text can appropriately be represented as a graph with a set of vertices (nodes), which correspond to linguistic units like paragraph [14], sentences [52, 104], words [17, 34, 148, 23, 140] or characters [45, 61], and a set of edges on the basis of meaningful statistical [72] (e.g. co-occurrences), syntactic [56] (e.g. grammatical relations), or semantic [87] (e.g. synonym, meronymy) relations between these node. Researchers who studied graph-based models found that they can reach better results than VSM for many NLP tasks [137], yet, these models still need to be improved, especially to increase their ability to capture the structural and semantic aspects of a text.

Schenker et al. [148] proposed a set of pure graph-based methods to represent web documents as graphs, so as to capture the word-ordering in the text. A directed graph is constructed from a text document by representing each term as a unique node, and a directed edge is created between two nodes only if the terms representing these nodes are adjacent in the original text, where the edge direction indicates the terms ordering in the text. They also introduced several distance measures to compute the similarity between two graphs. Evaluation on text classification task showed a significant improvement in classification accuracy over a BoW-based VSM model. Despite the improved results, the connections in the graphs still lack the context needed to understand why a certain connection makes sense. As a way to overcome this issue, Schenker et al. [148] have extended their approach with an $n-Distance$ parameter. Instead of connecting strictly adjacent terms, a term is connected to up to n succeeding terms ahead. An edge label corresponds to the window distance between two terms, or in another version of the representation, the total number of their co-occurrences in the text. Despite the better ability to encode some kind of structural aspects of the text (i.e., words proximity, word location), the resulted graph representation is still difficult to be processed by or adapted to most ML algorithms, since the whole method is entirely carried out based on graph theory analysis. Additionally, due to the extremely high number of edges, comparing graphs using graph similarity and matching measures, such as maximum common sub-graph, is computationally expensive.

An improvement to the pure graph representation is hybrid representation [97]. Hybrid representation is built upon both vector space and graph models, thus, merging the capabilities and overcoming the limitations of each. In summary, the idea of hybrid representation is as follows: first, all documents in a given corpus are represented as graphs. Second, subgraphs that occur frequently are retrieved, weighted and used as terms in VSM. Hybrid representations have shown better performance than pure graph-based methods [98, 39], yet, representing text by means of word order has limited ability to capture long-distance dependencies, especially when dealing with languages that allow flexible word ordering. Relying on word order is also far from universal, especially when assessing the similarity of differently phrased sentences. There are some emerging approaches to using more meaningful and universal information to represent text than just words and simple word order, like dependency structure and semantics.

Dependency grammars represent sentence structures as a set of dependency relationships. It can be considered as an intermediate layer between surface syntax and semantics. A single dependency relation is a directed relationship between two terms, a head and a dependent [122]. These individual dependency relations interconnect together to form a tree connecting all the words in a sentence. To create dependency graph from a text document, a text is parsed using a dependency parser, and a graph is constructed by aggregating dependency trees for all sentences [126]. Two major valuable advantages of dependency structures are its ability to bring long-distance dependency between words and its close correspondence to meaning. Dependency graphs however can be very complex and challenging to work with, especially for large corpus. To avoid such complexity, graph-based ranking approaches are applied to estimate and select the top import nodes and edges. Simple graphs express less, but are easier to work with. Variety of natural language processing tasks have benefited from this simplification approach, such as automated keyphrases/keywords extraction, to extractive text summarization and word sense disambiguation [169, 101, 118]. A widely successful graph-based ranking algorithm in the area of NLP is TextRank [104].

All the graph models we discussed above have the capability of capturing text structural information, but not explicit semantic relations between words. Conventional methods to capture the semantic relations between words are Thesaurus Graph (TGs) and Conceptual Graph (CGs) [156]. TGs are constructed based on dictionaries or thesaurus, such as, the nodes denote terms, and edges between them denote sense relations, e.g. synonymy or antonymy. CGs, on the other hand, are constructed directly from text documents. In CGs, there are two types of nodes, *Concepts* and *Relations*. A *Relation* node indicates the semantic role of the incident *Concepts*. After preprocessing raw text and mapping disambiguating nouns to Wordnet concepts, VerbNet is used to find the semantic roles in a sentence. In a bigger picture, these *Concepts* and *Relations* collectively make up a CG. CGs contain rich semantic information, so they are more interpretable than word graphs.

Based on the previous discussion of graph-based models, we conjecture that an ideal graph-based representation should abstract over surface word order, mirror semantic relationships as closely as possible, and incorporate word-based information in addition to syntactic analysis.

1.3 Contribution of this Dissertation

This work is mainly motivated by a need for richer representations of text that provides a better basis for text analysis and understanding. In this dissertation, we propose a new graph-based approach to representing text by-means-of Domain Dependency Graphs (DDGs). A single DDG helps to advance the understanding of a specific domain from mixed-domain documents by aggregating individual dependency relations between domain-specific content words for a single topic. The obtained DDGs encode both structural and shallow semantic information extracted from dependency parser and topic model, respectively. We summarize the contributions and features of this research in the following points:

- **Incorporate structure features:** Based on further statistical analysis of derived DDGs, we define a set of structural and semantic features and evaluate the utility of these features to train various models for different Natural Language Processing (NLP) tasks, namely semantic similarity [85, 86], sentiment analysis [84, 89], and question answering [64, 117]. We show that the integration of our DDG features yields superior results than those captured by traditional systems with a robust baseline feature set.

- **Facilitate professional search, knowledge extraction and summarization in specific domains:** We aim to derive a comprehensive domain-specific representation – one that is rich enough to allow us to perform other specialized analysis on top of it. A single DDG provides a representation tailored toward a specific domain that comprises all the relevant concepts of that domain in a structured way, hence, easy to customize to particular domain-specific applications, like, automatic indexing systems, professional search engines and personalized recommendation systems. It would also facilitate drawing inferences and increase the precision of domain-specific manual rules by domain experts.

- **Promote better interactive visualization**: The ability of the users to obtain a better understanding of extracted DDGs is further assisted by an interactive web-based visualization tool called DDG_{viz}. DDG_{viz} with drill-down possibilities helps users to visualize extracted DDGs at multiple levels of granularity. Hence, users can gain more in-depth insight into the main extracted DDGs, as well as their detailed internal structure. Users can select relevant sub-graphs and perform different operations on these sub-graphs like searching, filtering and summarization.

- **Unsupervised:** Our method is entirely unsupervised and does not require labeled training data, manual annotation or previous domain knowledge. It builds the graphs from scratch without the need for any external knowledge base. This makes it easily applicable to any new corpus across many genres and languages, without the need for modifying the graphs construction procedure. Furthermore, the extraction of DDGs is based on topic modeling and syntactic analysis, which are well-established, and do not require complicated processing steps.

1.4 Thesis Outline

The work presented in this thesis has two parts. The first part (Chapters 2 and 3) introduces and describes the construction and visualization of Domain Dependency Graphs (DDGs), while the second part (Chapters 3-5) investigates the utility of DDGs features over multiple natural language processing tasks, namely Aspect-based Sentiment Analysis (ABSA) and Semantic Textual Similarity (STS). In each chapter, a review of relevant literature is conducted separately. In detail, the rest of this dissertation is organized as follows:

Chapter 2 defines the main concept of DDGs, and explains in detail how it is constructed based on syntactic structure extracted from dependency parsing output. It also explains how to use additional structural characteristics to filter DDGs beyond textual contents in order to select the most representative documents for each domain graph.

Chapter 3 presents DDG_{viz}, an interactive web-based tool that allows the user to visualize and easily interact with DDGs. Furthermore, DDG_{viz} enables users to filter, analyze and search generated DDGs by adjusting various parameters and configurations.

Chapter 4 describes how DDGs-based features are applied to solve the task of ABSA and how the obtained graphs are used to improve the overall understanding of opinion patterns and distinguish the most effective aspects for different product categories.

Chapter 5 studies the impact of using DDGs similarity and coverage features in improving STS estimation. We show how by integrating DDGs-based features, most relevant documents to a given query text can be more accurately ranked and scored. Additionally, we introduce an approximate dependency sub-graph alignment approach allowing node gaps and mismatch, where a certain word in one dependency graph cannot be mapped to any word in the other graph.

Chapter 6 concludes the dissertation by summarizing the main contributions and discussing possible future directions.

1.5 Published Work

The content of this thesis is mainly based on the publication listed bellow:

1. Kohail, S. (2015). Unsupervised Topic-specific Domain Dependency Graphs for Aspect Identification in Sentiment Analysis. In *Proceedings of the Student Research Workshop associated with RANLP*, pages 16–23, Hissar, Bulgaria

2. Kumar, A., Kohail, S., Ekbal, A., and Biemann, C. (2015). IIT-TUDA: System for Sentiment Analysis in Indian Languages using Lexical Acquisition. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 684–693, Hyderabad, India

3. Kumar, A., Kohail, S., Kumar, A., Ekbal, A., and Biemann, C. (2016). IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1129–1135, San Diego, California

4. Kohail, S. and Biemann, C. (2017). Matching, Re-ranking and Scoring: Learning Textual Similarity by Incorporating Dependency Graph Alignment and Coverage Features. In *18th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'17, pages 377–390, Budapest, Hungary

5. Kohail, S., Salama, A. R., and Biemann, C. (2017). STS-UHH at SemEval-2017 Task 1: Scoring Semantic Textual Similarity Using Supervised and Unsupervised Ensemble. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 175–179, Vancouver, Canada

6. Nandi, T., Biemann, C., Yimam, S. M., Gupta, D., Kohail, S., Ekbal, A., and Bhattacharyya, P. (2017). IIT-UHH at SemEval-2017 Task 3: Exploring Multiple Features for Community Question Answering and Implicit Dialogue Identification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 90–97, Vancouver, Canada

7. Akhtar, M. S., Kohail, S., Kumar, A., Ekbal, A., and Biemann, C. (2017). Feature Selection Using Multi-objective Optimization for Aspect Based Sentiment Analysis. In Frasincar, F., Ittoo, A., Nguyen, L. M., and Métais, E., editors, *Natural Language Processing and Information Systems*, NLDB'17, pages 15–27, Liège, Belgium

8. Gupta, D., Kohail, S., and Bhattacharyya, P. (2018). Combining Graph-based Dependency Features with Convolutional Neural Network for Answer Triggering. In *The 19th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'18, Hanoi, Vietnam

First Author Contribution (Publication 1 & 4): contributed fully to the design and implementation of the methodology, data collection, experimentation, result analysis and evaluation, and writing the manuscript.

Co-first Author Contribution (Publication 3, 5, 7 & 8): contributed equally to the study as the first author. My main contribution was to preprocess the text, build DDGs, and evaluate the results after incorporating DDGs-based features. I also contributed significantly to writing and revising the final research paper.

Co-author (Publication 6): provided minor contribution to this research by preparing few experiments for the purpose of comparison.

Supervision and Writing (Publication 2): mentored the work progress, acquired background resources, provided insights to refining and improving the experiments, and majorly involved in drafting and reviewing the manuscript.

Chapter 2

Domain Dependency Graphs

2.1 Introduction

Natural Language Processing (NLP) helps Artificial Intelligent (AI) agents to understand the language nuances and contexts expressed by humans. NLP applications such as search engines, machine translators, or text summarization are developed to fulfill complex tasks for the end-users. In order to perform these tasks, different layers of linguistic processing and annotation steps have to be applied. Raw text data has to be transformed into a proper text representation, on top of which further task-specific computations by machines can be performed. The most widely adopted text representation is Bag-of-Words (BoW). The BoW model represents a text document as a vector of word occurrences across a vocabulary. Despite its simplicity and flexibility, the BoW model disregards the two most important aspects of a meaningful text, structure and semantic. Language entities such as words, phrases, and complete sentences, originating form a meaningful text, are connected through various grammatical and semantic relationships. These relationships reflect the cohesion of the text and support the structure and text unity [67]. As an alternative to the classical BoW representation, graph-based representations have shown a potential ability to hold and understand these relationships [103].

This dissertation proposes an unsupervised generic method to model a multi-domain document collection by means of domain dependency graphs (DDGs). DDGs represent textual documents as statistical graphs whose vertices correspond to unique domain-specific content words and whose edges correspond to the dependency relations between the words. DDGs can encode structural (through dependency parsing) and shallow semantic information (through topic modeling), which are both necessary for future text processing tasks. Our method is completely unsupervised and follows the Structure Discovery paradigm [20], meaning it requires no labeled training data or previous knowledge about the domains.

The remainder of this chapter is organized as follows: Section 2.2 describes the proposed methodology for generating DDGs. Section 2.3 concludes the chapter by a final remark regarding the evaluation of our methodology and the generated DDGs.

2.2 Methodology

The purpose of this work is to advance the understanding of a specific domain from mixed-domain documents by building compact directed DDGs. A DDG aggregates individual dependency relations between domain-specific content words of a single topic. It gives a good summarization of a certain domain, and facilitate information and relation extraction.

The methodology of generating DDGs for a given text corpus is summarized as follows: after preprocessing the text, we apply LDA topic modeling to discover underlying topics in a collection of textual data, and calculate a probabilistic topic distribution to select the most related phrases to each topic. POS tagging and dependency parsing are used to select essential domain-specific phrases and content words. Finally, we aggregate DDG per topic from the dependency parses, and use Tf-Idf and word frequency measures to weigh the graph nodes and edges.

2.2.1 Dataset Preprocessing and Topic Modeling

Preprocessing

Prior to training topic models, it is common to perform a text preprocessing step, which includes text normalization and contentless words removal [28]. Preprocessing eliminates high-probability words that can slow the inference process, improves the resulted topic model coherence, and produces more interpretable high-quality topics [36]. We apply a standard text preprocessing pipeline as follows: (1) word tokenization based on punctuation and whitespace; (2) text normalization, confined only to lower-casing; (3) stop words removal based on pre-defined stopwords lists, and low- and high-frequent words as such are assumed to be either non-distinctive words, typos or misspelling; (4) filtering out very short documents with less than 4 words.

LDA

Statistical topic models enable the exploration of large document collections by identifying co-occurring words that can capture thematic patterns. A popular and successful statistical topic modeling technique is LDA (Latent Dirichlet allocation) [24]. LDA is a Bayesian probabilistic topic modeling approach that treats a document as a multinomial distribution of topics, each topic is a multinomial distribution of words [25]. LDA compresses multiple words into a latent dimension to uncover the underlying topics behind large text collections and reduces the dimensionality of the feature space. LDA is completely unsupervised and does not require labeled training data, however, the user has to provide the number of latent topics to be learned n. Determining the value of n usually involves a trade-off between topic quality and resolution [161]. A large value of n generates small noisy topics due to insufficient data, while a small value produces generic topics that do not have sufficient details for in-depth analysis. Several methods have been proposed to automatically estimating the optimal value of n, however, recent studies enumerated the drawbacks of these methods, and suggest that the number of topics depends on the data and task, and therefore should be determined by experimentation [12]. For the experiments of this dissertation, we rely on the latter to define the number of topics. When using a commonly-used corpus, we benefit from the findings of researchers who used the same corpus for topic modeling to define the optimal number of topics.

While running, LDA tries to find the most probable topic structure that best explains the observed posterior probability distribution in a given training corpus, also called the inference process. Although calculating the probability of any particular topic structure is simple, however, the number of possible topic structures is exponentially large. A solution to this problem is approximating the posterior distribution by drawing many random samples. The most used sampling algorithm is Gibbs sampling [59, 48]. For the experiments described in this dissertation, we use the GibbsLDA implementation provided by Phan and Nguyen [130]. After an LDA model is estimated using a sufficiently large unlabeled background corpus, we use it to infer the topic distribution of new unseen data. Each word is assigned to its most probable topic, according to LDA. Some frequent general words may end up equally assigned to different topics. To stabilize the topic assignment, we run 100 inferences and choose for each word in a context, the topic that was assigned to that word in the most inferences overall. A coarser text unit (i.e., sentence, paragraph, etc.) is assigned to a certain topic, in which the majority of its words, excluding stopwords, belong to that topic.

From this point on, we perceive all texts belonging to one topic as one document. The terms "domain" and "topic" are used interchangeably throughout the text.

2.2.2 Topically-pure Content Selection

Back to the previous step, each text document is classified into one topic, to which its majority of words were assigned. However, majority may not be decisive, especially in cases where the majority is not absolute, e.g., imagine a the following topic probability distribution of a document over 8 topics, (t_1:0.20, t_2:0.05, t_3:0.15, t_4:0.10, t_5:0.15, t_6:0.15, t_7:0.15, t_8:0.05). Note that the document is almost evenly distributed along major topics, which makes it difficult to decide on a single topic. We use the vocabulary distribution of the documents produced by LDA to find per topic a collection of topically pure documents. We retain only documents that have a single dominating topic, which covers at least 60% of the document[1]. This step is significant to eliminate documents that contain too much noise or are too general to characterize a specific topic.

Proceeding with selected domain-specific texts per topic, we perform sentence segmentation followed by Part-of-Speech (POS) tagging and dependency parsing. Sentence segmentation is a natural choice, since dependency parsing works within a sentence scope. Dependencies between words in a sentence tend to be local and the distance between syntactically linked words in a sentence decays exponentially.

The output of this step is important for generating syntactic features that will be used later to construct and filter DDGs.

2.2.3 Dependency Parsing

With the advent of high-performance computers, deep sentence-level linguistic analysis, like dependency parsing, for large scale text corpora has become practical. The task of dependency parsing is to analyze a sentence in terms of directed links (dependencies) expressing the grammatical relations between its words. Dependency parsing is widely adopted and does not require complicated text processing. To extract the dependency structure of a sentence, the sentence is parsed by a dependency parser, which is based on the theoretical foundations of dependency grammar. Normally the dependency structure forms a tree where every word in a sentence, except for the root word, is dependent upon another word and each of these dependencies has a type. This structure abstracts word order away, meaning that multiple sentences can map to the same dependency tree.

For each topic i, $i \in \{1,...,n\}$, we generate a document d_i, which includes all the directed typed dependency relations resulted from parsing topically-pure texts belonging to topic i.

2.2.4 Filtering Non-Content Words

Since non-content words do not contribute as much information about a specific topic, we retain relations where at least one word in a relation is a content word, i.e., common and proper nouns, adjectives, verbs and adverbs. From this step onward, the work followed is done completely on collapsed dependency documents.

[1] Threshold was determined in preliminary experiments

2.2.5 Term Weighting

Term weighting is a commonly used procedure in NLP. It assesses the importance of a word to a document in a collection of documents or a corpus, or how well it contributes to solving a particular task. One of the best and advanced known term weighting schemes is Tf-Idf [155]. The core idea behind Tf-Idf is: given a document d_i, a word w is more relevant as a keyword for d_i if it appears many times in d_i and very few times or none in other documents in a corpus D. There exist several other variants of the Tf-Idf formula, however, the most common is expressed by the following equation:

$$Tf\text{-}Idf(w, d_i, D) = Tf(w, d_i) \times Idf(w, D) \tag{2.1}$$

where term frequency Tf is the number of times that word w occurs in document d_i, and inverse document frequency Idf is usually calculated by taking the logarithmic of the total number of documents N in a corpus D, which is also the number of topics n in our case, divided to the number of documents containing the term w, can be more formally written as: $\sum\limits_{d_i \in D} [w \in d_i]$.

Using Tf-Idf as a weighting scheme, unique terms per topic are weighted highest, while more common terms and stop words, are given lower weights. The strength of Tf-Idf is it does not require a dictionary lookup, thus making it language independent.

We calculate Tf-Idf in three levels of granularity:

1. Word level: for each word w_{ij} in d_i, we calculated Tf-Idf using Equation 2.1.

2. Pair level: for each pair of words w_{ij} and w_{ik} in d_i, occurred together in a typed dependency relation R_{ijk}, we calculated Tf-Idf using the following equation:

$$Tf\text{-}Idf(w_{ij}w_{ik}, d_i, D) = Tf(w_{ij}w_{ik}, d_i) \times Idf(w_{ij}w_{ik}, D) \tag{2.2}$$

   w_{ij} and w_{ik} represents the j^{th} and k^{th} words in document i. Order of words w_{ij} and w_{ik} within the relation is not considered at this level.

3. Relation level: for each typed dependency relation R_{ijk} in d_i between two words w_{ij} and w_{ik}, we calculate Tf-Idf using the following equation:

$$Tf\text{-}Idf(R_{ijk}w_{ij}w_{ik}, d_i, D) = Tf(R_{ijk}w_{ij}w_{ik}, d_i) \times Idf(R_{ijk}w_{ij}w_{ik}, D) \tag{2.3}$$

   At this level, the order of R_{ijk}, w_{ij} and w_{ik} matters.

2.2.6 Domain Dependency Graphs (DDGs)

A DDG is a graph with labeled nodes and labeled edges. For each document d_i, DDG_i is constructed by aggregating individual dependency relations between domain-specific content words. DDG_i=$\{V_i,E_i\}$, where nodes represent words, that is V_i=$\{w_{ij} \mid w_{ij} \in d_i$, *Tf-Idf*$(w_{ij},d_i,D) \geq \alpha_1$, *Tf*$(w_{ij}) \geq \alpha_2\}$, and edges E_i connect words by the means of dependency relations. E_i =$\{(w_{ij},w_{ik}) \mid w_{ij},w_{ik} \in d_i$, *Tf-Idf*$(w_{ij}\ w_{ik},d_i,D) \geq \beta_1$, *Tf*$(w_{ij}\ w_{ik}) \geq \beta_2$, *Tf-Idf*$(R_{ijk}\ w_{ij}\ w_{ik},d_i,D) \geq \lambda_1$, *Tf*$(R_{ijk}\ w_{ij}\ w_{ik}) \geq \lambda_2\ \}$.
Thresholds, α_1, α_2, β_1, β_2, λ_1, λ_2 are defined by the user, each node is labeled with the term it represents, and edges are labeled by the type of dependency relation between words. Edge directions are optional. In some experiments, we only weigh the resulted DDG using α_1, β_1 and λ_1. Using Tf-Idf to weighting words and relations, have a potential ability to highlight a large set of domain-specific words and relations as will be demonstrated later in Chapter 4. Each topic is assigned a label based on the word that has the highest word-level Tf-Idf in its corresponding DDG.

2.3 Final Remark

The novelty of our work is not so much in the individual components of the previous methodology itself (LDA, parsing, Tf-Idf weighting), but rather in the way in which these components are orchestrated to generate DDGs, and the utility of DDGs for different NLP applications. We choose to evaluate our DDGs representation on several extrinsic tasks. During the evaluation experiments, DDG can be seen as a method of feature selection (i.e., aggregate domain-specific content words), a possible back-end model for a visualization tool that can improve the understanding of a large unstructured corpus, a structured representation on which rule mining and pattern discovery can be performed (i.e., utilizing dependency information), or a rich textual representation from which many features can be extracted (i.e., comprises both structural and semantic aspects of a text).

Chapter 3

Domain Dependency Graphs Visualization

3.1 Introduction

The evolution of unstructured text available has led to the development of highly advanced text analysis and knowledge discovery methods. Running on top, various interactive visualization tools have been offered to making the results of these methods accessible to the end-users. These tools, while effective, treat text representation models as a black box. Users can only visualize the final analysis results, while the internal text representation remains opaque to the end-users. A classic example in this respect is the search engine. It allows users to access stored text through a simple keyword-based search mechanism. However, users who search for information on a specific domain, may not know what they know and what they do not, and hence specifying the appropriate keywords is a challenge. It is important to allow users to visualize their intended information within a domain-specific context. If a system represents necessary information chunks concatenated in a context, users can access not only the target information but also unknown/unexpected relevant information. Bringing the domain experts' judgment and interpretation into the process of interactive visualization enables more efficient analytic workflows by reducing the amount of time and effort involved in analyzing and inspecting text.

The incorporation of human cognition into the visualization process does not make the design of the visual representation any easier, rather, on the contrary. The visualization component must account for interpretability issues at multiple levels and allow the user to switch seamlessly to the level that matches an intended analysis task in a specific domain. Results and findings should be put in context and made easy to relate to the original corpus.

As users know more about the text, they will have clearer exploration goals, which in turn, further initiates an iterative exploratory process. This fundamental procedure is expressed in the Visual Analytics Mantra of Keim [80]: "*Analyze First—Show the Important—Zoom and Filter, and Analyze Further—Details on Demand*".

To address the aforementioned challenges, this chapter presents DDG_{viz}, an interactive web-based tool, which enables the user to visualize and easily interpret a large amount of textual data at different levels of details. DDG_{viz} represents the most relevant topics in a corpus as compact clusters of DDGs. Users can navigate through a DDG of a topic and drill-down into its detailed structure. DDG_{viz} also enables users to interact, filter and search a DDG by adjusting various parameters and configurations.

Section 3.2 describes the back-end and front-end components of the DDG_{viz} and the detailed implementation of each component. It also includes several demonstrations across multiple tasks and options. Section 3.3 ends the chapter with a final remark concerning the evaluation of the DDG_{viz} front-end component and suggests possible future work.

3.2 DDG_{viz} Components

To implement DDG_{viz}, we adopt the concept of back-end/front-end decomposition, making the processing and visualization components independent. The back-end component is responsible for loading and preprocessing the text, and DDGs generation, while the front-end component handles the user interactions and Graphical User Interface (GUI). We describe the design and implementation details of each component in the following subsections.

3.2.1 Back-end Processing

The back-end component is written in Java. It runs a pipeline that reads a raw text corpus, preprocesses the text, enriches it with annotations, and generates and transforms the DDGs into a format that can be used by the front–end component. The annotation pipeline consists of: sentence segmentation[1], tokenization, lemmatization, topic assignment, domain-specific sentence selection, POS tagging, dependency parsing, Tf-Idf computation, and DDGs generation. We use the Stanford parser associated with collapsed dependencies[2] to parse and POS tag English texts, the German collapsed parser by Ruppert et al. [142] for German

[1] Using lt.seg script from: https://tudarmstadt-lt.github.io/seg/

[2] Stanford Natural Language Processing tools: https://nlp.stanford.edu/software/

texts[3][142]), and the Mate-Parser[4] [26] trained on Universal Dependencies (UD) treebanks[5] [123] for otherwise languages. We use pre-estimated models by GibbsLDA [130] to infer the topics of the new/unseen corpus. To get the PageRank[6] scores for the nodes, we use the JUNG library[7]. To get the list of distributionally similar words to a given word, we use the JoBimText Distributional Semantics framework[8] [21]. Cross-lingual similar words to a given word are obtained using BabelNet indices[9]. The final DDGs are sent by the java servlet to the front-end web client via HTTP/HTTPS in a JSON structure. The JSON file includes the necessary annotations required by the front-end visualization. Similarly, requests from the front-end client are sent to the servlet in a similar fashion.

3.2.2 Front-end Interface

The front-end component handles the user interactions and results visualization. It reads the graph directly from the generated JSON file. We use the D3.js[10] [27] to implement the user interface. D3.js uses only W3C-compliant web standards and languages like Scalable Vector Graphics (SVG), JavaScript, HTML5, and Cascading Style Sheets (CSS3), making it a suitable choice for our purpose.

DDG_{viz} provides two visualization modes: overview mode and ego mode.

Overview Mode

In the overview mode, users can get a general insight into the induced DDGs topics and their most relevant subtopics. Figure 3.1 shows the overview mode interface of the Amazon reviews corpus[11] and illustrates 200 topics identified in the corpus. Topics are represented using circles, packed together and distributed on a 2D space according to the hierarchical structure of the JSON file structure. If the JSON file is modified, the visualization will be modified when the user refreshes the web-page. The size of the circle represents the popularity of the topic in the corpus. As shown in Figure 3.1, by clicking on the circle, the circles can be zoomed, revealing the six most popular subtopics discussed within each topic. These subtopics represent the nodes with the highest Tf-Idf scores. Zoomable circle packing

[3] Dependency Collapsing framework: http://jobimtext.org/dependency-collapsing/

[4] Mate-Parser https://code.google.com/archive/p/mate-tools/

[5] Universal Dependencies: http://universaldependencies.github.io/docs/

[6] The value of the alpha parameter is set to 0.15

[7] http://jung.sourceforge.net/

[8] JoBimText: http://jobimtext.org

[9] BabelNet Indices 3.7 (August 2016), we were granted access to the full indices for non-commercial research use

[10] D3.js: https://d3js.org/

[11] SNAP: Web data: Amazon reviews https://snap.stanford.edu/data/web-Amazon.html

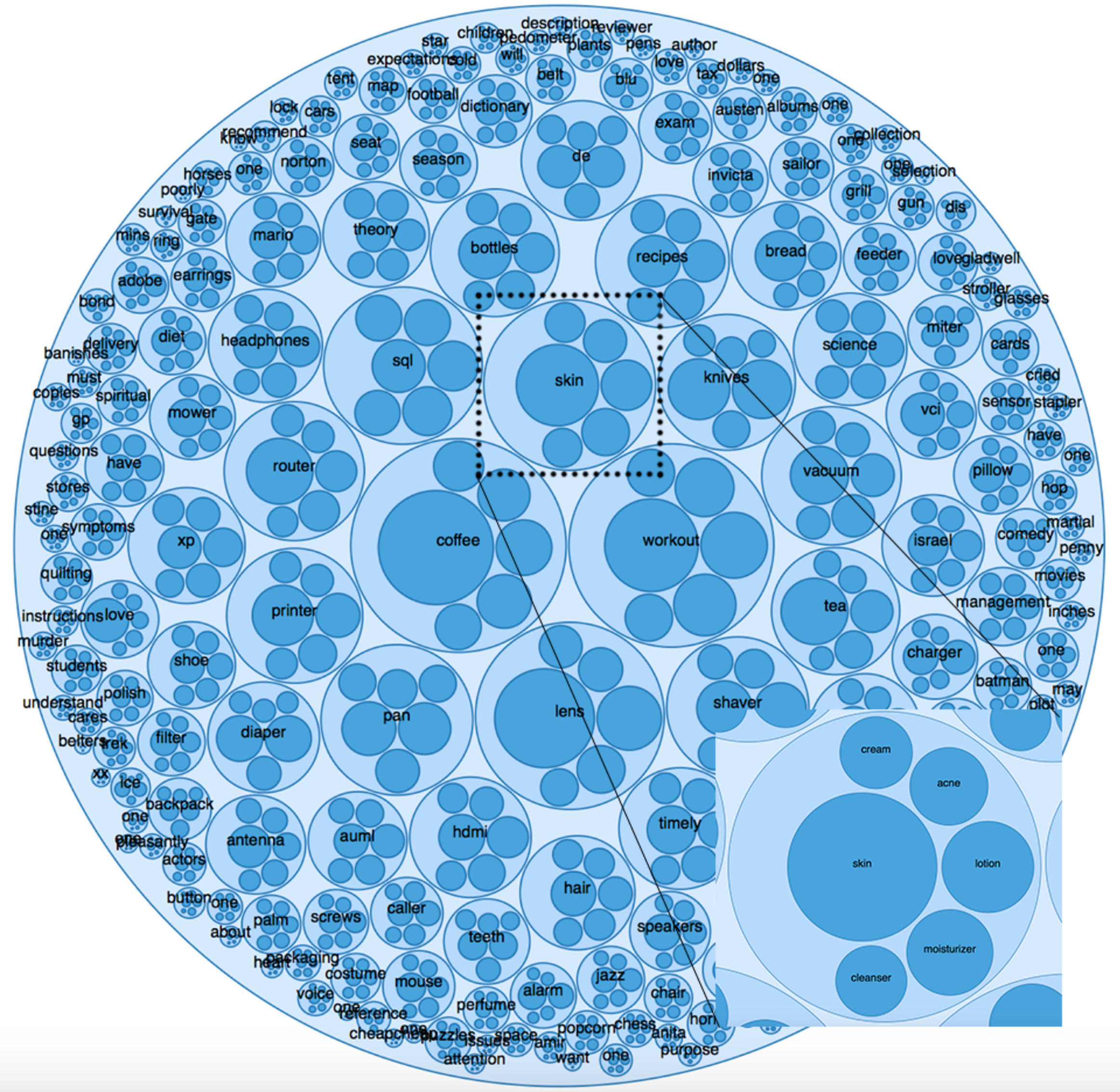
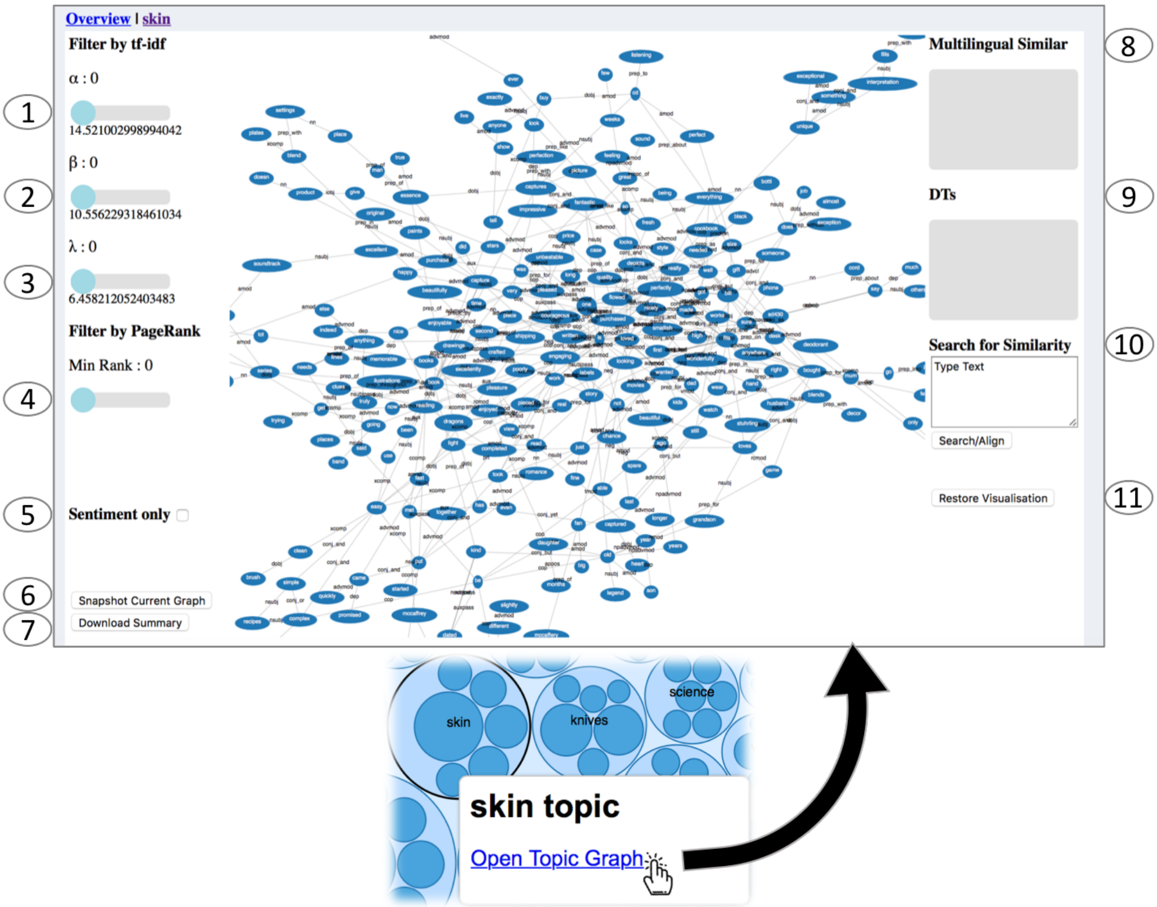


Fig. 3.1 Overview mode: topic distribution of Amazon reviews corpus. The circles show the topics discussed in the corpus and the size of a circle reflects the popularity of the topic.



Overview | skin

Filter by tf-idf

α : 0

14.521002998994042

β : 0

10.556229318461034

λ : 0

6.458212052403483

Filter by PageRank

Min Rank : 0

Sentiment only □

Snapshot Current Graph

Download Summary

Multilingual Similar

DTs

Search for Similarity

Type Text

Search/Align

Restore Visualisation

skin topic

Open Topic Graph

Fig. 3.2 Ego mode interface of the "skin" topic. Users can enter the ego-mode by clicking the link on the pop-up context menu of a certain topic. A new tab will appear in the upper-left corner with the topic name, showing the detailed structure of the DDG in the middle. Control elements are numbered to correspond with their associated description in Table 3.1.

functions of the D3.js library are used to handle all the interactions within this part of the visualizations. A right-click on any of the main circles will bring up a context menu with a hyperlink to the ego mode interface to that selected topic.

Ego Mode

When in **ego mode**, the users will be able to drill-down into the detailed structure of a DDG. Figure 3.2 shows the inner ego mode interface of the "skin" topic.

In the DDG visualization area (middle), the user can view the interconnections between words across the selected domain. The visualization container is zoomable and draggable, which enables users to focus on certain areas of a DDG. The nodes can be manually dragged and highlighted (single click), allowing users to modify the automatically generated layout

Element ID	Element Type	Description
1	Slider	The user can move the sliders to adjust the values of α_1, β_1 and λ_1
2		
3		
4		The user can adjust this slider to choose the minimum value of the PageRank weight
5	Checkbox	When checked, only aspect-sentiment relations will be shown in the visualization area, and the user will be able to download the list of identified aspect-sentiment pairs in a plain text file format
6	Button	Prompts a "save-as" dialog window, which enables the user to save a snapshot of the current DDG view as an image
7	Button	Prompts a "save-as" dialog window that allows the user to save the generated summary in a plain text file format
8	Scroll Box	Lists the distributionally most similar words retrieved by the JoBimText framework API
9	Scroll Box	Lists the top most similar words across other languages based on the BabelNet multilingual semantic network
10	Textarea	Allows the user to search the DDG by typing a search query.
11	Button	Resets the DDG visualization to default and clears all the applied filters

Table 3.1 Ego mode control elements functionality description, see Fig 3.2.

or examine the graph at varying degrees of detail. Several visual control elements are located on both sides of the visualization area. The user receives immediate visual feedback in response to any interaction with the input controls. DDG_{viz} supports incremental processing, such as the resulted DDG of an activity is the input DDG to the next activity unless the user presses the "Restore Visualization" button. A description of the control elements (numbered in Figure 3.2) and their functions are given in Table 3.1. Following, we explain the various control elements in detail.

(1,2,3 and 4) DDG Weighting Sliders: The sliders allow the users to adjust the different DDG weights' values, namely α_1, β_1 and λ_1 (see Section 2.2.6) and PageRank [53], respectively. By increasing the weights, only nodes and relations that have Tf-Idf or PageRank scores higher than the selected weights will be instantly displayed on the DDG visualization area. This way, users can narrow down their focus into highly relevant relations.

(5) Sentiment only: When checked, only aspect-sentiment relations identified by our unsupervised rule-based approach (Section 4.3) will be shown on the DDG visualization area. Also, a download link will appear and the user will be able to download a text file, which lists all the identified aspect-sentiment pairs, see Figure 3.3.

(6) Snapschot Current DDG: The button initiates a "save-as" dialog window, which enables the user to save a snapshot of the current DDG view as an image to the local computer.

(7) Download Summary: We choose to describe the details of the summarization task in this chapter, as the results of the back-end methodology are not officially evaluated. Following the idea of TextRank by Mihalcea and Tarau [104], we utilize the ability of DDGs in retrieving highly relevant domain-specific words and relations to generate unsupervised domain-specific extractive multi-document summaries. The suggested approach requires no gold-standard text-summary pairs for training (unsupervised) and produces a summary that includes a subset of sentences from the original text corpus (extractive). The size of a summary depends on the DDG weighting parameters' values. The lower the given weights, the shorter the summary length, and vice versa.

The process of generating a summary may be described as follows:

1. Based on the current status of the DDG, the client-side sends a request to the back-end with the DDGs' edges as a JSON file. The back-end, in turn, runs a function that traces back the edges (relations) to their sentences in the original corpus.
2. Since all the selected sentences belong to the same topic and are traced back using the same set of relations, we observe a high percentage of redundant information among sentences.
3. Clustering provides a bare-bones way to help determine what information is repeated in multiple sentences. However, to compare sentences for clustering, we need a way to generate a fixed-length vector representation for each sentence. This representation (also called sentence embeddings) should be able to encode the meaning of the corresponding sentence. For this task, we use the Skip-Thoughts approach, as described in [83][12]. For clustering, we choose the Affinity Propagation (AP) algorithm. AP

[12] Skip-Thoughts encoder is trained using Wikipedia dumps

clustering is based on the concept of "message passing" between data points [57]. Unlike K-means, AP does not require the number of clusters to be determined or estimated before running the algorithm.

4. Each cluster of sentences includes a set of semantically similar sentences whose meaning can be represented by just one sentence to be added to the final summary. For each identified cluster, we choose to select the sentence whose embeddings vector is the closest to the cluster representative (also called centroid).

5. The final step is to form a cohesive summary by re-ordering the selected sentences. The selected sentences are ordered according to the majority position among the sentences in their corresponding clusters.

The following summary is generated based on the DDG of the "Flüchtlinge" (tr. refugees) topic extracted from the Spiegel Online German news corpus. The news were collected over a one-year period (2015), with a total of 33121 articles, thus the topics are highly correlated with one-fifth of the total articles discussing the "refugees" topic.

> *Bei weiteren Einsätzen nahmen die Küstenwache sowie ein niederländisches Frachtschiff 320 Flüchtlinge an Bord, die vor Malta in Seenot geraten waren. Gegen den Kapitän und ein Besatzungsmitglied ermittelt die Staatsanwaltschaft im sizilianischen Catania. Im Januar schreckte die europäische Grenzschutzagentur Frontex den Kontinent mit einer ungeheuerlichen "Beobachtung" auf: Schleuser würden Flüchtlinge auf Frachtschiffe ohne Besatzung packen, sogenannte Geisterschiffe. Vor einem "neuen Grad der Grausamkeit" im Mittelmeer wurde gewarnt, Innenminister Thomas de Maizière verurteilte den "grenzenlosen Zynismus der Schleuser" und forderte die EU auf, "mit größter Entschlossenheit und Beharrlichkeit aktiv zu werden ". Die Schlepper waren auch keine gewissenlosen Menschenhändler, sondern syrische Seeleute. Die vermeintliche Skrupellosigkeit, Geldgier, Niedertracht der Schlepper ist das einzige Narrativ, auf das sich Europas Politiker in der Asyldebatte verständigen können. So auch jetzt. Lediglich in einem Punkt besteht Einigkeit: Der Kampf gegen Schlepper soll ausgeweitet werden. Schätzungen zufolge kamen seit Jahresbeginn mehr als 1750 Menschen bei der Überfahrt von Libyen ums Leben. Es sei zudem ein "gravierender Fehler" gewesen, den italienischen Seenotrettungseinsatz " Mare Nostrum" einzustellen. "Wir müssen die Türe ein wenig öffnen, damit die Menschen nicht durch ein Fenster einsteigen müssen", sagte er.*

The summary above was manually evaluated by 10 linguists who are also native German speakers. The judges were asked to evaluate the above summary based on coherence and informativeness by assigning two scores on a scale of [0-5], 0 is poor, and 5 is good. The informativeness score assesses the degree to which the summary provides sufficient background information to understand the news without repetition, while the coherence score assesses the topical flow of the text. The average coherence and informativeness scores provided by the evaluators are 4.55 and 4.77, respectively. The only remark we received from the evaluators is regarding the first sentence, as some found it illogical to start the summary with. The first three words of the sentence (tr. "In other operations . . . ") implies that it is preceded with a text.

When applied directly to a corpus, clustering-to-summarize was found to yield poor-coherent summaries, as many of the selected sentences are difficult to make sense of in isolation [172]. However, running on top of DDGs, this approach can generate highly-coherent self-contained domain-specific summaries.

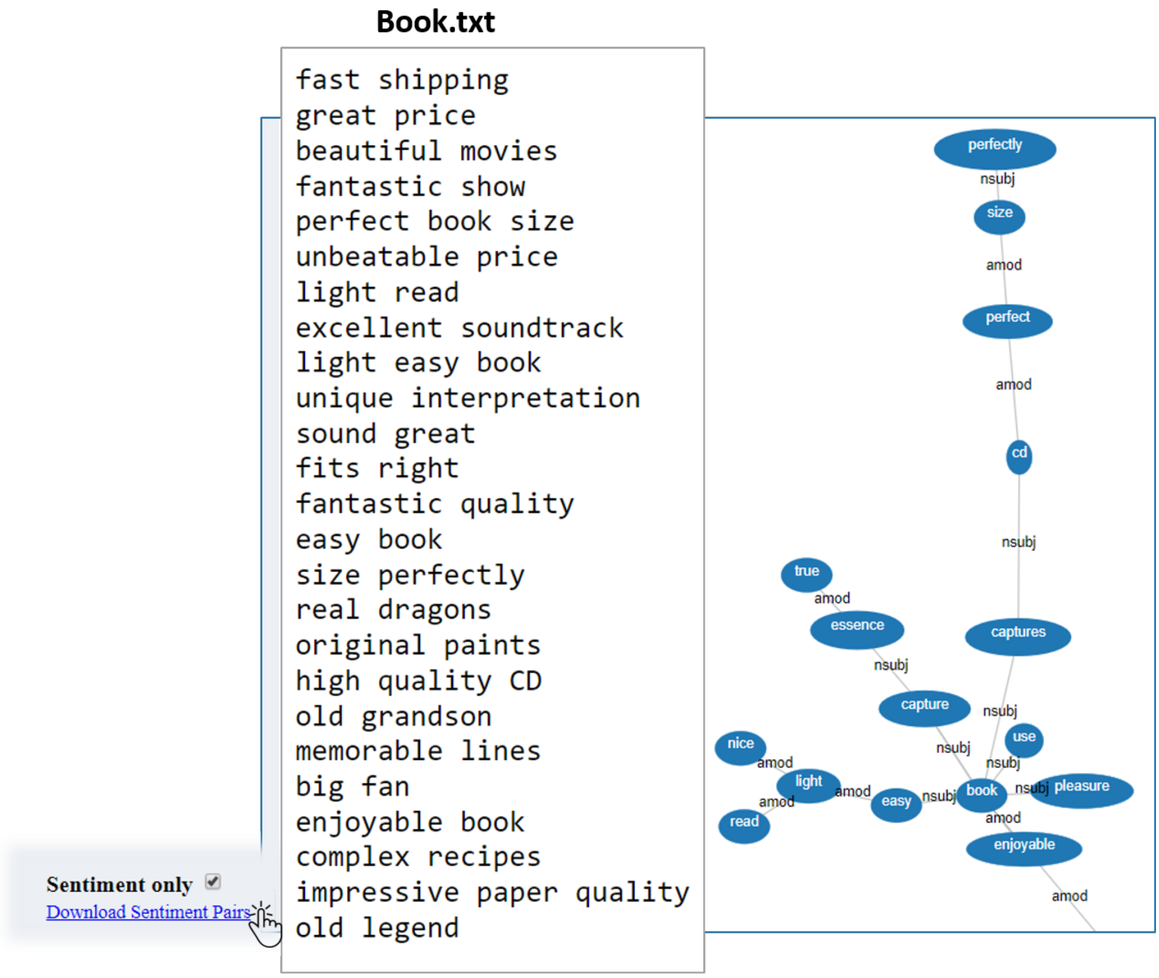
(8 and 9) Multilingual and Distributionally Similar Words: When a node is selected with a single click, the scroll boxes will list the top 10 similar words according to distributional thesaurus and BabelNet. An illustration is shown in Figure 3.4.

(10) Search DDG: This option gives the user the ability to search the DDG by typing a search query. The back-end algorithm uses dependency structure similarity of local neighbors within a Shortest Path (SP $\leq \mathbf{t}$) to find an approximate sub-graph in the DDG that matches the given query. The algorithm is explained at length in Section 5.3.2. A demonstration of the search option is given in Figure 3.5.

(11) Restore Visualization: The button resets the DDG visualization to default and clears all the applied filters.

3.3 Final Remark

At this stage of our research, an evaluation of the front-end GUI is not yet provided. The goal of this chapter is to offer the users a tool to be able to grasp and interact with DDGs at different levels of abstraction and to allow them to interpret and explain the results of different NLP tasks while looking at the visual text representation. In the future, we plan to conduct a systematic evaluation based on the Human-Computer Interaction (HCI) experimental evaluation method [135].



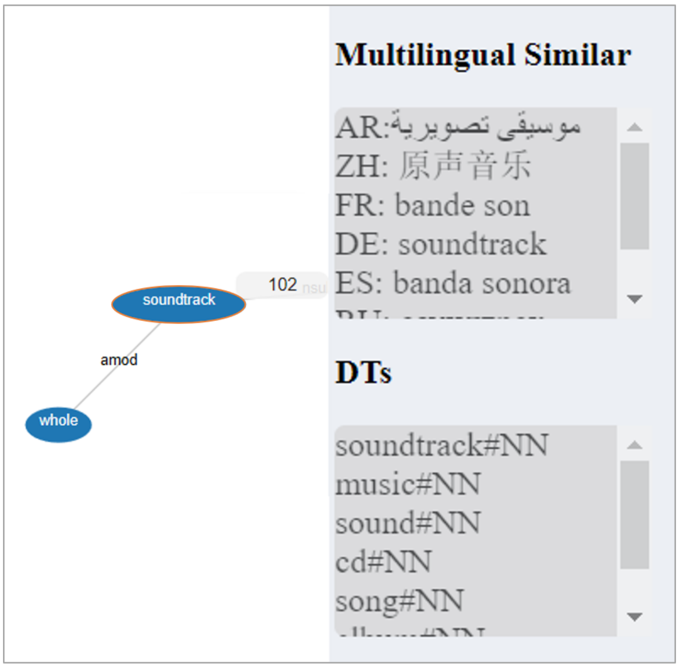
Book.txt

```
fast shipping
great price
beautiful movies
fantastic show
perfect book size
unbeatable price
light read
excellent soundtrack
light easy book
unique interpretation
sound great
fits right
fantastic quality
easy book
size perfectly
real dragons
original paints
high quality CD
old grandson
memorable lines
big fan
enjoyable book
complex recipes
impressive paper quality
old legend
```

Sentiment only ☑

Download Sentiment Pairs

Fig. 3.3 A demonstration of the "sentiment only" option. When checking the box (left side), only sentiment relations will be shown in the DDG visualization area (right side), and a download link will appear to allow the user to download all the aspect-sentiment pairs in plain-text format, each in a separate line (middle segment). False positive sentiment expressions will be discussed in Section 4.5.



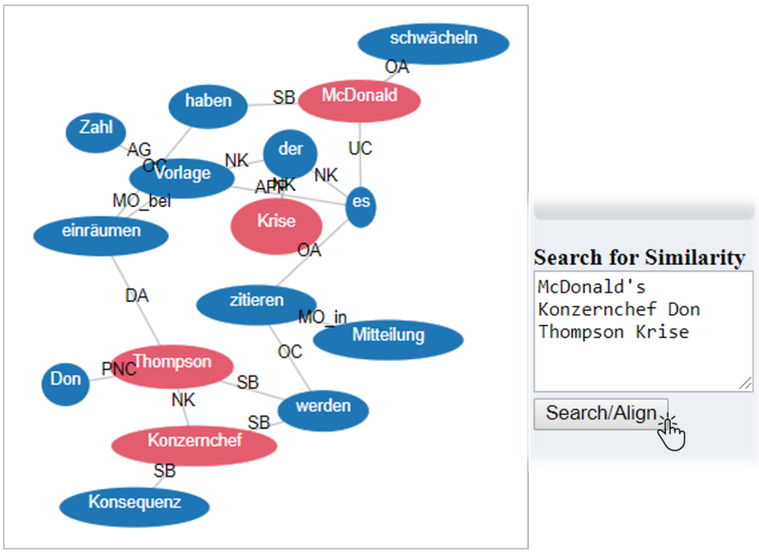
Multilingual Similar

AR:موسيقى تصويرية
ZH: 原声音乐
FR: bande son
DE: soundtrack
ES: banda sonora

DTs

soundtrack#NN
music#NN
sound#NN
cd#NN
song#NN

Fig. 3.4 The user can single click on a node and the node will be highlighted along with its direct neighbors. All the other nodes will remain hidden until the user clicks once on the empty DDG area. On the right side, similar multilingual words from the BabelNet network will be displayed on the "Multilingual Similar" scroll box, and distributionally similar words retrieved by the JoBimText API will be displayed on the "DTs" scroll box. Hovering over a node reveals a transparent tooltip showing the frequency of the word in the corpus.



Search for Similarity

```
McDonald's
Konzernchef Don
Thompson Krise
```

Fig. 3.5 A demonstration of the approximate sub-graph matching using Algorithm 1. The algorithm uses dependency structure similarity of local neighbors within the Shortest Path (SP $\leq$ **t**), to find an approximate sub-graph in the DDG that matches the given query. In this example, **t = 3**. The example is taken from the Spiegel Online German news corpus. Red nodes represent the overlapping words between the query text and the DDG. Direction and dependency types are ignored during the matching process. The resulted sub-graph will be displayed in the DDG visualization area, and otherwise nodes and relations will become invisible.

Chapter 4

Aspect-Based Sentiment Analysis

4.1 Introduction

E-commerce and social media technologies have become an excellent platform for a vast number of users to share and explain their opinions online. Websites (e.g., amazon.com, tripadvisor.com) allow users to post and read reviews about various services and products. Such reviews are important for customers to make a purchase choice, as well as for organizations to monitor and improve their products and reputation. However, user-generated reviews are unstructured and noisy. In the past few years, there has been a significant body of work that adopts NLP and text mining tools to better process, analyze, and understand arguments and opinions from various types of information in user-generated reviews. Such efforts have come to be known as sentiment analysis or opinion mining. In Oxford Dictionary[1], "opinion" is defined as "a view or judgment formed about something which is not necessarily based on fact or knowledge". In general, it is the aggregation of individual beliefs/attitudes to an event, product, person or public policy held by the adult population [93].

In most relevant works to the date, sentiment analysis and opinion mining have been explored in three levels of granularity [93]: document-level, sentence-level and aspect-level. For document-level sentiment analysis, the task is to decide if the general opinion discussed in a document (e.g., review) is positive, negative, or neutral (also called polarity identification or sentiment classification). Sentence-level sentiment usually requires a subjectivity classification step prior to sentiment classification, which is important to identify if the sentence is subjective (opinionated text) or objective (fact). Aspect-based sentiment analysis (ABSA) performs a finer-grained sentiment analysis by addressing three subproblems: (1) extracting aspects from the review text, (2) identifying the entity that is referred to by the aspect, and

[1] https://en.oxforddictionaries.com/definition/opinion

finally (3) classifying the sentiment polarity toward the aspect. For example, a review of the "entity" camera is likely to discuss distinct "aspects" like zoom, lens, resolution, battery life, price and memory. A single product review by a user may trigger a positive "sentiment" about one aspect, and negative about another. Some researchers may refer to the first sub-problem as Opinion Target Expression (OTE) extraction. The terms "OTE" and "aspect" are used interchangeably in this research. SemEval shared task of ABSA organizers have introduced a new subtask called: aspect category detection (E#A). The task is to identify every entity E and attribute A pair toward which an opinion is expressed in the given text. E and A should be chosen from predefined inventories of entity types (e.g., LAPTOP, MOUSE, RESTAURANT, FOOD) and attribute labels (e.g., DESIGN, PRICE, QUALITY), respectively.

This chapter explores the effectiveness of using Domain Dependency Graphs (DDGs) features to tackle several tasks in ABSA, namely aspect identification, E#A identification, OTE extraction and aspect-based sentiment polarity classification. Throughout this text, we distinguish between two terms "aspect identification" and "aspect extraction". Aspect extraction or OTE extraction focuses on finding the starting and ending aspects offsets in a given text reviews, while aspect identification defines the aspect list of a certain entity. Aspect identification is a preceding step to aspect-based summarization, which aggregates sentiment over each aspect to provide the user with an average numeric or symbolic rating [163].

The remainder of this chapter is organized as follows. Section 4.2 summarizes related works and discusses the main research problems within the field of ABSA. Section 4.3 describes our unsupervised methodology for aspect identification and OET extraction. Within the same section, we introduced LexiExp, a tool for expanding existing sentiment seed lexicon based on the notion of distributional thesaurus. We also show how LexiExp can provide a polarity estimation for the new expanded lexicon using statistical co-occurrence calculation. In Section 4.4, we describe our supervised models for different subtasks of ABSA, and the features we used to train each model. Finally, in Section 4.5, we explain our experimental results and evaluation. We demonstrate the effectiveness of our approaches into four different tasks:

1. **Unsupervised data-driven aspect identification:** we try to improve the overall understanding of opinion patterns and distinguish the most compelling aspects for different product categories from the Amazon reviews corpus.

2. **SemEval-2016 ABSA Shared Task:** we experiment and evaluate our supervised and unsupervised approaches on the SemEval ABSA shared task benchmark datasets (reviews) across various domains and languages. Our experiments cover the three

sub-problems mentioned above, namely aspect category identification, opinion target expression (OTE) and sentiment polarity classification.

3. **Sentiment Analysis for Indian Languages:** we perform tweet-level sentiment polarity classification experiments on tweets in two Indian languages, Bengali and Hindi. The experiments are part of our participation in the Shared Task on Sentiment Analysis in Indian Languages (SAIL).

4. **Sentiment Analysis of Customer Reviews:** we experiment explicit and implicit aspect extraction, as well as aspect-based sentiment polarity classification on a dataset of customer reviews for five electronic products.

4.2 Related Work on ABSA

Before 2014, there was no established subtask decomposition for ABSA, nor there were any established evaluation benchmarks or measures to support this decomposition. Aspect-based sentiment categorization researches skipped the step of aspect extraction by either, labeling the aspects manually, selecting the highly frequent nouns [22, 70], or using a predefined lexicons/dictionaries, which include well-known aspects for entities like hotels/restaurants aspects (décor, food, service, etc.). If not regularly updated, pre-collected lexicon may not recognize new aspects or aspect synonyms (e.g., battery, charge system, power). Zhai et al. [182] developed a semi-supervised technique to build clusters of aspects. Each cluster has a set of synonyms that are likely to refer to the same product aspect.

Researches are placing more effort into relating extracted aspects with their appropriate opinion words to automate the process of OTE extraction. Early attempts have used shallow parsing features, such as Part-of-Speech (PoS) tags. A linguistic parser is first used to parse each review and produce the PoS tags (verb, noun, adjective, etc.) for each word. For each detected noun, the sentiment regarding this noun is judged by its nearest adjacent adjective opinion word [70]. However, the limitation of these methods is that many frequent noun phrases that may not represent product aspects are retrieved, especially multiword aspects (i.e., aspects made up of more than one word, e.g., "battery life"). They also fail to identify infrequent entities effectively and handle long-distance dependency, as we discussed previously in Section 1.1. Some researchers used pattern matching methods. They define hand-crafted rules and templates from fully labeled data and use them later to extract OTEs from unlabeled text [76, 180]. Although these methods have been applied successfully in many cases, it is not easy to manually summarize enough rules and patterns to cover all the possible OTE patterns. ABSA task is also sensitive to the domain of the training data;

thus, extensive annotation for a broad set of data for every single domain has to be carried out, which is not practically feasible [168]. Efforts for cross-domain sentiment analysis apply domain adaptation by limiting the set of features to those that are domain-independent [74, 92, 139]. An issue with these methods is that words and phrases used for expressing opinions can differ considerably from one domain to another.

Since the official release of the SemEval ABSA shared task in 2014 [134], unsupervised approaches started to emerge so as to reduce the amount of manual labeling and hand-crafted rules. They all use dependency-tree-based features to automatically recognize the OTE patterns. However, these ideas are still unable to cope with more challenging problems, like co-reference, long-distance dependencies and negation. They consider sentences that contain only a single aspect, ignoring the phrasal structure of some aspects.

Most literature on sentiment classification relies on semantic resources such as a sentiment lexicon. Sentiment lexicon contains a list of words and phrases that convey sentiment. A widely used sentiment lexicon resource is SentiWordNet[2] [54]. SentiWordNet is a sentiment lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information. While there are several sentiment lexicons for English, only a few, if none, available for other languages.

4.3 Unsupervised Approach for ABSA

We present a generic unsupervised approach to automatically identify aspects from user's reviews and assess the degrees of overall sentiment toward each identified aspect. Our method is completely unsupervised and needs no labeled training data or previous knowledge about the domains, and follows the Structure Discovery paradigm [20]. It is mainly composed of three processes, as shown in Figure 4.1.

The process starts by extracting the underlying DDGs from unstructured mixed-domain reviews, each DDG aggregates in-domain reviews for a particular entity (e.g., camera, car, computer). To generate such graphs, we have first filtered out irrelevant low-quality reviews, and cataloged high-quality in-domain reviews into representative topics using topic modeling. The goal is to summarize this huge amount of unstructured user-generated data from the web and turn it into a compressed structured representation, which is considered a necessary basis for the next stage. It facilitates the accurate identification of opinion patterns and helps to distinguish the most effective aspects for different entities. A snapshot from an automatically generated DDG from Amazon reviews corpora, which represents and summarizes all reviews within the "camera category" is shown in Figure 4.2.

[2] https://sentiwordnet.isti.cnr.it/

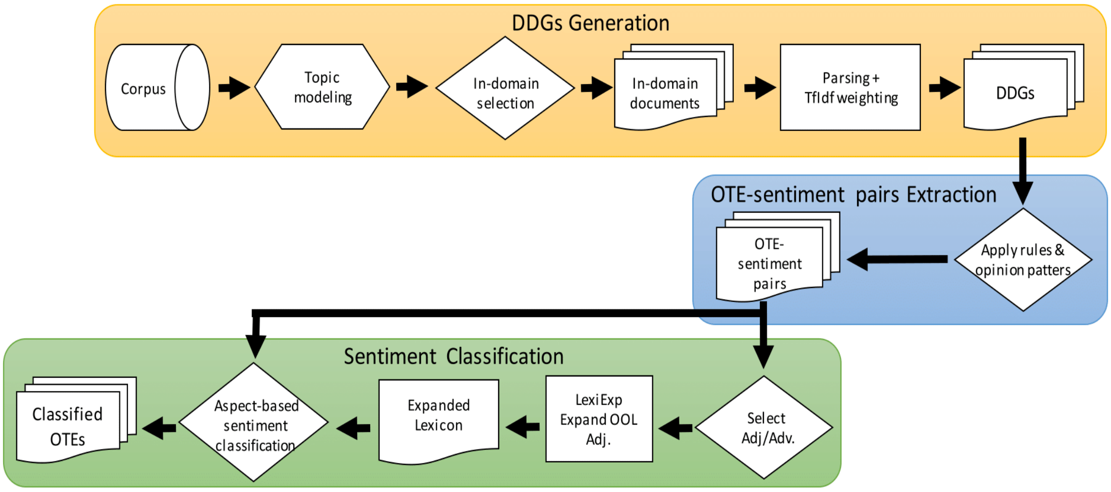


Fig. 4.1 The pipeline of the unsupervised ABSA framework.

The discussion of Stage 1 will be skipped since it was explained in detail in Chapter 2. In Stage 2, we identify opinion-related expressions, and summarize them as a set of OTE-sentiment pairs using high-level linguistic features (e.g., sharp-lens, clear-photo, nice-color). We identify adverb-adjective-noun phrases based on clause structure obtained by parsing sentences into a well formed linguistic structure. The approach uses the enhanced++ dependencies to identify and extract sets of OTE patterns [149]. Finally, in Stage 3, each extracted OTE-sentiment pair is assigned one of the following polarity labels: "positive", "negative", "neutral". However, reviewers use a wide variety of vocabulary to express their opinion toward aspects, many of which may not exist in sentiment lexicons. As a part of Stage 3, we present LexiExp, an algorithmic framework to handle Out of Lexicon (OOL) sentiment expressions. The following two subsections explain Stages 2 and 3 in detail.

4.3.1 Stage 2: OTE-sentiment Pairs Extraction

After DDGs are constructed, the next step is to extract opinion phrases. This involves the extraction of two pieces of information, the entity aspects and the corresponding opinion toward each extracted aspect. We propose a novel rule-based model, which incorporates high-level linguistic structural information to enhance the performance of OTE-sentiment pairs extraction from text reviews, and eventually improve the accuracy of sentiment polarity classification. Before going into further detail, we state two assumptions behind our model. First, we assume that each discovered DDG discusses only one coherent topic (i.e., product or

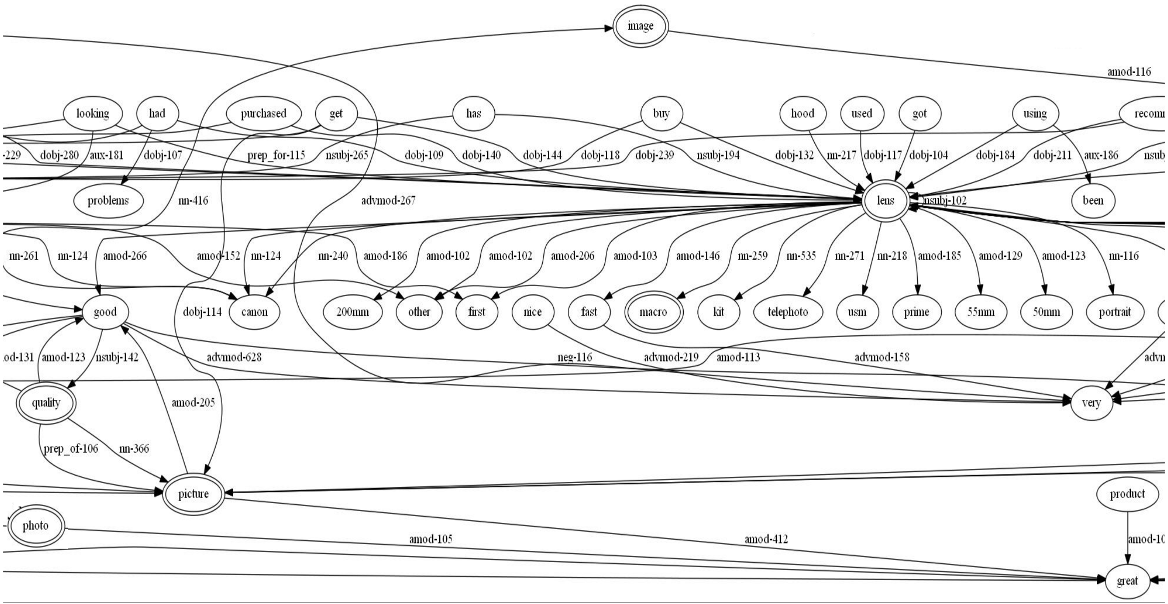


Fig. 4.2 An excerpt from the automatically generated DDG of the camera review topic. Double-lined nodes represent aspects, and lines in bold highlight aspect-sentiment relations. Only the most frequent relations are shown for the purpose of presentation. The graph is created using the Graphviz software package [51].

entity), and therefore comprises all its relevant in-domain aspects. Irrelevant out-of-domain noisy sentences are filtered out. Second, aspects and their corresponding sentiment(s) are explicitly mentioned and can be extracted from the DDG discussing the corresponding entity.

Most ABSA systems that require a syntactic representation use basic dependency relations, which are guaranteed to be a strict surface syntax tree. However, strict surface-structure dependency trees tend to follow the linguistic structure of sentences too closely, and frequently fail to provide direct relations between content words. Thus, it is difficult for these systems to cope with complicated phrases and challenging problems like implicit long-distance and negation. To clarify, Figures 4.6, 4.7, 4.8 and 4.9 present four examples of nontrivial opinion expressions and describe the problematic issues around these examples.

Figure 4.6 shows the constituency-based parse tree and the enhanced dependency structure for the sentence "The chicken recipe, which reviewers said very good, is average". The opinion target "recipe" has a long-distance dependency with its correct sentiment "average". By extracting the adjacent adjective "good", the final polarity classification will not be accurate. The same applies to the example in Figure 4.7, "The pizza with mushroom or pepperoni is really good", if the adjective "good" is attached to its closest noun, only "pizza with pepperoni-good" pair will be extracted while "pizza with mushroom-good" will not be identified.

Another challenging issue is to accurately determine the scope of negation [113]. The most popular approach of sentiment negation is to flip the sentiment polarity when a negation is detected. For example, consider a review sentence "This hotel staff is not friendly", the negation word "not" is used to express the opposite sentiment of "friendly", which is negative. However, this may not always be the case, for example, consider a review sentence "This retina screen isn't just clear, it's also cheap". The type of negation using "n't" in this sentence does not represent the opposite sentiment. The same can be noticed in the examples in Figure 4.8 and Figure 4.9.

From the generated DDGs, we extract opinion phrases and patterns based on carefully-designed defined rules. These rules are domain-independent, and based solely on high-level syntactic structure. Besides, the combination of Tf-Idf-filtered DDGs makes our approach insensitive to frequent words. Our approach uses PoS tags and typed dependency relationships which are obtained by the enhanced++ parser [149]. The enhanced++ dependency parser extends the basic collapsed Universal Dependencies (UD) representation. It contains additional and augmented relations that make the otherwise implicit relations between content words more explicit and easy to capture. It also encodes an additional layer of semantic dependencies and therefore provides higher-level insight into the semantic meaning of the text.

To extract the OTE-sentiment pairs, a set of generation rules is carefully defined to only extract sets of related adjectives and nouns as follows:

1. All modifying adjectives attached directly to a noun in a noun phrase (e.g., nice camera).

2. Adjectives related to nouns in a subject-predicate relationship.

These rules correspond to our observation that opinions or relations between opinion word and opinion target, are mostly expressed with adjectival modifier (amod), nominal subject (nsubj), or both as in Figure 4.3. In the case of the latter, both adjectives are listed as sentiments to the aspect. It is important to mention that before applying the rules, compounds of nouns (i.e., NN, NNS), and consecutive foreign words (FW) are merged into a single token. In opinionated text, noun compounds, and foreign words represent multiword aspects (e.g., zoom range, battery life, etc.) and foreign borrowed words (e.g., mee ka tee), respectively, see Figure 4.5. We also propagate the effect of conjunction relations "conj:and and conj:or" to expand the effect of the sentiment when more than one aspect is mentioned, see Figure 4.7.

Relating aspects with their corresponding sentiment words can also be more accurately accomplished, even in the case of long-distance dependencies. To demonstrate, we show

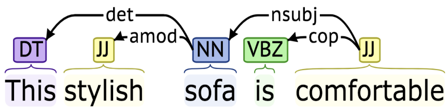


Fig. 4.3 Review sentence example that has both opinion patterns: a nominal subject (nsubj relation) and a modifying adjective (amod relation). Both sentiments "*stylish*" and "*comfortable*" are related to the same aspect noun "*sofa*". In this case, both adjectives are listed as a sentiment to the aspect "*sofa*".

how the issues discussed previously in Figures 4.6, 4.7, and 4.9 are resolved. In Figure 4.6, the enhanced dependencies show that the syntactic subject of a compound clause "chicken recipe" is actually "average", which indicates that the "chicken recipe" phrase should be associated with the adjective complement. In Figure 4.7, the adjective "good" is related to the aspect "pizza" through a subject-predicate "nsubj" relationship. Since "pizza with pepperoni" and "pizza with mushroom" are associated using a conj relation "conj:or", and both have a nominal modifier "nmod-with" relation with "pizza", we can easily associate "pizza" with both "mushroom" and "pepperoni". Flatten this tree into a graph representation, the two opinion patterns would be easily distinguished "good-**nsubj**-pizza-**nmod:with**-mushroom" and good-**nsubj**-pizza-**nmod:with**-pepperoni", which can be seen more clearly through the constituency parse tree in Figure 4.7. In Figure 4.9, OTE-sentiment pairs mentioned in the review (i.e., "high quality pictures", "large zoom range") can be easily captured by enforcing the first rule (i.e., by extracting "amod" relations). We can also directly relate both aspects to their main entity "camera" by following the "nsubj" relations and the implicit "conj:and" provided by the enhaned++ parser.

If a noun or a noun compound is in an nsubj relation with a verb, which is not an auxiliary verb (i.e., is, was, were, has, etc.), and the verb is modified by an adjective or an adverb, or has an adjective complement, we identify this as an opinion expression and merge the verb into the nsubj relationship, see Figure 4.4.

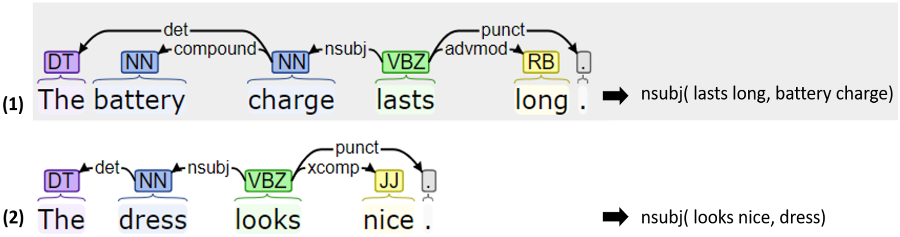


Fig. 4.4 "*battery charge*" and "*dress*" are in an "nsubj" relation with verbs "*lasts*" and "*looks*", respectively. In (1) "*lasts*" is modified by the adverb modifier "*long*", and in (2) "*looks*" is in an open clausal complements relation (xcomp) with the adjective "*nice*".

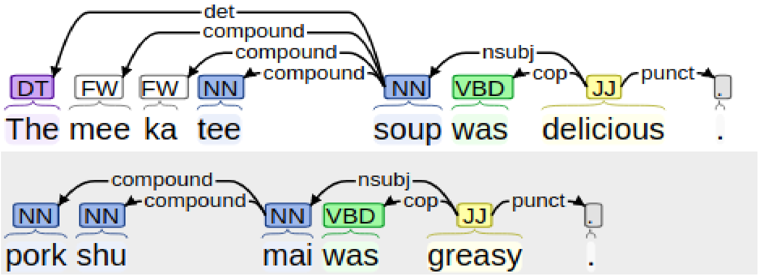


Fig. 4.5 Multiword aspects usually surface in the form of compound nouns.

Interpreting the scope of negation is another important issue. To examine the impact of negation words on the polarity of the sentiment, we initially find all the occurrences of negation words, such as "no" and "not". Then, for each negation term, we traverse the "neg" relations through the dependency graph edges. Every expression paired with a negation word is affected by the negation trigger. An exception to this rule is "not only", "not even" and "not just". In these cases, the scope of negation is limited to "only", "even" and "just", see Figure 4.8 and Figure 4.9.

We noticed that the degree/intensity of the sentiment has a negligible effect on the sentiment polarity class, so adverbial intensifiers are not considered in this study.

4.3.2 Stage 3: Sentiment Classification

After extracting the OTE-sentiment pairs, the next step is to accurately classify whether the sentiment associated with each extracted OTE is "positive", "negative" or "neutral". To do so, we compare the sentiment word against a sentiment lexicon. Sentiment lexicons can be viewed as dictionaries of words, each is associated with a value that indicates the degree to which this word is positive or negative (i.e., degree of sentiment polarity). Many sentiment analysis methodologies rely on sentiment lexicons to classify/predict sentiment polarity. Such methodologies are often called lexicon-based methodologies. Lexicon-based approaches have achieved state-of-the-art results on the SemEval shared task benchmark datasets for the sentiment classification task of the years 2013-2016 [108], [65], [89]. Inaccurate classification of sentiment to the opposite sentiment can only happen due to sophisticated forms of language, such as sarcasm or humor. Lexicon-based sentiment classification approaches are also proved to be efficient and have low run-time overhead compared to other methodologies [73].

Sentiment lexicons can be generated manually using existing dictionaries, such as the General Inquirer [157], or automatically using dictionary-based and corpus-based methodologies [93]. Dictionary-based approaches rely on semantic relations (i.e., synonym, antonym, etc.) from lexical resources like WordNet, in order to bootstrap a given seed set of sentiment

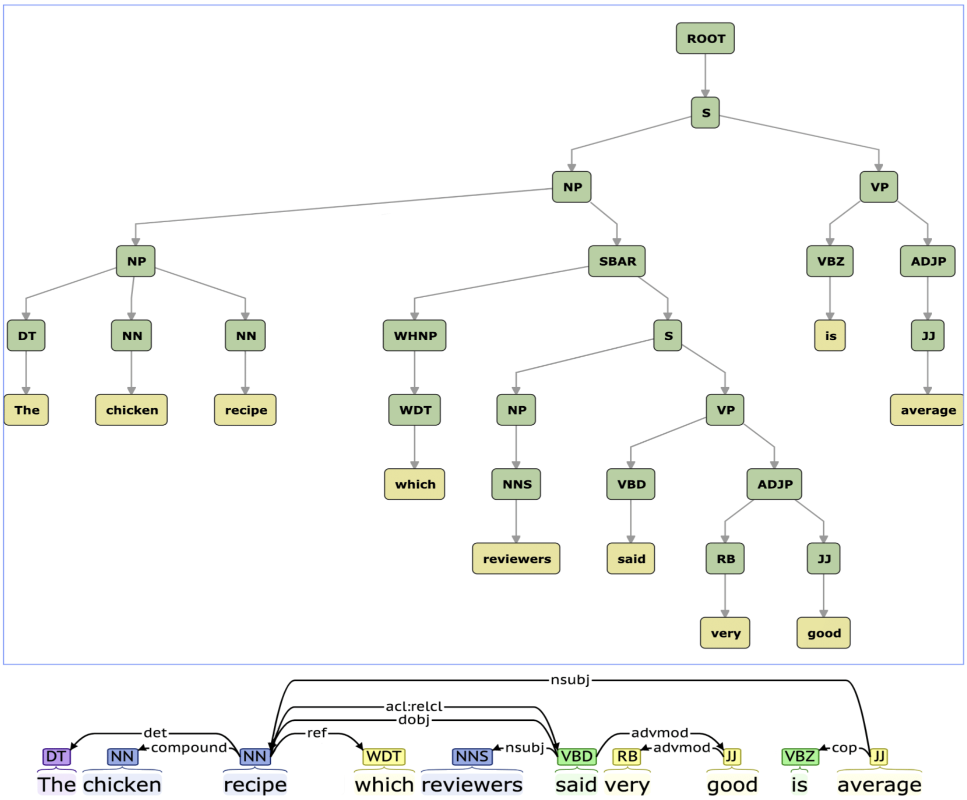


Fig. 4.6 The constituency-based parse tree and enhanced dependency structure for the sentence "*The chicken recipe, which reviewers said very good, is average*". OTE-sentiment pair "*recipe-good*" does not have a direct dependency relation. If the nearest adjective "*good*" is only considered, the final polarity classification will not be accurate. The enhanced dependencies indicate that "*chicken recipe*" should be associated with the adjective complement "is average", and hence the syntactic subject of a compound clause "*chicken recipe*" is actually "*average*".

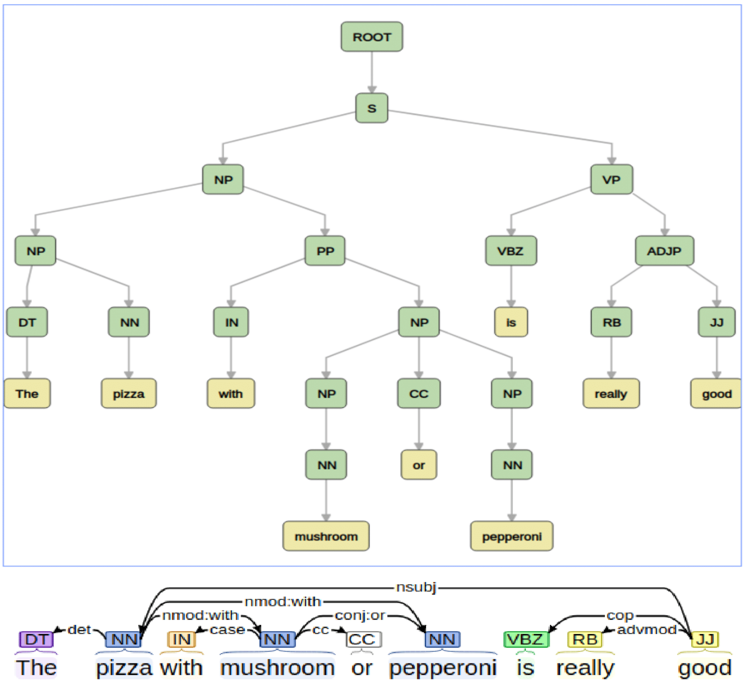


Fig. 4.7 Enhanced++ dependencies and constituency parsing of the sentence "*The pizza with mushroom or pepperoni is really good*". If an adjective is just attached to its closest noun, only "*pepperoni-good*" pair will be extracted, while "*mushroom-good*" will not be identified. This is caused by long-distance dependency in the sentence. However, since both are associated using a "conj" relation and both have a nominal modifier "nmod-with" relation with "*pizza*", we can easily associate "*pizza*" with both "*mushroom*" and "*pepperoni*". Traversing the DDG, two opinion patterns can be easily distinguished "*good-nsubj-pizza-nmod-with-mushroom*" and "*good-nsubj-pizza-nmod-with-pepperoni*".

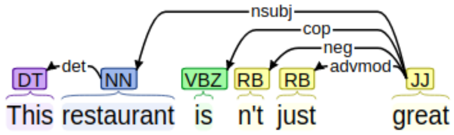


Fig. 4.8 Enhanced dependency structure of the sentence "*This restaurant isn't just great*". Due to the presence of the adverbial modifier "advmod" relation between "*just*" and the sentiment word "*great*", the negation word "*n't*" will not affect the polarity of the sentence.

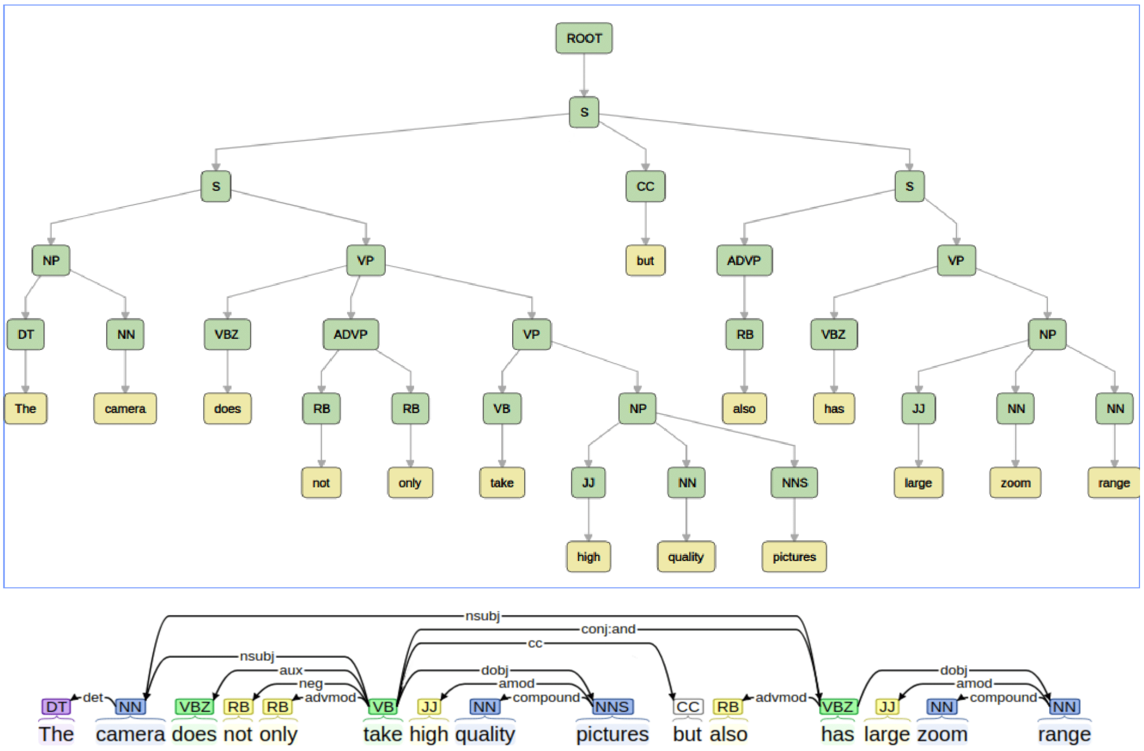


Fig. 4.9 The constituency-based parse tree and enhanced dependency structure of the sentence "*The camera does not only take high quality pictures, but also has large zoom range*". The entity "*camera*" has two subject-predicate relationships with "*take high quality pictures*" and "*has large zoom range*". Due to the presence of the adverbial modifier "advmod" relation between "*only*" and the verb "*take*", the negation word "*not*" will not affect the polarity of the clause.

words, while corpus-based approach exploits large corpus, and apply various linguistic rules and logical connectives like "AND" and "OR" to generate lexicons. For example, in the sentence: "The place was clean and classy", if "clean" is known to be positive, we can infer that "classy" is also positive.

Several opinion lexicons are publicly available [38]. However, many lexicons remain non-comprehensive for most languages compared to English sentiment lexicons. Additionally, most existing lexicons were compiled using formal text, such as news, thus, it misses a large number of informal or slang sentiment words used in micro-blogs and social media platforms. For this reason, it is necessary to expand existing sentiment lexicons to obtain up-to-date lexicons with higher coverage. In the next subsection, we present LexiExp, a framework for sentiment lexicon expansion.

LexiExp[3]: Lexical Expansion for Sentiment Lexicon

Lexical expansion is an unsupervised technique that is based on the computation of distributional thesaurus [21]. A distributional thesaurus lists semantic neighbors, generated automatically from a corpus by finding the similarity of contexts in which the words occur [141]. It was best described by the famous quote of Firth (1957): "You shall know a word by the company it keeps". Lexical expansion can provide a useful back-off technique for rare words and unseen instances. Let us consider the following examples:

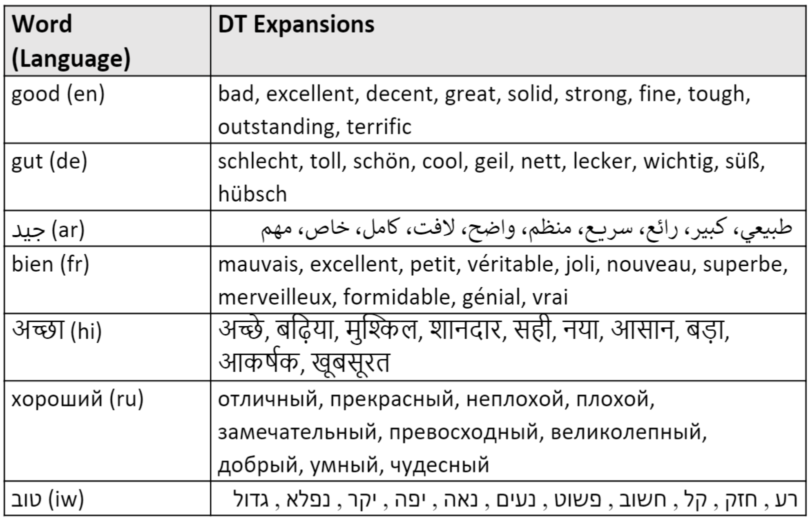
"The dvd player has a ***sleek*** *design and works fine."*
"I think this camera is ***splendid"***

The words "sleek" and "splendid" are hardly found in sentiment lexicons. Using lexical expansion, we retrieve the top 10 similar words to "sleek" and "splendid" together with their corresponding PoS tags as follows:

sleek: shiny/JJ, stylish/JJ, elegant/JJ, compact/JJ, fancy/JJ, smooth/JJ, sporty/JJ, vintage/JJ, compact/JJ, luxurious/JJ.
splendid: spectacular/JJ, exquisite/JJ, elegant/JJ, stunning/JJ, beautiful/JJ, superb/JJ, sumptuous/JJ, glorious/JJ, dazzling/JJ, wonderful/JJ.

The expansion list of each word indicates that both words describe a positive sentiment polarity. From the examples above and our extensive expansion results analysis, we have made two assumptions about the resulted expansion output of sentiment words. First, sentiment words are similar to other sentiment words with similar PoS tag (mostly adjective). Second, words tend to be similar to more words from the same sentiment, since words may

[3] https://github.com/uhh-lt/LexiExp



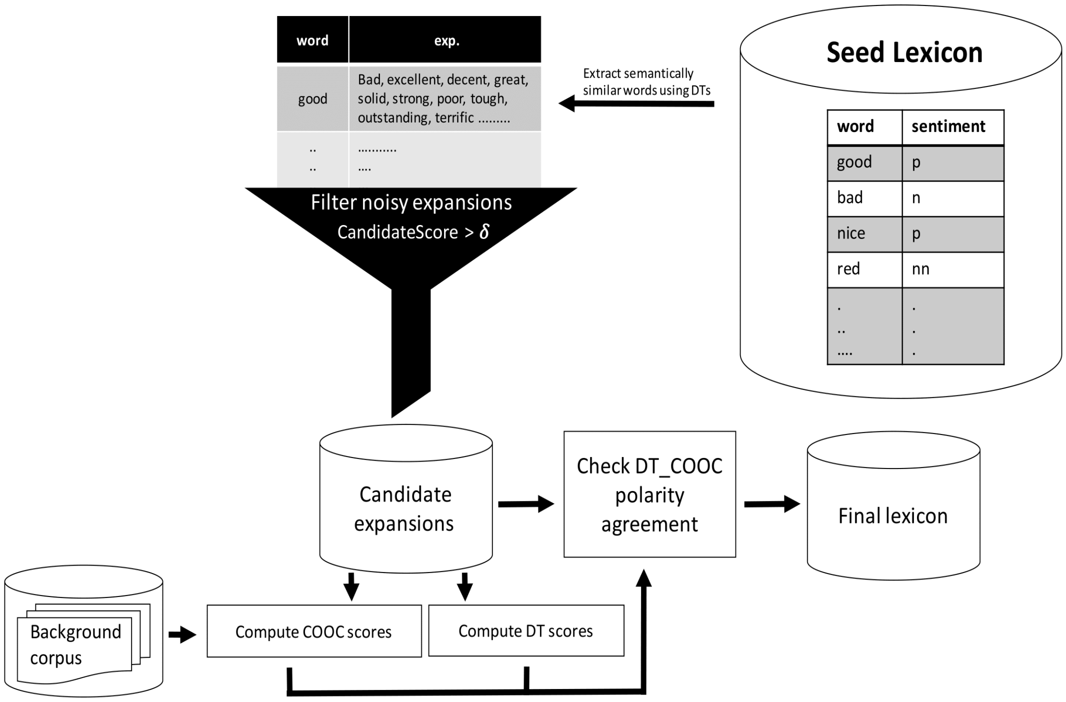
Word (Language)	DT Expansions
good (en)	bad, excellent, decent, great, solid, strong, fine, tough, outstanding, terrific
gut (de)	schlecht, toll, schön, cool, geil, nett, lecker, wichtig, süß, hübsch
جيد (ar)	طبيعي، كبير، رائع، سريع، منظم، واضح، لافت، كامل، خاص، مهم
bien (fr)	mauvais, excellent, petit, véritable, joli, nouveau, superbe, merveilleux, formidable, génial, vrai
अच्छा (hi)	अच्छे, बढ़िया, मुश्किल, शानदार, सही, नया, आसान, बड़ा, आकर्षक, खूबसूरत
хороший (ru)	отличный, прекрасный, неплохой, плохой, замечательный, превосходный, великолепный, добрый, умный, чудесный
טוב (iw)	רע , חזק , קל , חשוב , פשוט , נעים , נאה , יפה , יקר , נפלא , גדול

Table 4.1 DT Expansions of the word "good" in different languages. The sentiment word "good" is similar to sentiment words with the same polarity and PoS tag (Adjective).

also be distributionally similar to words from the opposite sentiment, such as ’good’ and ’bad’. More examples are given in Table 4.1. The table shows the expansion list of the word "good" in different languages.

We exploit the concept of lexical expansion and sentence-level co-occurrence from large background corpora to automatically expand existing (small) lexicons, and assign each new entry, which represents a sentiment word, a probability distribution over given polarity classes, indicating the likelihood of each sentiment polarity class. LexiExp is completely unsupervised, domain and language independent. Figure 4.10 shows the pipeline of LexiExp.

Our seed lexicon is formed by using a single hand-annotated lexicon like SentiWordNet, or by combining multiple lexicons of the same language. After constructing the seed corpus, for each word w in the seed set S, we obtain the top n most similar DT expansions $E_w = \{e_1, e_2, \ldots, e_n\}$. We employ an open-source implementation of the DT computation as described in [21], where complete details of the computation are explained. In the context of further use, we refer to E_w as "the expansion list" of word w, and each expansion list is assigned a polarity class $c \in C$, $C = \{c_1, c_2, \ldots, c_k\}$, depending on the sentiment polarity class of its corresponding seed word, $Pol(w)$, given in the original seed lexicon. For example, the DT expansion list of a seed word that has a positive sentiment polarity in the seed corpus, $Pol(”good”) = c_{positive}$, is called a positive expansion list and is assigned to the same polarity class, $Pol(E_{”good”}) = c_{positive}$. Throughout the explanation of LexiExp, we avoid to define



word	exp.
good	Bad, excellent, decent, great, solid, strong, poor, tough, outstanding, terrific ....
.. ..	....

word	sentiment
good	p
bad	n
nice	p
red	nn
. .. .	. . .

Fig. 4.10 LexiExp pipeline. The process of expanding existing small *seed lexicon* to a more comprehensive lexicon with higher coverage (*final lexicon*). LexiExp assigns each new entry, which actually represents a sentiment word, a probability distribution over given polarity classes, indicating the likelihood of each sentiment polarity class.

a specific sentiment classification scheme as some seed lexicons may classify sentiments differently, e.g., using emotions like anger, disgust, fear, happiness, sadness, surprise, etc.

The subsequent points describe the steps of LexiExp in more detail:

1. **Finding the candidate sentiment terms:** depending on the used background corpus, lexical expansion output may include noisy expansions that do not reflect sentiment meaning, so we start first by filtering out the resulted expansions and creating a final candidate expansions list. To filter out the candidate terms (sentiment words) from the noisy tokens, we rank each term t in the complete expansion according to its candidateScore.
$$candidateScore(t) = \frac{\sum\limits_{w \in S} [t \in E_w]}{Tf(t, DT\_corpus)} \tag{4.1}$$
where $Tf(t, DT\_corpus)$ is the frequency of the term t in background corpus used for the DT computation. Highly frequent words, which occur in so many contexts, will appear in every expansion list. To ensure that these words are not selected for the next step, we set a threshold δ. Only terms with $candidateScore > \delta$ are kept as candidate terms for the final lexicon generation.

2. **Calculating the probability distribution of polarity classes:** as we explained earlier, classifying expansion words directly into the polarity of its original seed words is illogical, since some expansions may belong to the opposite sentiment. In this step, we classify each candidate expansion into its correct sentiment polarity class based on the known sentiment of its distributionally similar words. A candidate sentiment word is expected to occur more often in a document near words from the same sentiment polarity, as well as in their expansion lists. We express these assumptions in mathematical form in Equations 5.6 and 5.7.
$$COOC_c(t) = \left| \frac{\sum\limits_{\substack{w \in S \\ Pol(w)=c}} count(t, w)}{\sum\limits_{w \in S} count(t, w)} \right| \tag{4.2}$$
$$DT_c(t) = \left| \frac{\sum\limits_{\substack{w \in S \\ Pol(w)=c}} [t \in E_w]}{\sum\limits_{w \in S} [t \in E_w]} \right| \tag{4.3}$$
$COOC_c(t)$ and $DT_c(t)$ act as membership functions, which measure the degree of membership of a candidate sentiment word t to a polarity class c based on two indicators:

(1) the proportion number of times the word t appear nearby sentiment words with sentiment polarity class c, (2) the proportion number of times the word t appear within the expansion lists of sentiment words with sentiment polarity class c. Based on a large background corpus, the *count* function returns the sentence-level co-occurrence of a candidate word t with seed word w. $COOC_c(t)$ and $DT_c(t)$ scores are each bounded between 0 and 1, and further sums up to 1.0 over all polarity classes for one candidate sentiment word, $\sum_{c \in C} COOC_c(t) = 1.0$, $\sum_{c \in C} DT_c(t) = 1.0$.

3. **Classify candidate sentiment words:** based on Equations 5.6 and 5.7, each candidate word is assigned two polarity classes, c_{DT} and c_{COOC}, according to the following equations:

$$c_{COOC} = \operatorname*{argmax}_{c \in C} COOC_c(t) \tag{4.4}$$

$$c_{DT} = \operatorname*{argmax}_{c \in C} DT_c(t) \tag{4.5}$$

Out of all classes in C, each equation returns the one class that has the maximum probability.

4. **Generating the final lexicon:** To construct the final lexicon, we consider the agreement between c_{DT} and c_{COOC}. Words are added to the final lexicon if, and only if, both c_{DT} and c_{COOC} return on the same polarity class:

$$\mathscr{L}(t) = \begin{cases} \textit{add to final lexicon}, & \text{if } c_{COOC} = c_{DT} \\ \textit{discard}, & \text{otherwise} \end{cases} \tag{4.6}$$

If t is added to the final lexicon, both $COOC_c(t)$ and $DT_c(t)$ of the class with maximum probability will be added to the final lexicon as well.

In principle, the expansion procedure can be iterated to bootstrap sentiment lexicons where the output of one step can serve as an input of the next expansion step. It should be emphasized that the size of the final induced lexicon depends on two factors: (i) the number of words in the seed lexicon that have expansions in the DT corpus and (ii) the pruning threshold δ. Absence of expansions for some seed words and a higher value of δ reduce the size of induced lexicon.

4.4 Supervised Models for ABSA

This section describes our supervised models for Aspect-Based Sentiment Analysis (ABSA).

4.4.1 Extracting Opinion Target Expression

The OTE is defined by its starting and ending offsets. Our supervised methodology for the OTE extraction task deals with it as a sequence labeling task. When dealing with text, the goal of sequence labeling is to assign a categorical label to each word in a sentence. Following the standard BIO notation [164], each word in a sentence is assigned to one of three labels, 'B-ASP', 'I-ASP' or 'O', depending on whether the token positioned at the beginning, intermediate or outside of an OTE respectively. For example, in the review sentence, "*The beef (B-ASP) steak (I-ASP) was (O) rather (O) dry (O) . (O)*", 'beef steak' is the OTE. When there is no explicit mention of the entity, the OTE is set to "NULL".

One of the most successful methods for performing sequence labeling is that of employing Conditional Random Fields (CRFs) [90]. CRFs main strength lies in the fact that it is a discriminative probabilistic model. Unlike generative models, discriminative models represent arbitrary feature-vector representations rather than spatial relations and co-occurrences. CRFs can easily incorporate non-dependent features of a sequential text, such as capitalization, suffixes, prefixes and adjacent words, which are important when dealing later with words unseen in training set.

For the OTEs extraction subtask, we focus upon the features that characterize the surface word form, structural and syntactic properties of the text as follows:

- Current token and surrounding context: this feature represents the lowercase strings of the current token and its surrounding context in a window of [-2..2] (2 preceding tokens from the left and 2 succeeding tokens from the right of the current token). Using this feature would appear to be straightforward, however, it provides a good indicator of the different OTE types, especially when the training set has few examples of a specific OTE type in a certain domain. To adjust the window size, we rely on the experiments by Brun et al. [31]. They found that using a window of [-2..2] leads to better results than using a larger feature window size, which they proved have caused an overfitting behavior.

- Identified-as-OTE flag: based on the OTEs list identified by our unsupervised DDG-based OTE identification approach (see Section 4.3.1), we create a binary feature that checks whether the current token already exists identified OTEs list or not.

- Character 5-grams: writing style features like word and character n-grams features, often incorporated in stylometry research, have also shown to be effective in sentiment analysis [1]. They are also commonly applied to non-formal texts and user-generated content. We extracted all the possible sub-strings of 5 continuous characters-grams from the current token.

- Part-of-Speech (PoS) tags: this feature represents the PoS tags of the current token and its surroundings. Using the pre-trained unsupervised PoS tagging models by Biemann [19], we obtain the PoS tags of the current token and its surrounding tokens in a window of [-2..2]. This feature has three-fold advantages: (1) it provides a powerful mechanism for lexical disambiguation for tokens that have different senses with different PoS tags in different contexts, (2) gives a good indication of relevant OTEs since many researchers have experimentally shown that nouns make good candidates for OTEs [70, 179], (3) it helps in extracting multiword OTEs (i.e., noun compounds).

- Head word and its PoS tag: we extract the from the head word from the sentence parse tree and use it as a feature along with its PoS tag information. This feature can be a good indicator of the semantic category of the phrase.

- Word affixes (prefixes and suffixes): we use the prefixes and suffixes of the current token of length up to four characters. Previous research in sentiment analysis has used these features to handle negations. The presence of negation in an opinionated sentence implicates, in most cases, the existence of an OTE [44].

- Frequent nouns: we build a list of frequently occurring nouns from the training set. A noun is considered to be frequent if it appears at least four times in the training corpus. We define a binary feature to indicate whether the current token exists in the list or not.

- Dependency relations: similar to [165], we extract two features: the first contains the relation strings, restricted to "amod", "nsubj" and "dep", where the current token is the *governor* (i.e., head) of the relation, while the second feature contains the relation strings, restricted to "nsubj", "dobj" and "dep", where the current token is the *dependent* of the relation. Each set of strings is used as a feature value for the current token, resulting in two separate features. The purpose of these features is to (1) encode structural information, and (2) indicate whether the current token is involved in any grammatical relations with an opinion word.

- Word shape: this is a binary feature, which checks whether the first letter in the current token is capitalized or not. Capitalized words are candidate names of entities (foreign food dishes, restaurants, brands hotels etc.).

- DT expansions: we expand the current token to its most similar words using distributional thesaurus (DTs) [21], and use the top 5 as features, Table 4.2 presents some examples. Incorporating lexical semantic features may bring some important hidden information, which may improve the extraction of rare aspects.

Token	DT Expansion
drinks	beers, wines, coffee, liquids, beverage
price	prices, pricing, cash, cost, pennies
fresh	fresher, new, refreshing, clean, frozen
laptop	pc, computer, notebook, tablet, imac
toshiba	samsung, sony, acer, asus, dell
touchpad	mouse, trackball, joystick, trackpad

Table 4.2 Example of DT expansions for frequent aspects.

- Sentiment score (using LexiExp): OTEs have sentiment words around them. Based on our expanded lexicons, we calculate the average sentiment score of surrounding sentiment words in a window of [-2..2] (preceding 2 and following 2 tokens of the current token). This feature shall help the model to distinguish between opinionated sentences and non-opinionated sentences.

Due to the lack of some tools and resources for non-English languages, we extract the following features only for English texts:

- Word cluster: this feature was created using the Brown clustering algorithm [30]. Brown clustering is a data-driven hierarchical clustering algorithm, which clusters words based on the contexts in which they occur (i.e., groups words to maximize mutual information between adjacent words). We used a precompiled clustering model provided by Owoputi et al. [125], which is commonly used in sentiment analysis literature. They induced 1000 hierarchical clusters using approximately 56 million English tweets. The current token is assigned to its cluster in Brown's dictionary, and the induced cluster is included as a feature. Semantic features as Brown clusters can give a rich representation, which can be useful for reducing the sparsity [68].

- Chunk tags: while PoS tagging shows OTEs at word-level; however, a review can often have multiword OTEs (noun phrases), such as "battery life", "spicy tuna rolls", etc. Text chunking is used to recognize the simple syntactic structure of sentences. In our case, it helps in identifying the boundaries of multiword aspect terms. We use the chunk tags of the current token and surrounding context in a window of [-1..1] as features.

- Lemma: lemmatization is a word normalization technique that trims the inflectional forms of a word to a common root form called lemma. We add the lemmatized form of the current token to our feature set.

- WordNet synsets: we use the top 4 noun synsets of the current token from the WordNet [106] as features.

- Named-Entity sequence: if the current token is a part of a named entity, we use the NER-sequence labels in BIO-scheme as features.

- SentiWordNet sentiment score: SentiWordNet [54] is one of the most popular sentiment lexicons for the English language. We calculate the average sentiment score of sentiment words in the surrounding context of the current token within a window of [-2..2].

4.4.2 Aspect Category Detection

The SemEval shared task first released the aspect category detection subtask in 2014. Given a predefined inventories of entity types E (e.g., LAPTOP, MOUSE, RESTAURANT, FOOD) and attribute labels A (e.g., DESIGN, PRICE, QUALITY), the goal of this task is to identify every entity E and attribute A pair (E#A) toward which an opinion is discussed in a given review sentence. Aspect categories are typically coarser than OTEs, and they do not necessarily need to be mentioned explicitly in the text. For example, in "crowded and expensive", the text has no explicit indication about the actual aspect category RESTAURANT#PRICE, but can be inferred through the adjectives 'crowded' and 'expensive'. To identify aspect categories, we deal with the problem as a multi-label classification problem at sentence-level where the classification model outputs a probability for each of the predefined categories (i.e., pairs). The category is selected if its probability exceeds a predefined threshold. We select a threshold that maximizes the F1 score during the training/validation phase. Users in microblogs can refer to a certain entity type or attribute using different terms in their reviews. In this case, domain-specific, and semantic features are very effective. The following features are extracted at sentence-level and used for training the classifier:

- Topic with the highest probability: using a pre-computed topic model, each review sentence is assigned to a particular topic with a certain probability. We associate each sentence to the topic with the highest probability and integrate it into the features list.

- DDGs top domain words: we use in-domain data to construct DDGs, as we explained in Chapter 2. As a result, a DDG is constructed for each entity type. It includes distinctive terms related to the domain, as shown in Figure 4.2. For each entity, we extract the top five words with the highest $Tf\text{-}Idf(w_{ij}, d_i, D)$ weights. Binary features denoting the presence or absence of each word in a given review are used as features.

- Distributional thesaurus: for each of the top five significant words from the previous point, we find ten most similar words using DTs, see Table 4.2. Features indicating the presence or absence of each expansion in a given sentence are added to the feature vector.

- Bag-of-Words: this feature denotes the frequency of each word in a given review sentence.

- DDGs-based identified OTEs list: we add a set of binary features that indicates the absence/presence of each identified OTE in a given review sentence.

4.4.3 Sentiment Polarity Classification

The polarity classification task can be treated as a supervised multiclass classification problem. Each given OTE has to be classified into one of three polarity classes: "positive", "negative" or "neutral". For this subtask, we rely mostly on lexicon-based and n-gram features. A classification model is trained using the following features:

- Target OTE and surrounding context: we use the target OTE, which is to be classified, as a feature. Since polarity orientation of a given OTE relies heavily on the context where it appears, we also include the OTE's surrounding tokens in a window of [-2..2]. Binary features indicating the presence or absence of each token are added to the feature vector.

- Sentiment word and sentiment scores (LexiExp): based on our expanded lexicon, we extract the nearest sentiment word to the target OTE, its sentiment polarity, and its associated $COOC_c(t)$ and $DT_c(t)$ scores, and use it as features.

- Entity-Attribute (E#A) pair: we add the aspect category in which the sentence was classified into previously, as we explained in Section 4.4.2, as a feature. This feature is exclusively extracted for the SemEval task experiments, since it is the only dataset that has E#A annotations.

- DDGs domain features: we use both "topic of highest probability" and "DDGs top 5 domain words" features from the previous subsection. These two features shall increase the ability of our model to accurately determine the correct polarity when a sentiment word is domain-dependent, e.g., "long battery life" in electronics domain vs. "long waiting time" in restaurant domain.

We also experimented short text polarity classification to classify tweets into positive, negative and neutral categories. In this case, we use different set of features:

- Word and character n-grams: word unigrams and bigrams are extracted for each term in a given tweet. For unstructured short texts like tweets, small values for n have shown to be most effective [60]. We also compute n-grams at character-level on the basis of character trigrams and quadgrams.
- Sentiment lexicon features: since we are dealing with a sentiment polarity classification task, the majority of features used are based on our expanded sentiment lexicons:
  - Sentiment words per polarity class: for each tweet, we compute the sum of sentiment words defined in our expanded lexicon that convey one sentiment class. We repeat that for all sentiment polarity classes.
  - Sentiment scores: additionally, we calculate the average $DT_c(t)$ score and average $COOC_c(t)$ score for all sentiment words in a given tweet, which are defined in the expanded lexicon.

4.5 Experimental Results and Evaluation

We evaluate our ABSA approaches on four different datasets from different sources, namely Amazon customer reviews dataset [100], Hu and Liu customer reviews benchmark dataset [70], the SemEval-2016 ABSA benchmark dataset [132], sentiment analysis for Indian languages (SAIL) subtask benchmark dataset [128]. In the following subsections, for each of the four datasets, we explain the preparation and preprocessing pipeline, explain the experimental setup, and discuss the obtained results.

4.5.1 Aspect Identification from Amazon Corpora

An implementation of our unsupervised approach (Section 4.3) is applied to identify and extract product aspects from a large set of Amazon product reviews. We use an unlabeled version of Amazon dataset[4], which has been commonly used in opinion mining research [82, 167]. The corpus consists of ∼35 million reviews (∼18.4 million unique reviews), of ∼2.5 million products from 28 different categories, up to March 2013. Reviews include products' and users' information, ratings, and reviews plain text [100]. In this work, we only use the plain text. We filter out non-English reviews, redundant reviews, reviews with less

[4] SNAP Web data: Amazon reviews https://snap.stanford.edu/data/web-Amazon.html

than three words and noisy reviews which contain only emojis or punctuation, as we consider these irrelevant for aspect identification. The final number of reviews we use to train the LDA model is ∼13.93 million reviews. We experimentally determined a reasonable number of topics to be 200, which is in line with other works using LDA for information extraction, e.g., [35]. We use the resulted LDA model to select topically pure reviews (Section 2.2.2). The total number of remaining in-domain reviews is ∼1 million. Of the 200 topics we induced with LDA, we observed a large number of product-specific topics, as well as some mixed topics and spurious topics [107]. We then perform sentence segmentation followed by POS tagging and dependency parsing [99]. The output from this step is important for generating the syntactic features which will be used later to filter DDGs and extract topically pure relations.

To construct DDGs and distinguish most related domain-specific words and relations, we filter out words and relations below thresholds α_1, α_2, β_1, β_2, λ_1, λ_2. Figure 4.2 illustrates a snapshot from the DDG that captures camera product reviews. We list below some words from the camera's domain, categorized by PoS tags, to illustrate the role of Tf-Idf weighting in capturing potential domain-specific words. It can be seen that all the listed words are strongly related to the camera domain.

- **Adjectives:** digital, 50mm, focal, 200mm, optical, sharp, indoor, blurry, wide, prime, compact, chromatic.
- **Nouns:** lens, camera, canon, nikon, SLR, EF, shots, shutter, USM, telephoto, aperture, macro, flash, sigma, focus, pictures, zoom, tripod, powershot.
- **Verbs:** taking, focuses, capture, shoot, photographing, zoomed, produce, cropping, adjust, purchase.
- **Adverbs:** dimly, flawlessly, steadily, sharply visually, clearly, optically, professionally.

We apply a set of appropriate rules to extract OTE-sentiment pairs, as discussed in Section 4.3.1. We highlight some of these OTE-sentiment pairs in Table 4.3. The table shows the dependency relation type R_{Camjk}, source word w_{Camj}, destination word w_{Camk}, relation frequency *Tf* and relation-level *Tf-Idf*. Similarly, we create DDGs for another 14 topics, including movies, coffee makers, electro-voice, shoes and footwear, hair products, food and baking machines, films, mp3 players, cars, TVs, mobiles, computers and perfumes. We will refer to these DDGs again throughout the subsequent subsections.

To evaluate the performance of our unsupervised OTE identification approach, we compare the OTEs identified by our DDG-rule-based approach to those obtained by DDG without weighting (i.e., $\alpha_1 = \beta_1 = \lambda_1 = 0$) by keeping only "amod" and "nsubj" relations. We evaluate

R_{Camjk}	w_{Camj}	w_{Camk}	***Tf***	***Tf-Idf***
amod	lens	fast	146	770.60
nsubj	great	lens	121	638.65
amod	picture	good	205	467.88
amod	images	sharp	116	451.45
nsubj	sharp	images	93	388.69
amod	photos	great	105	269.85
amod	picture	clear	84	241.93
nsubj	good	quality	142	50.85

Table 4.3 Examples of opinion dependency relations from the camera topic. From left to right headers represent dependency relation type R_{Camjk}, source word w_{Camj}, destination word w_{Camk}, relation frequency *Tf* and relation-level *Tf-Idf*.

the OTEs manually by human judgment. We order the identified relations from both approaches according to relation frequency. For the top 50 unique identified OTEs per product entity, we check the number of OTEs that are relevant to a particular product entity.

Table 4.4 shows the experimental results of five different product entities. Results show that our DDG-rule-based approach outperforms the baseline frequency-based (FB) approach in terms of accuracy, and has not been worse in any case. The unsupervised DDG-rule-based approach with Tf-Idf weighting identifies domain-specific OTEs with an average accuracy of 53.2% across the five topics, when the FB approach achieves only 37.2%. The FB baseline tends to identify general OTEs, such as price, shipping, quality, value, service and company. Weighting DDGs by means of Tf-Idf gives our method the ability to detect detailed domain OTEs, which is clearly evident in the car's topic.

Error analysis shows that most false positives by the Tf-Idf-based method consist of product domain-specific words that are not OTEs. Examples from the cameras domain are: fast results, great job, cheap camera, excellent choice, sharp razor, perfect bag, great portrait, advanced photographer, easy c330. On the other hand, frequency-based ranking provides general noisy errors and common nouns that are not related to the product like: problem only, buy great, complaint only, time hard, addition great, drawback only, light available, room enough.

To evaluate the identified OTEs coverage, we selected 50 recent reviews about cameras from the Amazon website and manually annotated the explicit OTEs in the text. Implicit aspects are not considered. The reviews are selected for their high density of OTEs. We compared the annotated aspects against the 33 cameras OTEs listed in Table 4.4. Out of 183 annotated OTEs in the 50 reviews, 115 OTEs are extracted, approximately 63%. Most of the missed aspects are present in the cameras DDG before filtering. Changing the filtering

Category / Parm.	**Method**	**Acc.**	**Extracted Aspects**	
			Common	**Difference**
Camera α_1: 100, α_2:180 β_1: 2, β_2: 2 λ_1:7, λ_2:5	Tf-Idf-based	0.60	lens, pictures, shots, quality, images, photos, focus, light, depth, color, zoom, size, weight, range, distortion, card, autofocus, speed.	tripod, resolution, controls, battery, mode, contrast, optics, flash, sharpness, software, screen, flexibility, distance.
	FB	0.40		price, value, capability.
TV α_1: 50, α_2:20 β_1: 1, β_2:1 λ_1:2, λ_2:5	Tf-Idf-based	0.44	cable, picture, quality, remote, setup, image.	system, audio, resolution, output, video, tuner, hdtv, quality, connection, capability, control, speakers, screen, model, component, connector.
	FB	0.26		price, sound, value, shipping, colors, monitor, pixels.
Computer α_1:150, α_2:50 β_1: 2, β_2:2 λ_1:2, λ_2:5	Tf-Idf-based	0.58	card, software, memory, adapter, performance, setup, support, camera, driver, ram, disk, space, cable.	upgrade, programs, ports, system, processor, speed, motherboard, version, machine, units, USB, slots, OS, mouse, graphics, interface.
	FB	0.38		price, power, value, quality, shipping, case.
Mobile α_1: 50, α_2:20 β_1: 1, β_2:1 λ_1:5, λ_2:1	Tf-Idf-based	0.40	sound, keyboard, screen, price, reception, weight, quality, size, case, camera, service, software.	pictures, apps, life, interface, looks, speakerphone, bluetooth, battery, version, calls.
	FB	0.36		card, program, version, design, charger, player, value.
Cars α_1: 20, α_2:5 β_1: 2, β_2:1 λ_1:5, λ_2:1	Tf-Idf-based	0.64	price, performance, exhaust, wiring, plugs, installation, power, length, kit, sound, shocks, sensors, ride, instructions, parts.	work, rumble, breaks, pads, muffler, replacement, wipers, harness, connectors, idle, engine, hitch, system, unit, lights, mileage, tensioner.
	FB	0.46		quality, shipping, value, struts, company, service, look, room.

Table 4.4 Manual evaluation of OTEs identification on five different domains using DDG-rule-based approach and frequency-based (FB) approach. Performance is measured in terms of accuracy (percentage of correctly identified OTEs in the top 50 frequent relations). The table shows also common identified aspects along with the difference between the two methods. The first column shows the product entity name along with the weighting thresholds values. For the frequency-based ranking method, $\alpha_1 = \beta_1 = \lambda_1 = 0$.

	Training			**Testing**		
Language	Reviews	Sentences	Opinions	Reviews	Sentences	Opinions
	Restaurant					
EN	350	2000	2507	90	676	859
DU	300	1711	1860	100	575	613
ES	627	2070	2720	268	881	1072
FR	355	1733	2530	120	694	954
RU	312	3548	4089	103	1209	1300
TR	300	1104	1535	39	144	159
	Phones					
DU	200	1389	1393	70	308	396
	Laptop					
EN	450	2500	2909	80	808	801
	Hotels					
AR	1839	4802	10509	452	1227	2604

Table 4.5 Statistical information of SemEval-2016 Subtask 1 training and testing datasets that we used in our experiments. The table shows the number of reviews, sentences and opinions per domain per language.

parameters can help to increase the aspects coverage but may also increase the false positive rate.

In summary, our evaluation shows a definite improvement using Tf-Idf-based filtering over the FB baseline. This, however, is only possible for mixed-domain document collections, as Idf for a single topic is inconsequential.

4.5.2 ABSA SemEval Shared Task 2016

In an attempt to support the development of Aspect-Based Sentiment Analysis (ABSA) research techniques and applications, the SemEval shared tasks ABSA [134, 133, 132] offers the opportunity to experiment and evaluate on benchmark datasets (manually annotated reviews) across various domains and languages through different subtasks. In this section, we present a systematic evaluation of the proposed supervised and unsupervised ABSA approaches conducted on the ABSA SemEval-2016 benchmark datasets of Subtask 1. Subtask 1 covers the three ABSA sub-problems[5] we mentioned earlier in this chapter, namely aspect category detection (Slot 1), Opinion Target Expression (OTE) extraction (Slot 2), and sentiment polarity classification (Slot 3). Statistics on the SemEval-2016 Subtask 1 training and testing sets used in our experiments are shown in Table 4.5. The table shows the number of reviews, sentences and opinions in each domain per language. Since we are analyzing

[5] The SemEval organizers use the term "slot" to refer to each subtask.

sentiment in aspect-level, the number of opinions is always higher than the number of sentences, which means, one sentence may discuss more than one aspect.

Since the original data is provided in XML format, we start first by removing XML tags and extracting the needed text and annotations. We tokenize the text, normalize all digits to 'num', and remove stop words and punctuation for the purpose of Tf-Idf computation. We also perform lemmatization, text chunking, Named Entities (NE) extraction, part of speech tagging and syntactic parsing for each sentence. For English text, the above preprocessing pipeline is implemented using the Stanford CoreNLP [96] toolkit[6]. For other languages, we trained the Mate-Parser[7] [26] on Universal Dependencies (UD) treebanks[8] [123] of different languages. Due to resource limitations, lemmatization, text chunking and NE extraction were not performed for non-English text reviews. Since the domains are already given, we skip the step of topic modeling and directly construct DDGs from the parsed text, and weigh the graphs using the given Tf-Idf parameters. For languages where datasets from different domains are provided, Tf-Idf computation is straightforward. However, for French and Spanish, the provided datasets are only from the restaurant's domain. Thus, it was necessary to use external reviews from different domains to compute tf-idf. We use movies reviews for Spanish [9]; books, music and DVD reviews for French[10] [136].

For all languages, lexicons used for sentiment polarity classification are constructed by expanding external seed sentiment lexicons. We use AFINN[11] [121], NRC Hashtag [109], Sentiment 140 [82], NRC Emotion[12] [110] and Bing Liu [70] combined, as a seed lexicon for English; Salameh et al.'s [145] Arabic version of Bing Liu's lexicon [70] for Arabic; VU sentiment lexicons[13] [95] for French, Dutch and Spanish; a lexicon by Panchenko [127] for Russian; and a combination of SentiTurkNet [49] and NRC Emotion [110] for Turkish.

As a background corpora for topic modeling, DT computation and co-occurrence analysis, we use the following resources: Amazon reviews corpus [100] for English[14], The WaCky Wide Web corpus[15] [15] for French, COrpora from the Web corpora[16] [147] for Spanish and

[6] Stanford CoreNLP Tool: https://stanfordnlp.github.io/CoreNLP/

[7] Mate-Parser https://code.google.com/archive/p/mate-tools/

[8] Universal Dependencies: http://universaldependencies.github.io/docs/

[9] Corpus Cine (Spanish cinema): http://www.lsi.us.es/~fermin/index.php/Datasets

[10] Originally crawled from Amazon: https://webis.de/data/webis-cls-10.html

[11] AFINN Sentiment Lexicon: http://corpustext.com/reference/sentiment_afinn.html

[12] NRC Emotion Lexicon: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[13] VU-sentiment-lexicon: https://github.com/opener-project/VU-sentiment-lexicon

[14] https://snap.stanford.edu/data/web-Amazon.html

[15] Linguistically processed web-crawled corpora: http://wacky.sslmit.unibo.it/doku.php?id=corpora

[16] COrpora from the Web: http://corporafromtheweb.org

Language	Seed Lexicon			Induced Lexicon	Common Entries
	Positive	Negative	Neutral		
English	2005	4789	-	12953	4120
Dutch	3314	5923	-	8496	2992
French	9338	10339	5993	18308	7636
Spanish	2175	1737	7869	12480	4306
Russian	3217	8849	-	7697	2945
Turkish	1900	2515	1382	6547	1838
Arabic	1916	4467	-	9077	1447

Table 4.6 Expansion statistics of induced lexicons. Common entries denote the number of words that are present both in the seed lexicon and the induced lexicon.

Dutch, Library Genesis ebooks for Russian[17], and Leipzig Corpora Collection[18] [62] for Arabic.

We obtain the top n =100 DT expansions of each word in the seed lexicon and keep candidate terms that have a candidateScore > 0.2 ($\delta = 0.2$). For each candidate term t, we compute the normalized $COOC_c(t)$ and $DT_c(t)$ scores for $c \in \{positive, negative, neutral\}$ and generate the final lexicon by checking $\mathscr{L}(t)$ for each term. Expansion statistics for the induced lexicons are provided in Table 4.6.

The supervised implementation of Slot 2 is provided by the CRFsuite tool[19] [124]. CRFsuite provides a fast training and tagging implementation of CRFs and supports the cross-validation evaluation scheme. As a classifier for Slot 1 and Slot 3, we use Support Vector Machine (SVM), in particular Liblinear implementation [20] [43, 55]. SVM is widely used in NLP research. Fast linear SVM methods can cope well with a large number of nominal features and can still achieve accuracy close to that of using highly nonlinear kernels [37]. The CRF model and the SVM classifiers were trained using the features described earlier in the three subsections of Section 4.4. For feature selection and hyperparameter tuning, we perform 5-fold cross-validation over the training set. For aspect category detection experiments, we set the classification probability threshold to 0.185, 0.145, 0.130, 0.180 for restaurants, phones, laptops and hotels, respectively, during the evaluation phase.

We participated in Slot 1 and Slot 3 for English, Spanish, Dutch, French, Turkish, Russian and Arabic languages spanning four domains: restaurants, laptops, phones and hotels. We also conducted experiments on Slot 2 for English, Spanish, Dutch and French in the restaurant

[17] lib.ruc.ec

[18] Leipzig Corpora Collection: http://corpora2.informatik.uni-leipzig.de

[19] CRFsuite Software: http://www.chokkan.org/software/crfsuite/

[20] LIBSVM: https://liblinear.bwaldvogel.de/

domain. For testing, we use the dataset provided by the task organizers. The results of the supervised and unsupervised ABSA approaches are presented in Table 4.7 and Table 4.8, respectively.

Model	EN/Rest.	DU/Rest.	ES/Rest.	FR/Rest.	RU/Rest.	TR/Rest.	DU/Ph.	EN/lap.	AR/Htls
Slot 1: Aspect Category Detection (E#A) - F1									
Baseline	59.92	42.81	54.68	52.60	55.88	**58.89**	33.55	37.48	40.33
All-DDGs	60.10	53.87	58.81	54.81	58.50	56.54	44.01	40.32	46.80
All-OTEs	61.22	54.98	59.06	57.03	58.19	55.70	45.04	42.60	47.30
All Feat.	**63.05**	**55.42**	**59.90**	**57.50**	**62.70**	56.63	**45.44**	**43.91**	**48.90**
Best	73.03	60.15	70.59	61.21	64.83	61.03	45.55	51.94	52.11
Rank/#Par	17/30	3/6	6/9	2/6	3/7	3/5	2/4	12/22	2/4
Slot 2: Opinion Target Expression (OTE) - F1									
Baseline	44.07	50.64	51.91	45.45	–	–	–	–	–
All-LexiExp	66.90	63.37	68.97	68.90	–	–	–	–	–
All-DTs	61.22	54.98	59.89	57.033	–	–	–	–	–
All Feat.	**68.45**	**64.37**	**69.73**	**69.64**	–	–	–	–	–
Best	72.34	64.37	69.73	69.64	–	–	–	–	–
Rank/#Par	2/19	1/3	1/5	1/3	–	–	–	–	–
Slot 3: Sentiment Polarity Classification - Accuracy									
Baseline	76.48	69.33	77.80	67.40	71.00	72.32	80.80	70.03	76.42
All-E#A	86.22	76.25	82.92	71.90	73.16	83.65	82.06	82.31	80.65
All-LexiExp	86.39	74.22	79.58	70.15	70.65	80.78	80.89	82.45	78.68
All Feat.	**86.72**	**77.00**	**83.58**	**72.22**	**73.61**	**84.27**	**82.57**	**82.77**	**81.72**
Best	88.12	77.81	83.58	78.82	77.92	84.27	83.33	82.77	82.71
Rank/#Par	2/29	2/4	1/5	5/6	3/6	1/3	2/3	1/22	2/3

Table 4.7 Evaluation results of our supervised approach in terms of F1 measure (for aspect category detection and OTE extraction), and accuracy % (for sentiment polarity classification). Feature ablation, baseline system and best performing systems results are provided. The last row on each slot shows the ranking of our system over all participants.

The task organizers provided the baseline results. For Slot 1 and Slot 3, the baseline is simply a bag-of-words-based SVM classifier. For Slot 2, a list of frequent OTEs is created for each aspect category using the training data. Then, given a test sentence and an assigned category, the baseline checks the occurrences of OTEs from the given category list in the given sentence.

All Feat. rows contain the results of all-feature combination, which in all cases yields the best performance across all Slots. Our results outperform the baseline provided by the task organizers by a large margin. This, however, does not hold for the Turkish restaurants/Slot 1 results, where we perform lower than the baseline. We suspect that this is due to the limited test set size (144 sentences). We achieve comparable results to those obtained by best-performing systems in Slot 2 and Slot 3. In fact, our system is placed first in Slot3 for the laptop's domain in English, restaurant's domain in Spanish and Turkish; and Slot 2 for restaurant's domain in Dutch, Spanish and French. We score in medium to high ranks for

Dataset	**EN/ Rest.**				**DU/ Rest.**			
Slot	**Slot 2**			**Slot 3**	**Slot 2**			**Slot 3**
	P	**R**	**F1**	**Acc%**	**P**	**R**	**F1**	**Acc%**
Unsup.	64.35	58.99	61.55	81.74	65.40	50.10	56.73	72.68
Unsup.-LexiExp	-	-	-	79.21	-	-	-	70.44

Table 4.8 Experimental results using our unsupervised approach. The second row shows the results of sentiment polarity classification when using available lexicons "as is" without applying lexicon expansion.

Slot 1. We can explain this as due to the misclassification of fine-grained aspect categories (e.g., Restaurant#Prices, Food#Prices, Drink#Prices). Slot 1 results are worse for the laptops reviews. We believe this can be attributed to the large number of possible categories (18 categories) in this domain.

Table 4.7 also shows how the different features we introduce, in particular, DDGs and LexiExp features, affect the results of our supervised ABSA approach. For Slot 1, eliminating the DDGs and the OTEs features reduces the F1 score by an average of 2.2 and 1.4 respectively. It is obvious in this case that DDGs features are more influential since they provide domain-specificity, which is important to distinguish the different aspect categories. Aspects obtained through DDGs also provide strong signals about the category class. For example, if the OTE is "staff", then the entity is most likely to be "SERVICE". Another interesting finding is the impact of the DTs features on Slot 2 results with an average of 9.8 F1 score reduction when being removed. We assume that this is due to its ability to capture the lexical variability of aspects. LexiExp features cause slightly increased performance, but not as much as DTs features. They represent a useful indicator of opinionated sentences. The effect of LexiExp features, however, is more noticeable in both supervised and unsupervised Slot 3 evaluation results. We get a significant improvement of 2.3 average increase in F1 score on adding information from the induced lexicons over all languages. The improvement is more evident for languages other than English, where existing sentiment lexicons are less comprehensive. Using entity-attribute pairs as features also help in resolving conflicting sentiments (e.g., cheap food (positive) to cheap service (negative)).

Our supervised approach provides better results than the unsupervised approach. We performed an error analysis on missed aspects by our unsupervised OTE extraction approach (Slot 2), as the recall is consistently lower than the precision. We observe that in most cases, the correct offsets (B-ASP and I-ASP) of the aspect are being misclassified due to: (1) strict matching that does not consider slight variations when comparing offsets, (2) parsing errors or sentences with complex structure.

4.5.3 Sentiment Analysis for Indian Languages

The majority of existing work in sentiment analysis is dedicated to processing languages such as English, German and French. Sentiment analyzers developed for such languages are not directly applicable to Indian languages, which have their own challenges concerning language constructs, morphological variation and grammatical differences. Sentiment Analysis in Indian Languages (SAIL) shared task [128] is the first attempt to bring together the researchers for resource creation and knowledge discovery in Hindi, Bengali and Tamil. Given a set of annotated tweets in Indian languages, the task is to classify whether the tweet is of positive, negative, or neutral sentiment. The distribution of classes of the given datasets is given in Table 4.9.

Considering the lack and scarcity of available sentiment lexicons of Indian languages, we leverage our lexicon expansion approach (LexiExp) to expand Indian sentiment lexicons. We show that by integrating new features based on our expanded lexicon, sentiment polarity classification performance improves. The experiments were performed using our supervised system with extracted features explained previously in Section 4.4.3.

Dataset	Positive Tweets	Negative Tweets	Neutral Tweets	Total
		Hindi		
Training Set	168 (13.75%)	559 (45.74%)	494 (40.46%)	1221
Test Set	166 (35.54%)	251 (53.74%)	50 (10.70%)	467
		Bengali		
Training Set	277 (27.73%)	354 (35.43%)	368 (36.83%)	999
Test Set	213(42.60%)	151 (30.20%)	135 (27.00%)	499

Table 4.9 Distribution of training and test sets for Hindi and Bengali language.

Twitter constraints tweet length to a maximum of 140 characters, so we kept our preprocessing to a minimum. The preprocessing pipeline consists of a chain of simple regular expressions scripts. We normalize URL links in all tweets to ’someurl’, @username mentions to ’someuser’, multiple white spaces are replaced with single white space, and dates, RT (retweet) tags and numbers are removed.

We use separate external background corpora of Hindi and Bengali from Leipzig Corpora Collection[21]. The Hindi corpus contains a total of 2,358,708 sentences (45,580,789 tokens), and the Bengali corpus includes 109,855 sentences (1,511,208 tokens). Both corpora are constructed from online newspapers from 2011. The task organizers provided Indian sentiment

[21] Leipzig Corpora Collection: http://corpora2.informatik.uni-leipzig.de

lexicons [46], which they called SentiWordNet for Indian. Each includes a list of positive, negative, neutral and ambiguous words, with the corresponding PoS tag of each word. We use the given Indian SentiWordNet for both the Hindi and Bengali languages as seed lexicons for lexical expansion. We obtain the top n =125 DT expansions of each word in the seed lexicon and keep candidate terms that have a candidateScore > 0.2 ($\delta = 0.2$). Statistics of the final generated lexicons are presented in Table 4.10.

Dataset	**Positive**	**Negative**	**Neutral**	**Total**
Hindi	5521	3926	48	9495
Bengali	7213	1461	30	8704

Table 4.10 Statistical information of expanded lexicons of Indian languages.

We perform our experiments using the Liblinear-implementation of the SVM classifier. To tune and evaluate our model, we perform 5-fold cross-validation on the training set. We use classification accuracy (%) as a measure of the sentiment polarity classification performance. Table 4.11 shows the final results using different combinations of features. We find that word n-grams are the most important features in both languages. They improve the classification accuracy by 2%-5%. The next most important features are the "word per class" count features, which yield to ~0.6%-0.8% accuracy improvement. However, we observe a drop in performance when using the DT and COOC scores, probably because the external dataset is from a different domain (formal news).

Model	**Hindi**				**Bengali**			
	P	N	Neu	Total	P	N	Neu	Total
All-Sent. Words/Class	-	-	-	47.32 (-0.64)	-	-	-	41.20 (-0.80)
All-W. n-grams	-	-	-	43.25 (-4.71)	-	-	-	38.40 (-2.80)
All-Ch. n-grams	-	-	-	47.75 (-0.21)	-	-	-	42.20 (+0.20)
All-Sent. Scores	9.04	**73.70**	64.0	49.68 (+1.72)	23.47	59.60	**56.30**	**43.20** (+1.20)
All	4.22	69.72	68.00	47.96	24.88	54.30	55.56	42.00
Team AMRITA-CEN	**45.79**	57.37	**80.0**	**55.67**	**29.58**	34.44	39.26	33.60
Team AmritaCENNLP	36.14	64.94	2.0	47.96	27.23	**65.56**	0.0	31.40
Team JUTeam_KS	2.41	88.45	22.0	50.75	21.13	63.58	45.18	40.40
Team ISMD	4.22	58.17	72.0	40.47	-	-	-	-

Table 4.11 Sentiment polarity classification results of Hindi and Bengali in terms of accuracy (%). First 5 rows show our results using different features combinations. Values in parenthesis denotes the deviation from the score when all features are integrated. The last three rows show the obtained results by other participants. In boldface, the best result in each column.

We compare our results to those obtained by the top four participants [128]. Our system achieves the highest accuracy in Bengali (accuracy: 43.20%), and score third for Hindi (accuracy: 49.68%) amongst six participating teams. The best-performing system for Hindi language, AMRITA-CEN, used SVM classifier trained on classical n-grams and emojis-usage features. The Second-ranked system, JUTeam_KS, used multinomial Naive Bayes classifier trained on structural, PoS tags and NE features. We achieve satisfactory results when classifying neutral and negative tweets, but the accuracy is poor for positive tweets. The same could be said about results obtained by other teams. One obvious reason is the imbalanced training data, where the number of positive tweets is still limited compared to neutral and negative tweets, especially for the Hindi language, see Table 4.9. We also analyze the number of overlapping tokens between the train and test sets per polarity class. The percentage of unique overlapping tokens between the train and test sets is 49.71% and 41.36% for Hindi and Bengali, respectively. However, the values drop to 29.91% and 27.07% for the positive tweets for the two languages, respectively. On further investigation, we find that overlapping tokens between neutral tweets in the training set and positive tweets in the test set to be 45.21% for Hindi, which explains why the majority of positive tweets are misclassified as neutral. This proves that the training data is not rich enough to capture the new positive instances effectively. The next potential sources of misclassifications are sarcastic tweets and noisy tweets, like Indian tweets written in English alphabets.

We evaluate the coverage of the provided Indian SentiWordNet compared to our induced lexicons by LexiExp. Considering the adjectives to be the most dominating sentiment expressions, after PoS tagging[22], we extract the adjectives from the Hindi tweets. We found that only 17.57% and 25.98% of the adjectives in the train and test set is available in the HindiSentiWordNet list. The coverage improves to 36.56% and 42.29% adjectives in the training and test set, respectively, after using LexiExp.

4.5.4 Sentiment Analysis of Customer Reviews

We performed another set of experiments on the Customer Reviews benchmark dataset[23] published by Bing Liu's group [70]. This dataset contains a small number of English customer reviews for five electronic products, namely Canon G3 camera, Nikon coolpix 4300 camera, Nokia 6610 mobile phone, Creative Labs Nomad Jukebox Zen Xtra 40GB mp3 player and Apex AD2600 Progressive-scan DVD player. The dataset is manually annotated with OTEs, and sentiment polarities. Sentiment strength is graded on a three-point scale in each polarity direction (+3 strongest, and -3 weakest). Descriptive statistics about the dataset are shown in Table 4.12. Not all review sentences have aspect-level polarity annotations, some sentences have sentence-level polarity annotation only. We choose to proceed with the proportion of

[22] Hindi Part of Speech Tagger: http://sivareddy.in/downloads#hindi_tools

[23] Customer Reviews: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

sentences that contain sentiment annotations in aspect-level for both explicit and implicit aspects.

Dataset	**#Reviews**	**#Review Sentences**	**#Unique Aspects**	**#Aspect-level Polarity sentences**
Nikon Coolpix	34	346	74	159
Canon G3	45	597	100	236
Nokia 6610	41	546	109	258
Apex Player	99	740	110	341
MP3 Player	95	1716	180	706

Table 4.12 Customer review dataset overview statistics. From left to right, the table gives the number of reviews for each product, overall review sentences, unique aspects, and proportion of sentences that contain sentiment annotations in aspect-level.

We follow the same data preprocessing and experimental setup as for the SemEval English dataset experiments. For our polarity classification experiments, we ignore the polarity strength and only consider two polarity classes: positive and negative.

Table 4.13 reports the performance of our supervised (using different feature combinations), and unsupervised systems in comparison with approaches that used the same dataset, namely rule mining method (ARM), semantic-based product aspect extraction approach (SPE) and Hu & Liu approach. The SPE method, proposed by Wei et al. [170], uses a list of positive and negative adjectives defined in the General Inquirer (GI) dictionary to identify opinion words, and subsequently extract their corresponding OTEs from customer reviews based on predefined association rules. ARM [71] exploits the Apriori algorithm to extract frequent itemsets as explicit OTEs in the form of nouns or noun phrases. The ARM approach suffers from many incorrect frequent OTEs (low precision), and is less useful when it comes to low-frequency OTEs. Hu and Liu [70] advanced the ARM approach by integrating two types of OTEs pruning, compactness pruning and redundancy pruning, in order to eliminate noisy, redundant and meaningless OTEs.

Excluding Apex product results, it is notable that our proposed supervised and unsupervised approaches outperform other systems. Our supervised approach with "All features" combination turned out to be the best-performing for all products, except Apex, with average-F1 score of 81.34 and Macro-averaged precision (MAP) of 81.84, whereas SPE and ARM approaches show the poorest performance with average-F1 of 58.66 and 53.5, and MAP of 49.8 and 47.92, respectively. We experiment several supervised models by leaving one of three features, i.e., DTs, OTEs and LexiExp, out of model each time, and observe the change in performance. The results show that using OTEs as features contributes significantly to the supervised model performance. Without this feature, the average-F1 and MAP scores drop

Model	**Nikon**			**Canon**			**Nokia**			**Apex**			**MP3**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Unsup.	71.2	81.2	75.9	75.3	81.8	78.4	72.5	77.8	75.0	63.0	60.5	61.7	70.2	81.8	75.5
Sup.-LexiExp	81.3	84.0	82.6	79.4	85.7	82.4	81.1	77.1	79.0	81.0	71.0	75.7	80.0	82.2	81.1
Sup.-DTs	79.9	84.4	82.1	77.0	85.7	81.1	80.5	76.6	78.5	82.0	71.1	76.2	**84.8**	79.0	81.8
Sup.-OTEs	74.7	**88.5**	81.0	78.8	83.4	80.5	73.6	**82.1**	77.6	**82.0**	65.1	72.6	78.3	80.2	79.2
Sup.+All	**81.9**	86.6	**84.9**	**80.6**	**86.0**	**83.2**	**83.1**	76.1	**79.4**	81.0	72.9	76.7	82.6	**82.5**	**82.5**
SPE	47.4	75.7	58.3	48.7	75.0	59.1	56.5	72.5	63.5	52.4	70.0	59.9	44.0	65.0	52.5
ARM	51.0	67.6	58.1	51.1	63.0	56.4	49.5	57.8	53.3	51.0	60.0	55.1	37.0	56.1	44.6
Hu & Liu	71.0	79.2	74.9	74.7	82.2	78.3	71.8	76.1	73.9	74.3	**79.7**	**76.9**	69.2	81.8	75.0

Table 4.13 OTE extraction performance (precision, recall and F-score) from the customer reviews dataset [70] based on 5-fold cross-validation on each product dataset. The first 5 rows show our unsupervised and supervised OTE approaches results, including 3-fold feature ablation test. The last three rows show the results obtained by comparable approaches, namely rule mining approach (ARM), semantic-based product aspect extraction approach (SPE) and Hu & Liu model. In bold, the best obtained result in each column.

by 3.16 and 4.36, respectively. The DTs and LexiExp features also play an important role but still less noticeable than the OTEs features. When eliminating DTs and LexiExp features, the average-F1 and MAP scores decrease by 1.4 and 1.0, and 1.18 and 1.28, respectively, although looking at individual results, DTs features seem to have more impact than LexiExp features.

Relying on average-F1 and MAP to analyze results is a bit tricky since the results obtained for the Apex product dataset are not in-line with those obtained for other products. However, we still achieve the best precision with 7.7 improvement over Hu & Liu, which means that we extract less noisy OTEs, especially in the case of multiword OTEs. Our models can effectively extract multiword OTEs, such as "customer support website" or "battery charging system". The error analysis reveals that the reasons for our poor performance for the Apex dataset in terms of recall are: (1) the dataset has many implicit OTEs, which was not considered in these experiments (2) dataset annotation inconsistencies, e.g., real OTE: "DVD player" vs. annotated: "player", real OTE: "focusing in low light" vs. annotated: "low light focus". Hence, exact OTE extraction is too strict to reflect the effectiveness of our OTE extraction. We can also use the same arguments to interpret the significant difference in performance between our supervised and unsupervised approaches. According to our error analysis, we believe that adopting a fuzzy matching strategy (e.g., word stems), like Hu & Liu, would lead to an improvement of ~7.0 in F1 score.

We extended our supervised and unsupervised approaches by integrating an extra step to identifying implicit aspects using our predefined DDGs, which we created earlier using the Amazon corpus, see Section 4.5.1. The methodology is based on the assumption that implicit aspects often can be indicated by sentiment words, e.g., the aspect "size" can be indicated by the sentiment word "small". Hence, using our DDGs, we try to find associations among explicit aspects and sentiment words in order to identify implicit aspects. Although there are general sentiment words, which can describe many aspects, e.g., great, poor, however, usually in a specific domain or context, it indicates only to a specific aspect. This makes our DDGs a halfway solution to extracting implicit aspects.

For each DDG, we highlight all sentiment words from our expanded lexicon that have a direct edge to one or more aspects from the identified aspect lists, see Table 4.4. We create a dictionary that maps each highlighted sentiment word to the aspect in which the edge has the highest weight. Examples of selected sentiment-aspect pairs per product entity are given in Table 4.14. We randomly show some examples of top selected aspect for each opinion word per product entity. Note that the same opinion word may be mapped to different aspects depending on the DDG domain.

DDG Topic	Sentiment Word	Candidate Aspect
Data Storage	excellent	drive
	fast	drive
	easy	installation
	compatible	software
Phones	poor	reception
	loud	speakerphone
	powerful	battery
	small	size
Camera	sharp	pictures
	affordable	price
	easy	use
	poor	quality

Table 4.14 Examples of selected sentiment-aspect pairs per product DDG.

To identify implicit aspects, we first select sentences that have sentiment words but no aspects have been extracted by our supervised or unsupervised approaches. Second, we extract sentiment words (i.e., aspects indicators) from them. Third, we use the generated dictionary to map these aspects indicators to their corresponding explicit aspects, which they actually represent. Finally, we claim these explicit aspects as implicit aspects.

The customer reviews dataset has a total of 156 implicit aspects that do not appear in the text. Out of 156, our proposed approach is able to find 98 implicit aspects indicators (62.82%). The remaining implicit aspects have no explicit polar words indicators (i.e., sentiment word indicators) to refer to them, e.g., "can carry it in my pocket" in reference to "size", "definitely wouldn't survive a drop" in reference to "materials", "it holds soooo much music" in reference to "storage", "no complaints" or "i'm pleased" in reference to the product in general, or comparative phrases like "is much smaller than". These cases are very difficult to identify and still an area of ongoing research. Out of the 98 implicit aspects indicator, our approach is able to successfully identify 57 implicit aspects (36.54%). We observe that the remaining 26.28% recall errors are caused by: (1) 4.49% annotation errors and inconsistencies, e.g., annotators annotate "size" and sometimes "weight" as an implicit aspect referred to by "light"; and "clear" refers to both "voice" and "volume", (2) In most cases 11.54%, our proposed approach was able to detect the true aspect, however it was not an exact match to the golden aspect, e.g., picture vs. pictures vs. pics, load vs. loading, file transfer vs. transfer, xtra 30gb player vs. player, speaker vs. speakerphone, operate vs. setup in reference to "intuitive", (3) 10.26% implicit-explicit replacement errors and out-of-dictionary sentiment indicators (e.g., bang-for-the-buck, ergonomical, eye-candy, bulky). In some domains, a specific sentiment word may refer to more than one aspect equally, e.g., "clear" in the domain of phones can refer to "screen" and "voice" equally. Possible further improvement is to develop a more advanced strategy to identify sentences with implicit aspects.

Table 4.15 gives the results of our approaches for aspect-based sentiment polarity classification on the customer reviews dataset. Since we only focus on sentiment polarity classification in aspect-level and instances that serve this task, it would be unjustifiable to compare our results with approaches, which have considered the whole dataset for their experiments. We chose to report the performance of our supervised and unsupervised approaches, including feature ablation test results to examine the contributions of LexiExp and OTEs features to the overall classification accuracy. It is clear from the table that our supervised models have produced better results as compared to unsupervised models. Among all, the best performance (highest accuracy) is achieved by the supervised model, which is trained using "all features" combination. With LexiExp and OTEs features left out, the performance drops indicating both features contribute significantly to the model training. That is, LexiExp and OTEs features have an effect of average accuracy reduction of 5.48% and 3.02%, respectively in comparison to the supervised model trained with "all features".

Analyzing misclassifications, we notice that many are caused by wrongly classified generated sentiment words. Often these are words that are either neutral or polysemous as both polar (domain-dependent sentiment words). An example:

"the power key is small (+)"

"Small" is being used to express a negative opinion, but was added to our lexicon via its synonym "tiny", which was added mostly as a synonym of the positive seed word "cute". As a result, "small" was added to the final lexicon as a positive word. The next major source

Model	Nikon	Canon	Nokia	Apex	MP3
Unsup.-LexiExp	75.7	63.3	65.3	75.5	75.9
Unsup.	76.2	64.2	69.9	75.5	78.6
Sup.-LexiExp	82.2	64.2	68.2	76.7	79.4
Sup.-OTEs	83.5	68.0	71.5	81.4	78.6
Sup.+All	**85.1**	**72.1**	**75.0**	**81.7**	**84.2**

Table 4.15 Aspect-based polarity classification performance (Accuracy %) on the customer reviews dataset. Results of our unsupervised and supervised polarity classification approaches, including feature ablation test results, are presented.

of misclassifications is out-of-lexicon sentiment words. In the following example, the word "loose" does not exist in our expanded lexicon, which led to a false prediction:

"The included lens is very loose(+)"

Polarity shift due to evaluation suggestion or presupposition would also cause misclassifications:

"A protective case would have been nice (+)"
"I find it more convenient to use 1-touch dialing (+)"

Chapter 5

Semantic Textual Similarity

5.1 Introduction

Semantic Textual Similarity (STS) measures the degree of semantic equivalence between a given pair of texts. It also captures a graded notion that given two snippets of text, some texts are more similar than others. Measuring textual similarity is increasingly important to a wide range of Natural Language Processing (NLP) tasks and applications [5]. For example, in Information Retrieval (IR) applications, STS plays a crucial role in ranking the retrieved search results according to their degree of semantic equivalence to the text query [11]. For automatic answer grading, STS can be used to assess natural language answers (i.e., students) based on their similarity with expert-provided (i.e., teacher) correct answers [112, 111]. Similarly, STS can aid Question Answering (QA) applications in order to select the most suitable answer from a large pool of potential answers [171]. STS has also been employed in automatic text summarization to reduce redundancy in generated summaries [47]. Accurate estimation of STS could also act as an extrinsic evaluation method for many tasks like textual entailment, paraphrase generation and semantic relatedness.

In order to measure the similarity between a pair of texts, each text needs a representation. One way is using raw text (i.e., a list of words that form a sentence). Using raw text is straightforward, however, it lacks explicit information about syntactical structure or semantic meaning. A richer representation, which provides a better basis for comparing texts based on semantic or structural aspects, can provide better similarity estimation.

In this chapter, we propose a supervised and unsupervised models for learning textual similarity, which can identify and score textual similarity between a pair of texts, and we present a way for combining both models into an ensemble. Specifically, we try to examine the impact of using dependency graph similarity and coverage features extracted from Domain Dependency Graphs (DDGs), and leverage supervised machine learning techniques in order to improve the semantic similarity prediction. We also introduce an approximate sub-graph alignment approach to find a sub-graph in a candidate text dependency graph that is similar to a given query text dependency graph, allowing for node gaps and mismatches, where a certain word in one dependency graph cannot be mapped to any word in the query text graph, as well as graph structural differences. Our unsupervised methodology builds upon the word-alignment-based approach proposed by Sultan et al. [159]. Similar to Sultan et al., our aligner performs one-to-one word alignment based on the semantic similarity between the words as well as the similarity between their local contexts. However, in our approach, the final overall STS score is based on weighted word alignment scores.

We evaluate our approaches using four different datasets featuring different languages, domains and various text lengths:

1. Re-ranking to improving document retrieval precision using a news dataset [85].

2. Automatic short answer scoring using a standard benchmark dataset [111].

3. Multilingual and cross-lingual semantic similarity using the SemEval-2017 shared Task 1 benchmark datasets [86].

4. Question answering and answer triggering using the SemEval-2017 shared Task 3 and WikiQA benchmark datasets, respectively [117, 64].

This chapter covers the first three applications in detail. The work on question answering and answer triggering applications will only be reviewed briefly in Section 5.4.4. The remaining of this chapter is organized as follows: Section 5.2 reviews relevant literature on text similarity and STS, Section 5.3 introduces our methodology. Section 5.4 shows how our proposed models can be applied to common NLP tasks, namely learning to rank, automatic short answer grading, semantic textual similarity for both monolingual and cross-lingual texts, and question answering.

5.2 Related Work on Semantic Textual Similarity

Researchers have made substantial progress in estimating textual similarity motivated by the annual SemEval Task for STS [8, 6, 4, 3, 5, 7]. The task has been held annually from 2012 to 2017, with a total of 556 received submissions from 213 participating research teams. A review of the submitted models during this period provides a comprehensive insight into the evolution of STS methods.

Early methods in text similarity focused on training regression models using traditional lexical surface overlap features, like shared words, and basic syntactic similarity. Improved versions to these simple lexical matching methods have considered stop-word removal, lemmatization, stemming, part-of-speech tagging, longest subsequence matching, as well as various normalization and weighting factors [77]. However, in many cases, two pieces of texts may be semantically related despite having few, if none, words in common, e.g., "*exchange messages with a classmate*" and "*texting a friend*". A possible way to address this issue is by incorporating meaningful semantic information into measures of text similarity, like knowledge-based and corpus-based measures. Knowledge-based measures quantify the degree of semantic relatedness using information drawn from lexical databases (e.g., WordNet [106]) and semantic networks [151, 175]. While knowledge-base resources may be scarce or outdated, corpus-based metrics, like Pointwise Mutual Information (PMI-IR) [166] and Latent Semantic Analysis (LSA) [91], uses information derived directly from large corpora, like words embeddings, word distribution and word co-occurrences.

In 2014 and 2015, the top results on the STS SemEval shared task were achieved by unsupervised systems [158, 79]. Sultan et al. [158] present an unsupervised alignment algorithm that predicts the semantic similarity score based on the proportion of word alignments in two given sentences. Using WordNet [106], Word Embeddings [105, 16] and PPDB [58], words from each sentence are aligned based on their similarity as well as the similarity of the contexts in which they occur.

Whereas having high-dimensional features adds to the latency of a supervised system, having few features, on the other hand, is uninformative nor definitive, especially when comparing short texts. Since short texts contain less statistical information and syntactic patterns, multiple approaches have been proposed to operate specifically on short texts [176, 151, 63, 102]. Some attempts to incorporate the output score of unsupervised alignment methods as an additional feature to train supervised regression model are presented in [159, 79], which usually results in slight improvements.

Recently, deep learning and sentence embeddings models have achieved very promising results; the top performing-systems from the SemEval STS 2016 and 2017 use deep-learning-based models [143, 32, 2, 162, 152]. However, computing sentence similarity remains a non-trivial task due to the variability of natural language expressions, texts with different lengths and complex dependency structure [114]. This is why Pilehvar and Navigli [131] investigated a unified graph-based approach for measuring semantic similarity, which enables comparison of linguistic units at multiple levels: senses, words, and texts.

5.3 Learning Textual Similarity

Given a pair of texts q and d, textual similarity captures the fact of how much d conveys the same information as a q, and and vice versa. In the context of IR or QA experiments, we refer to q as *query text* (i.e., search query or question), and d as *candidate text* (i.e., retrieved result or answer).

The novel contribution described in this chapter is constituted by three feature types: dependency structure features, expansion features and coverage features. In the following subsections, we describe each of these features in more detail.

5.3.1 Supervised Model: Similarity-Based Features

Similarity-based features measure the shared feature types between a pair of texts, q and d. For texts d and q, we create vector representation $\vec{d}$ and $\vec{q}$ for various feature types. Each entry in one vector corresponds to the existence/presence (i.e., $d_i/q_i \in \{0,1\}$), frequency (i.e., $d_i/q_i \in \mathbb{N}$) or Tf-Idf weight (i.e., $d_i/q_i \in \mathbb{R}$) of a given feature type in a text. After removing stopwords, we consider the following feature types:

Bag-of-Words (BOW): we represent the content of each text using Bag-of-Words model (BoW). As we mentioned earlier, this representation disregards syntax or word order. In this case, similarity is measured by the shared vocabulary between both d and q. We also employ a second version of this feature using stemmed words.

Topic Distribution: we model each document as a vector of topics using Latent Dirichlet Allocation model (LDA) [25]. Topics are more semantically-based. Similarity between a pair of texts is estimated by comparing the topics of their words regardless of differences in word forms.

Dependency Structure: another important similarity measure is dependency structure similarity. Based on our DDGs construction methodology in Chapter 2, we aggregate individual dependency relations obtained from a parser, weigh them with Tf-Idf and produce a graph, which contains the highest-ranked content words and their dependency relations. For a text d, dependency graph $G_d = \{V_d, E_d\}$, where $V_d = \{w_1, ..., w_N\}$ represents the content words w_i in a text, and E_d is a set of edges, where each edge e_{jk} represents a directed dependency relation between w_j and w_k. Written also as a list of triples as follows: $"w_j" \rightarrow "w_k"[label = "e_{jk}"]$.

A generated dependency graph is then filtered according to the following three conditions:

- Tf-Idf $(w_j,d) \geq \alpha$ or Tf-Idf $(w_k,d) \geq \alpha$
- Tf-Idf $(w_j\ w_k,d) \geq \beta$
- Tf-Idf $(e_{jk}\ w_j\ w_k,d) \geq \lambda$

where α, β and λ are ≥ 0. When α, β and $\lambda = 0$, no Tf-Idf filtering is applied. For q, dependency graphs are generated, and filtered in the same manner.
Similarity is then measured on these three levels by representing each text as a vector of words, pairs and relations.

Named Entities: in this case, we measure similarity based on the shared named entities between a pair of text.

Expansion Features: as the variability of language allows expressing the same concepts, entities and facts in different words, measuring similarity purely based on exact word matching does not fully capture conceptual matching. We expand content words, i.e., (common and proper) nouns, adjectives, verbs and adverbs in each text to its most similar words using the distributional thesauri (DTs) [21].

Once all the above vectors are constructed, the similarity between each d and q text pair can be measured using the cosine similarity (see Equation 1.1).

PoS Tags Longest Common Subsequence: we measure the length of the longest common subsequence of POS tags (PoSLCS) between d and q text pair. Additionally, we also average this length by dividing it by the total number of tokens in each sentence separately. This results in three features, $PoSLCS$, $\frac{PoSLCS}{len(q)}$, and $\frac{PoSLCS}{len(d)}$.

5.3.2 Supervised Model: Coverage-Based Features

As a text gets longer, term frequency factors increase, and thus having a high similarity score is likelier for longer than for shorter texts. IR research has also shown that document length normalization is essential to guarantee that documents are retrieved with similar chances as their likelihood of relevance regardless of their length [10]. The same applies to QA and answers grading applications: longer answers should not receive unjustly higher scores. Normalizing vectors using the cosine L_2 norm has proven to have several limitations due to the use of the individual terms weights for text length normalization (see Equation 1.1) [33, 154]. This dependency is undesirable when the text includes infrequent terms with high Idf value, which can significantly increase the overall cosine normalization L_2 factor, and cause inaccurate weighting for the other terms in the text. Accordingly, the new weights may

not reflect the actual importance of the terms in the text representation. We try to solve this problem by incorporating a set of coverage features. These features are more powerful for tasks where more than one candidate text is being evaluated, e.g., IR and QA.
Let $G_d = \{V_d, E_d\}$ and $G_q = \{V_q, E_q\}$ be the dependency graphs of d and q, respectively. Coverage is measured at several levels as follows:

Vocabulary Coverage: We calculate vocabulary coverage by computing the number of one-to-one node correspondence between both q and d dependency graphs divided by the overall number of nodes in the query text q dependency graph, as in the following equation:

$$\frac{|V_d \cap V_q|}{|V_q|} \tag{5.1}$$

Relation Coverage: We calculate relation coverage by computing the number of one-to-one edge (triple) correspondence between both q and d dependency graphs divided by the overall number of edges in the query text q dependency graph:

$$\frac{|E_d \cap E_q|}{|E_q|} \tag{5.2}$$

Pair Coverage: As in relation coverage, however in this case, we ignore the relation type and edge direction.

Graph Coverage: Before we present more details on how graph coverage features are measured, it is necessary first to define our approximate dependency sub-graph alignment approach.

The main idea of the approximate dependency sub-graph alignment approach is to find a sub-graph $G_s = \{V_s, E_s\}$, where $G_s \subseteq G_d$, that is approximately similar to a query text graph G_q. Algorithm 1 shows the pseudo-code of the dependency sub-graph approximate matching algorithm. The key advantages of this algorithm is its ability to capture similarity based on the dependency structure, not words ordering.

First, we obtain the nodes intersection $V_{intersection}$ between both q and d dependency graphs. We then find the shortest path between every pair (w_j, w_k) of vertices belongs to the $V_{intersection}$ set in the candidate text dependency graph using Dijkstra's algorithm [50]. Each edge is given a weight of 1, and edges directions are ignored during the process of the algorithm. Due to linguistic variation, we may not find a sub-graph that match the exact query text graph, however, we may find a sub-graph that matches the query text graph approximately. We define a threshold parameter t to allow node gaps and mismatch in the

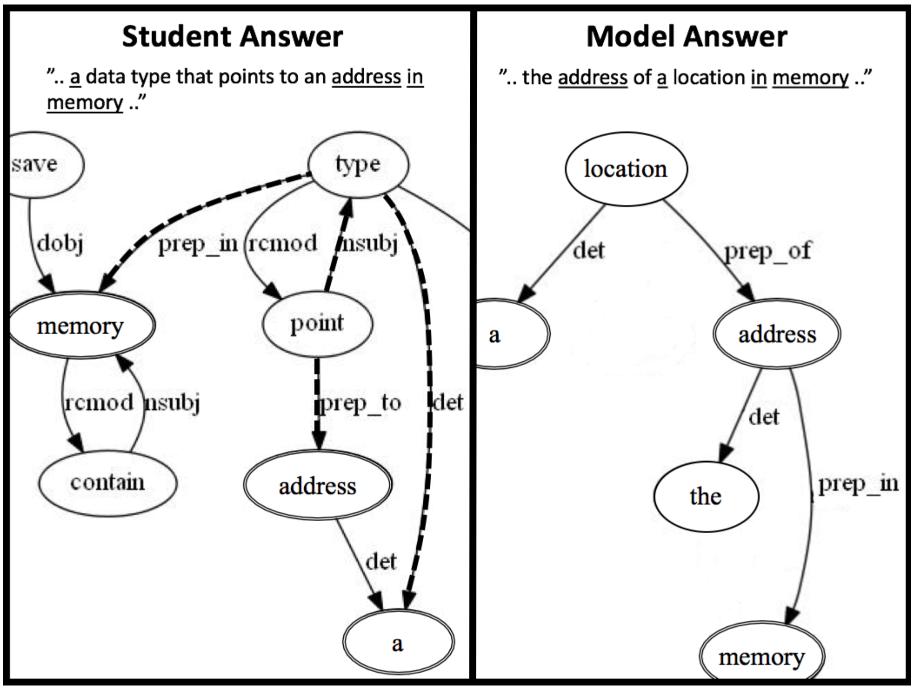


Fig. 5.1 Approximate sub-graph matching illustration. Example is taken from Mohler et al. [111]. Given a model and a student candidate answer, double-lined nodes represent the shared words between both answers and connections between words represent dependency relations. Direction and dependency types are ignored. Algorithm 1 uses the dependency structure similarity of local neighbors within the Shortest Path (SP $\leq$ **t**), to find an approximate sub-graph that match the model answer. The selected sub-graph is highlighted by bold dotted lines. In this example, **t = 3**.

case where some nodes in the query text cannot be mapped to any nodes in the candidate text graph. If the shortest path size (i.e., number of edges between w_j and w_k) is less than or equal t, the path will be added to the sub-graph G_s. By setting t to a value greater than 1, it is much more likely to capture syntactic variations. Figure 1 shows examples of sub-graph matching from the dataset of [111].

The resulting G_s is used to measure two graph coverage features as follows:

$$\frac{|E_s|}{|E_d|} \tag{5.3}$$

$$\frac{|E_s|}{|E_q|} \tag{5.4}$$

Since a much more relevant candidate text is much more likely to have a larger overlap with the query text than other less relevant candidates, this may well tend to improve the similarity assessment.

```
**Input:** $G_d$, $G_q$, Threshold $t$
**Output:** $G_s$
$V_{intersection} \leftarrow \{V_d \cap V_q\}$ ;
**for** $j \leftarrow 1$ *to* $|V_{intersection}| - 1$ **do**
    **for** $k \leftarrow j+1$ *to* $|V_{intersection}|$ **do**
        $Path \leftarrow dijkstra.getPath(G_d, w_j, w_k)$;
        **if** $Path \neq null$ *and* $Path.size(w_j, w_k) \leq t$ **then**
            $G_s \leftarrow G_s \cup Path$;
        **end**
    **end**
**end**
```

Algorithm 1: Dependency sub-graph approximate alignment methodology.

5.3.3 Unsupervised Model

Inspired by Sultan et al. [159, 32], our unsupervised solution calculates a similarity score based on the alignment of a pair of sentences. However, when calculating the final similarity score, we weigh the aligned words by their Tf-Idf. As presented in Figure 5.2, given a pair of sentences S_1 and S_2, the alignment task builds a set of matched word pairs. Each matched pair is then given a similarity score on the scale [0-1]. This matching score defines the strength of the semantic similarity between the aligned pair of words, with 1 indicating the highest similarity.

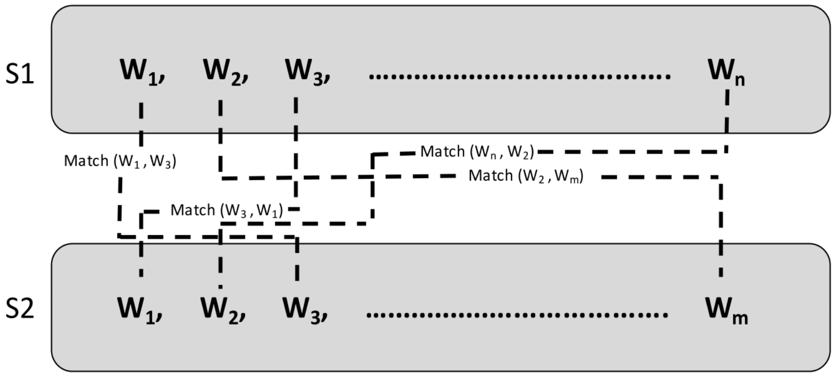

S1
W_1, W_2, W_3, ... W_n
Match (W_1, W_3)
Match (W_n, W_2)
Match (W_2, W_m)
Match (W_3, W_1)
S2
W_1, W_2, W_3, ... W_m

Fig. 5.2 Word alignment of S1 in S2.

After preprocessing the sentences and removing stopwords, the system starts with matching exact similar words (lemmas), and words that belong to the same WordNet hierarchy (synonyms, hyponyms, and hypernyms). Words matched according to these two types of alignment are considered to be an exact match and are given a score of 1. Afterward, we handle the words that have not been matched in the preceding step. We compute the cosine similarity between the embedded vectors of the unmatched words in one sentence and all the words in the other sentence. Using a greedy strategy, and based on the cosine similarity scores, we align each word from one sentence to the word in the other sentence, whose embedded vector yields the maximum cosine similarity. We apply this process for all words in both sentences S_1 and S_2. Figure 5.2 shows the alignment of words of S1 in S2. Note that the order of the pairs is crucial. This means if w_j from S_2 is the best match to w_i from S_1, that does not necessarily mean that w_i is the best match for w_j.

Given a set of matched pairs, $M = \{(w_{11}, w_{21}), (w_{12}, w_{22}), \ldots, (w_{1N}, w_{2N})\}$, the final alignment score is calculated as shown in Equation 5.5.

$$Score_{match} = \frac{\sum_{k=1}^{N} Tf\text{-}Idf(w_{1k}) \times match(w_{1k}, w_{2k})}{\sum_{w \in (S_1 \cup S_2)} Tf\text{-}Idf(w)} \tag{5.5}$$

For a matched pair of words w_{1k} and w_{2k}, from S_1 and S_2, respectively, $match(w_{1k}, w_{2k})$ returns the maximum alignment score of w_{1k} with its best matching word from the other sentence w_{2k}. The alignment score is weighted using the term frequency inverse document frequency of w_{1k}, $Tf\text{-}Idf(w_{1k})$.

Based on the previous alignment pairing set M, we use word embeddings to compute the cosine similarity between the embeddings of the syntactic heads of each matched pair. This leads to another unsupervised dependency alignment score as follows:

$$Score_{dep} = \frac{\sum_{k=1}^{N} Tf\text{-}Idf(\widehat{w_{1k}}) \times cos(\widehat{w_{1k}}, \widehat{w_{2k}})}{\sum_{w \in (S_1 \cup S_2)} Tf\text{-}Idf(w)} \tag{5.6}$$

For each pair of matched words w_{1k} and w_{2k}, we calculate the weighted cosine similarity $cos(\widehat{w_{1k}}, \widehat{w_{2k}})$ between the embedded vectors of their syntactic dependency heads $\widehat{w_{1k}}$ and $\widehat{w_{2k}}$, respectively.

The final unsupervised similarity estimation is then obtained by averaging the two scores from Equations 5.5 and 5.6 as follows:

$$Similarity = \frac{Score_{match} + Score_{dep}}{2} \tag{5.7}$$

5.4 Evaluation

We evaluate our model using four different applications, namely re-ranking to improving document retrieval precision, automatic short answer grading, cross-lingual and monolingual STS, as well as question answering and answer triggering. Each of the following subsections describes one application, for which we explain the preparation of the evaluation dataset, the experimental setup and our findings.

5.4.1 Re-ranking for Article-Summary Matching

Examining the effect of results ranking, Jansen and Spink [75] observed that most users do not browse results beyond the first page, and that the higher the document position in the first results' page retrieved by a search engine, the more likely a user is to read that document. Users care much less about what happens in lower positions (e.g., after the 10th) in the rank, as they typically do not browse the next results' page. By minimizing the huge amount of relevant results to few highly relevant to the users' query and re-ranking them to the upper top ranks, users are more likely to find their desired results quickly and easily.

Similar to Hagen et al. [66], we utilize the ranking output of an IR system to do re-ranking. We choose to select the top relevant documents initially ranked by an IR system, and incorporate our DDG features to improve the ranking precision.

Lucene Ranking

Our re-ranking method is built on the top of Lucene. Lucene offers an open-source information retrieval library, which provides IR-related tasks like indexing, querying, language analysis, results scoring and retrieval. Figure 5.3 shows how our re-ranking setup works with Lucene. Depending on the chosen language analyzer, the documents are internally tokenized, stemmed and filtered for stopwords. The analyzer preprocesses and extracts the terms on which the searching can be done. The terms are then indexed into a format that facilitates rapid searching, a.k.a. inverted index. When a query is issued, it gets analyzed and relevant results are selected from the collection by matching the query against the index. Finally, relevant documents are scored and ranked according to the following equation:

$$Score(d,q) = \sum_{w \in q} Tf(w,d) \times Idf(w) \times norm_{length}(d) \tag{5.8}$$

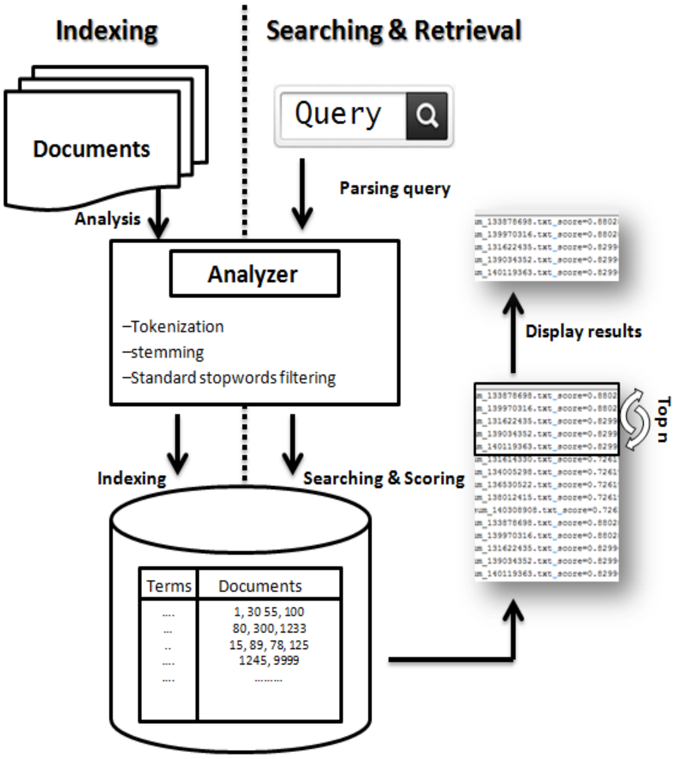


Fig. 5.3 Re-ranking top-n results of a retrieval system.

where $Tf(w,d)$ represents the word w frequency in document d, $Idf(w)$ is the inverse number of documents in which word w appears, and $norm_{length}(d)$ is the length normalization factor for document d.

Dataset

We used a set of 37,164 German news articles collected from the Spiegel online website from the year 2015. The corpus includes German news articles from different genres like sports, politic, economics, health, entertainment, etc. We remove images/video only articles, filter out irrelevant information like source agency, date and translator name, clean HTML tags, and extract only articles with summaries. The resulting corpus consists of 1130 article-summary pairs. Each article has one corresponding summary that was created manually by the article author. Summaries are abstractive, which involves paraphrasing the facts from the original article using new novel sentences. Length ranges are [45-950] and [1185-9560] characters, and [7-107] and [216-1390] words, for summaries and articles, respectively. The news are highly correlated due to the short one-year interval. Similar events are discussed in different contexts, therefore simple word similarity features would not be able to discern the correct summary from other relevant articles' summaries on closely related topics. Thus, this dataset is suited for testing the capability of methods that assess a certain semantic understanding of texts – as opposed to the DUC datasets[1], where we found string-based matching to yield almost perfect scores in a preliminary experiment. The remaining articles, which have no corresponding summaries, were used as a background corpus for Tf-Idf calculation and topic model training, as described in Section 2.2.1.

To create our dataset, we index the articles with Lucene, and use the summaries as queries to retrieve the n top-ranked articles. We label the correct matching summary-article pairs as "1" and "0" otherwise. Overall, we have 5650 examples and 11300 examples, in the cases where $n = 5$ and $n = 10$, respectively. We apply the same process with the summaries indexed and the articles as queries. Table 5.1 shows Lucene's retrieval results for both settings. Precision at n reports the fraction of relevant documents ranked in the top n results. Note that not all summary-article pairs could be correctly matched even at n=100. Another interesting observation is that the retrieval performance is not equally effective for retrieving texts and summaries, since summaries are much shorter and have less distinctive words. An obvious application for the Text/Sum model is text summarization, while a Sum/Text model is useful to be used for sentences weighting and selection when generating extractive summaries.

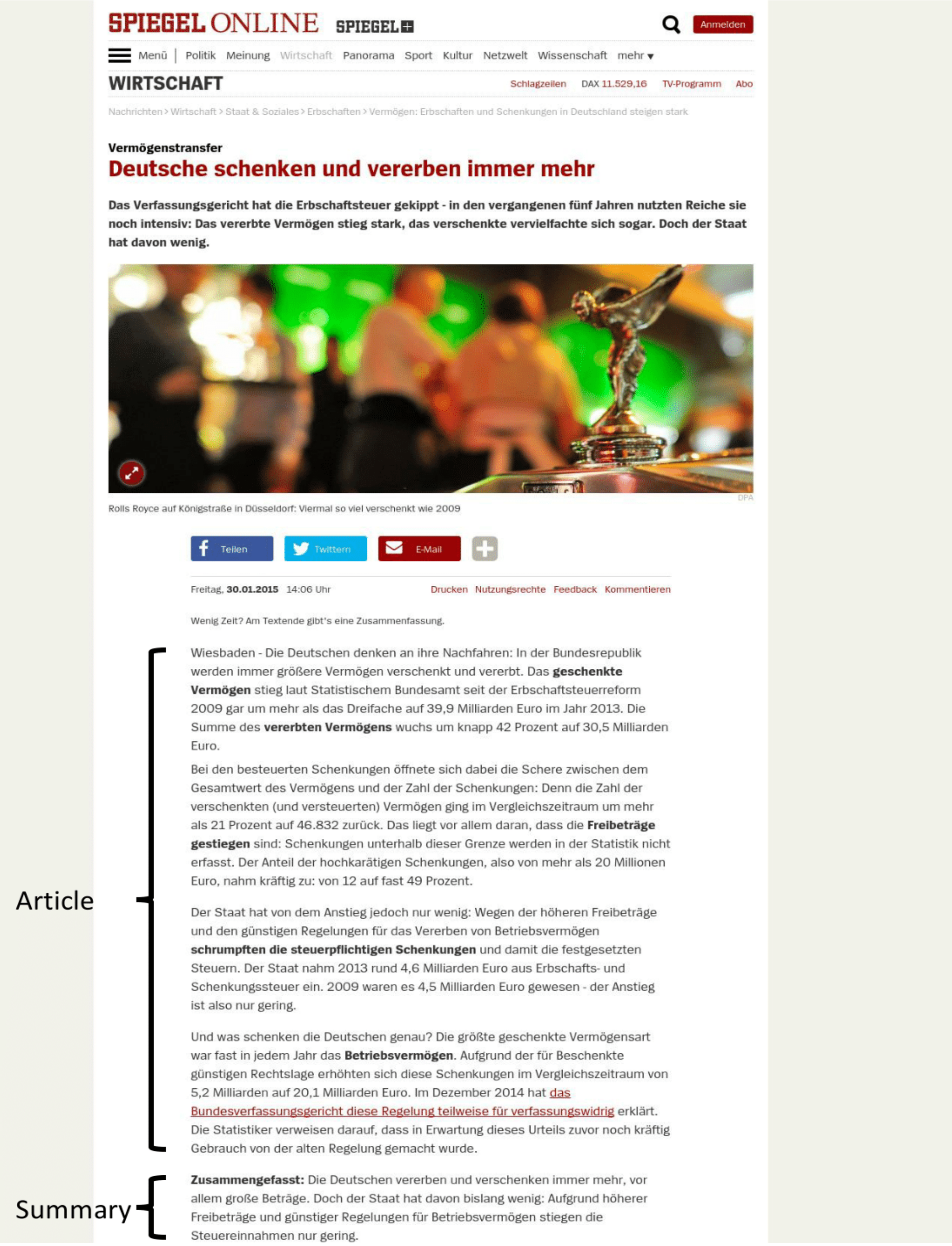
P@n	**Sum/Text**	**Text/Sum**
P@1	1029 (0.9106)	887 (0.7849)
P@3	1111 (0.9831)	1021(0.9035)
P@5	1122 (0.9929)	1052 (0.9309)
P@10	1128 (0.9982)	1090 (0.9646)
P@20	1128 (0.9982)	1114 (0.9858)
P@50	1129 (0.9991)	1123 (0.9938)
P@100	1129 (0.9991)	1129 (0.9991)

Table 5.1 Baseline retrieval performance by Lucene. Retrieval is evaluated by P@n. The number in parenthesis shows the overall P@n for the entire dataset of 1130 article-summary pairs. **Sum/Text:** summaries on index, articles for retrieval. **Text/Sum:** articles on index, summaries for retrieval.

Experimental Setup

To perform our experiments, we use our supervised methodology discussed earlier in Sections 5.3.1 and 5.3.2. To prepare the features, we use the implementation of GibbsLDA [130] for topic modeling, and GermaNER [18] for named entities extraction. Dependency graphs for both queries and documents are generated using the German collapsed parser by Ruppert

[1] http://duc.nist.gov/



SPIEGEL ONLINE SPIEGEL+
Anmelden
Menü | Politik Meinung Wirtschaft Panorama Sport Kultur Netzwelt Wissenschaft mehr ▾
WIRTSCHAFT
Schlagzeilen DAX 11.529,16 TV-Programm Abo
Nachrichten > Wirtschaft > Staat & Soziales > Erbschaften > Vermögen: Erbschaften und Schenkungen in Deutschland steigen stark
Article
Summary

Vermögenstransfer

Deutsche schenken und vererben immer mehr

Das Verfassungsgericht hat die Erbschaftsteuer gekippt - in den vergangenen fünf Jahren nutzten Reiche sie noch intensiv: Das vererbte Vermögen stieg stark, das verschenkte vervielfachte sich sogar. Doch der Staat hat davon wenig.

DPA

Rolls Royce auf Königstraße in Düsseldorf: Viermal so viel verschenkt wie 2009

Freitag, **30.01.2015** 14:06 Uhr

Wenig Zeit? Am Textende gibt's eine Zusammenfassung.

Wiesbaden - Die Deutschen denken an ihre Nachfahren: In der Bundesrepublik werden immer größere Vermögen verschenkt und vererbt. Das **geschenkte Vermögen** stieg laut Statistischem Bundesamt seit der Erbschaftsteuerreform 2009 gar um mehr als das Dreifache auf 39,9 Milliarden Euro im Jahr 2013. Die Summe des **vererbten Vermögens** wuchs um knapp 42 Prozent auf 30,5 Milliarden Euro.

Bei den besteuerten Schenkungen öffnete sich dabei die Schere zwischen dem Gesamtwert des Vermögens und der Zahl der Schenkungen: Denn die Zahl der verschenkten (und versteuerten) Vermögen ging im Vergleichszeitraum um mehr als 21 Prozent auf 46.832 zurück. Das liegt vor allem daran, dass die **Freibeträge gestiegen** sind: Schenkungen unterhalb dieser Grenze werden in der Statistik nicht erfasst. Der Anteil der hochkarätigen Schenkungen, also von mehr als 20 Millionen Euro, nahm kräftig zu: von 12 auf fast 49 Prozent.

Der Staat hat von dem Anstieg jedoch nur wenig: Wegen der höheren Freibeträge und den günstigen Regelungen für das Vererben von Betriebsvermögen **schrumpften die steuerpflichtigen Schenkungen** und damit die festgesetzten Steuern. Der Staat nahm 2013 rund 4,6 Milliarden Euro aus Erbschafts- und Schenkungssteuer ein. 2009 waren es 4,5 Milliarden Euro gewesen - der Anstieg ist also nur gering.

Und was schenken die Deutschen genau? Die größte geschenkte Vermögensart war fast in jedem Jahr das **Betriebsvermögen**. Aufgrund der für Beschenkte günstigen Rechtslage erhöhten sich diese Schenkungen im Vergleichszeitraum von 5,2 Milliarden auf 20,1 Milliarden Euro. Im Dezember 2014 hat das Bundesverfassungsgericht diese Regelung teilweise für verfassungswidrig erklärt. Die Statistiker verweisen darauf, dass in Erwartung dieses Urteils zuvor noch kräftig Gebrauch von der alten Regelung gemacht wurde.

Zusammengefasst: Die Deutschen vererben und verschenken immer mehr, vor allem große Beträge. Doch der Staat hat davon bislang wenig: Aufgrund höherer Freibeträge und günstiger Regelungen für Betriebsvermögen stiegen die Steuereinnahmen nur gering.

Fig. 5.4 Article from Spiegel Online German language news website.

et al. [142], and filtered using Tf-Idf thresholds in three levels. By manual inspection, α, β and λ are set to 10, 5 and 2 respectively. For lexical expansion features, we obtain the top 10 DT expansions using the JoBimText API[2].

Once the similarity and coverage scores are computed for each summary-article pairs, we use a cost-sensitive Multilayer Perceptron (MLP) neural network to handle the imbalance between positive and negative examples. Since we are only aware of the one correct summary for each article, we will always have fewer positive examples (i.e, labeled as "1") than negative examples. False positives are assigned a larger cost than false negatives, so the classifier would not be biased toward negative instances. The cost of false negatives is fixed to 1. During the validation phase, we explore different costs to find the best cost for the false positive class. We have found that a cost of $n-1$ performs best across all the training/validation rounds.

We run different experiments with different MLP architectures and learning parameters. To evaluate, we use 5-fold cross-validation. We choose the model that provides the best overall accuracy and balanced classification error rate between the two classes, which was determined on a smaller version of the dataset in preliminary experiments.

	Sum/Text				Text/Sum			
Features	Precision	Recall	F-Measure	TP	Precision	Recall	F-Measure	TP
All	**0.916**	**0.894**	**0.899**	**1026**	**0.915**	**0.889**	**0.896**	**960**
BOW	0.890	0.853	0.864	1013	0.893	0.850	0.862	929
Dependency	0.887	0.838	0.850	1003	0.890	0.843	0.856	921
Coverage	0.893	0.853	0.864	919	0.850	0.861	0.862	917
Cov+Dep	0.904	0.874	0.882	1011	0.904	0.870	0.880	942
All-(Cov+Dep)	0.902	0.865	0.874	1023	0.901	0.860	0.870	948

Table 5.2 Binary relevancy classification results using MLP (n=5). Results show binary relevancy classification precision (P@1), recall, F-measure and True Positives (TP).

The neural network structure includes 3 hidden layers, with $(f+c)/2$ neurons in each layer[3], where f is the number of input features and c is the number of classes (i.e., $c=2$). Training time is set to 1000 epochs.

To re-rank, MLP[4] is configured to return the probability distribution over the two classes. We re-rank the results according to the descending ordering of the probability distribution of the positive class.

[2] www.jobimtext.org/jobimviz-web-demo/api-and-demo-documentation/

[3] Following a common rule of thumb

[4] Learning rate = 0.5, momentum = 0.2

Features	Sum/Text				Text/Sum			
	Precision	Recall	F-Measure	TP	Precision	Recall	F-Measure	TP
All	**0.951**	**0.929**	**0.936**	**1038**	**0.949**	**0.923**	**0.931**	**994**
BOW	0.938	0.894	0.908	1024	0.936	0.889	0.904	971
Dependency	0.933	0.879	0.895	1014	0.933	0.883	0.899	954
Coverage	0.930	0.881	0.896	972	0.935	0.888	0.903	962
Cov+Dep	0.945	0.918	0.926	1020	0.941	0.899	0.912	989
All-(Cov+Dep)	0.943	0.906	0.917	1035	0.942	0.903	0.915	989

Table 5.3 Binary relevancy classification results using MLP (n=10). Results show binary relevancy classification precision (P@1), recall, F-measure and True Positives (TP).

Features	n=5		n=10	
	Sum/Text	**Text/Sum**	**Sum/Text**	**Text/Sum**
All	1080 (0.963)	996 (0.946)	1077 (0.954)	1018 (0.933)
BOW	1064 (0.948)	987 (0.938)	1048 (0.929)	994 (0.911)
Dependency	1050 (0.935)	980 (0.931)	1037 (0.919)	976 (0.895)
Coverage	1038 (0.925)	962 (0.914)	1031 (0.914)	973 (0.892)
Cov+Dep	1079 (0.961)	991 (0.942)	1072 (0.950)	1010 (0.926)
All-(Cov+Dep)	1064 (0.948)	986 (0.937)	1048 (0.929)	993 (0.911)

Table 5.4 Re-ranking results using the MLP probability distribution of different relevancy classifiers when n=5 and n=10. The table shows the improvement in P@1 after the re-ranking, see Lucene results in 5.1. Numbers in brackets are the results of dividing by the number of cases, (1122,1052), (1128,1090), where the correct document is in the top 5 or top 10 Lucene results respectively, which form an upper bound.

Results and Discussion

The best obtained relevancy classification results are reported in Table 5.2 and Table 5.3. Since we are only aware of the correct article-summary pairs, we use P@1 as a measure of performance. We also report the recall, F-measure and true positives. We test the performance using different sets of features. From the results, we observe the following: First, using all the features achieves the best performance overall in all cases. Second, using a combination of coverage and dependency features leads to the second-best performance and plays a role in providing comparable performance to that obtained using all the features with an unnoticeable drop in true positives. Third, F-measure falls with an average of 0.0215 (in four cases) when excluding these two features, and using each feature in isolation does not lead to any improvement, but achieves comparable results to the ones obtained when using BOW features. Forth, in most cases, we outperform Lucene P@1 in terms of true positives. The improvement is most clearly seen when we use our neural network models for re-ranking, see Table 5.4.

In manual error analysis, we generally observe limitations on very short summaries that have no intersection with the text – most extremely noticed for an article on fashion history (866 words) with the summary "Der Anzug ist die Uniform eines Gentlemans" (tr. the suit is the uniform of gentlemen). Other errors could be addressed by a German compound splitter, as there are frequently compounds where only the parts match, such as "*Ratenkreditangebote*" (tr. installment credit offers) and "*Ratenkredite*" (tr. installment credits), other examples include derivational matches like "*vorweihnachtliche*" (tr. pre-Christmas) and "*Vorweihnachtszeit*" (tr. pre-Christmas time), which could be addressed by an improved morphology component that also includes compound analysis.

5.4.2 Automatic Short Answers Grading

We provide another evaluation for our method on automatic short answers grading. The task of automatic short answer grading is to assess short natural language answers based on their similarity with expert-provided correct answers.

Dataset

We use the dataset from Mohler et al. [111][5]. The dataset consists of 81 computer science questions from data structures exams and 2273 student answers. Length ranges are [2-934] and [4-256] characters, and [1-160] and [1-53] words, for students' answers and model answers, respectively. The answers were graded by two human judges, using an integer scale from 0 to 5 based on the extent to which the student answers contain semantic overlap with the teacher's answer. The reported inter-annotator agreement (IAA) between both judges is 0.586% (Pearson's ρ) with 0.659 Root Mean Square Error (RMSE).

Experimental Setup

Dependency graphs for both questions and answers are based on collapsed dependencies from the Stanford Parser[6]. Since the average text length, in this case, is short, we choose small values for the weighting thresholds. By manual inspection, we set α, β and λ to 4, 2 and 1 respectively. We use New York Times articles [146] within the years 1998-2000 as a background corpus for Tf-Idf calculation. The topic model is trained using 36 million sentences from the recent English Wikipedia dump.

[5] http://web.eecs.umich.edu/~mihalcea/downloads.html#saga

[6] http://nlp.stanford.edu/software/lex-parser.shtml

We train a MLP with one hidden layer using default parameters[7] for 5000 training epochs to increase stability. Following [111], we apply a 12-fold cross validation over the entire dataset for evaluation.

Results and Discussion

Features	ρ	**RMSE**
Tf-Idf	0.327	1.022
BoW	0.480	1.042
Mohler et al. [111]	0.518	0.978
Inter-annotator Agreement (IAA)	0.586	0.659
Sultan et al. [160]	0.592	0.887
Sultan et al. [160] w/ Question Demoting	0.571	0.903
Ramachandran et al. [138]*	0.610	0.860
Saha et al. [144]	0.570	0.902
Our Method	**0.590**	**0.847**

Table 5.5 Comparing the performance of different models trained using the dataset by Mohler et al. [111]. The comparison is based on Pearson's ρ correlation and Root Mean Square Error (RMSE). Ramachandran et al. [138]* reports results on a smaller test dataset, thus scores are not directly comparable.

Several studies have used the dataset of Mohler et al. [111] as a benchmark to evaluate their methods. We compare our results to the original results by Mohler et al. [111] and three other recent comparable works [138, 160, 144]. Mohler et al. [111] train Support Vector Machine (SVM) on a combination of graph-based alignment and lexical similarity measures to score short students' answers using a 5-point scale. They find that the supervised SVM model in this task outperforms the unsupervised alignment model [112]. Ramachandran et al. [138] adopt a mechanism to automate the generation of regular expression (regexp) text patterns from the reference expert answers as well as top-scoring students' answers, to capture the structural and semantic variations of good answers. Sultan et al. [160] train a supervised model, namely a ridge regression model, on a set of similarity and word embeddings features. They apply a question demoting technique in an attempt to reduce the advantage of repeating words provided in the question by re-computing similarity features after removing these words from both the reference answer and the student response. Their ablation study shows that applying question demoting results in 0.021 correlation improvement and only 0.016 reduction in Root Mean Square Error (RMSE). Saha et al. [144] propose a deep

[7] Learning rate = 0.3, momentum = 0.2

learning feature encoding model that combines partial similarities of tokens (Histogram of Partial Similarities or HoPS), its extension to part-of-speech tags (HoPSTags), question type information and sentence embeddings features.

Table 5.5 shows our results in comparison to previous approaches. Our approach exhibits superior performance over existing models in terms of RMSE, except for IAA, and we perform quite well in comparison to IAA and Sultan et al. in terms of Pearson's correlation. Although Ramachandran et al. report better results, however, their evaluation is based on much smaller test data (453 examples) and they use in-domain model training. As can be seen as well, our coverage and alignment features are proven to have a significant effect on improving the performance than when only considering BOW or Tf-Idf features in isolation. Further manual error analysis revealed that a substantial portion of the errors is due to unstructured answers and misspelling. Again, a more lenient matching mechanism, e.g., using edit distance or automatic spelling correction, may alleviate these errors.

5.4.3 SemEval-2017 Task 1: Cross-lingual and Monolingual STS

In an attempt to support the research efforts in STS, the SemEval STS shared Task [7] provides researchers an opportunity to develop and evaluate their STS approaches using benchmark datasets gathered from diverse domains. Given a pair of sentences, the task is to provide a similarity score on a scale of [0-5] according to the extent to which the two sentences are considered semantically similar, with 0 indicating that the semantics of the sentences are completely irrelevant and 5 signifying semantic equivalence. Final performance is measured by computing the Pearson's correlation (ρ) between machine-assigned semantic similarity scores and gold-standard scores provided by human annotators.

Since 2016, the SemEval STS task has been extended to involve additional subtasks for cross-lingual STS. Similar to the monolingual STS task, the cross-lingual task requires the semantic similarity measurement for two snippets of text that are written in different languages. In contrast to all previous editions of the task, the 2017 edition was organized into six secondary sub-tracks and a primary track, which is the average of all of the secondary sub-tracks results except English-Turkish, which was run as a surprise track. Secondary sub-tracks involve scoring similarity for monolingual sentence pairs in one language (Arabic, English, Spanish), and cross-lingual sentence pairs from a combination of two different languages (Arabic-English, Spanish-English, Turkish-English).

This section reports our STS-UHH team participation in the SemEval-2017 shared Task 1 of STS. Our participation involves both supervised and unsupervised systems explained previously in Sections 5.3.1, 5.3.2 and 5.3.3. The two systems are then combined to create an average ensemble to strengthen the similarity scoring performance. Since the task requires the final similarity score to be on the scale [0-5], we rescale our final score to fit in this range by multiplying the average of the two [0-1]-normalized scores (i.e., supervised ans unsupervised scores) by 5. Our models are mainly developed to measure semantic similarity between monolingual sentences in English. For the cross-lingual tracks, we leverage the Google translate API to automatically translate non-English sentences into English. We show that combining supervised and unsupervised models into an ensemble provides better results than when each is used in isolation. Out of 84 submissions, STS-UHH best multi-lingual system is placed 10^{th} with an overall primary score of 0.65, 5^{th} among 31 participating teams. In the following subsections, we describe our data preprocessing, experiment setup, and discuss our results.

Dataset

The STS dataset consists of sentence pairs gathered from various sources, see Table 5.6. The dataset was constructed based on a collection of existing datasets from tasks that are related to STS. It covers many topics, such as technology, business, science, politics and newswire, as well as sentences with different lengths. The similarity between each pair of sentences was rated on a [0-5] scale (low to high similarity) by human judges using Amazon Mechanical Turk.

We use all the previously released datasets of the English monolingual track since 2012 to train and evaluate our models. The final total number of training examples is 14,619. The English training data has the following average lengths: 2012 10.8 words, 2013 8.8 words, 2014 9.1 words, 2015 11.5 words, 2016 13.8 words. We use our preprocessing pipeline to tokenize, lemmatize, dependency parse, and annotate the dataset for lemmas, part-of-speech (POS) tags, and named entities (NE). Stopwords are removed for the purpose of alignment, topic modeling and Tf-Idf computation. For cross-lingual datasets, non-English sentences are translated into English using machine translation. To obtain embedded word representations for our unsupervised alignment approach, we use GloVe[8] pre-trained on 840B tokens (2.2M vocab) of external corpora including Gigaword and Wikipedia [129].

Experimental Setup and Discussion

Overall we submitted three runs: **Run1** uses the unsupervised approach discussed earlier in Section 5.3.3. For the supervised run, **Run2**, we fed the similarity-based and coverage-based features as described in Sections 5.3.1 and 5.3.2, in addition to the similarity scores from

[8] https://nlp.stanford.edu/projects/glove/

Dataset	**Source**	**#Pairs**
MSRpar	Microsoft research paraphrase corpus	1472
MSRvid	Microsoft research video description paraphrase corpus	1500
SMTeuroparl	European parliament translation	1143
OnWN	Sense definition pairs from OntoNotes and WordNet	2051
SMTnews	Post-edited translation of news	399
FNWN	Sense definition pairs from FrameNet and WordNet	184
Headlines	News headlines gathered from several different news sources	2493
Deft forum	Forum post sentences	443
Deft news	News summaries	299
Images	Images description	1500
Tweet news	A pair of news tweet and a comment on that particular news	745
Answers forums	Paired answers collected from the Stack Exchange Q&A websites	344
Answers students	Answer pairs from a tutorial dialogue system	750
Answer-answer	Paired answers collected from the Stack Exchange Q&A websites (enhanced quality)	249
Question-question	Paired questions collected from the Stack Exchange Q&A websites	201
Belief	Pairs from a dataset tagged with committed belief	359
Plagiarism	Collection of short answers to computer science questions that exhibit varying degrees of plagiarism from related Wikipedia articles	228
Postediting	Human correction of translations by post-editing	244

Table 5.6 STS training set sources (2012-2017).

System	Primary	Track 1 AR-AR	Track 2 AR-EN	Track 3 SP-SP	Track 4a SP-EN	Track 4b SP-EN-WMT	Track 5 EN-EN	Track 6 EN-TR
Avg. length	-	7.1	5.9-8.2	9.1	8.8-7.2	20.8-19.4	8.7	8.4-6.3
Run1	0.57	0.61	0.59	0.72	0.63	0.12	0.73	0.60
Run2	0.61	**0.68**	0.63	0.77	0.72	0.05	0.81	0.59
Run3	**0.65**	**0.68**	**0.66**	**0.80**	**0.73**	**0.21**	**0.82**	**0.63**
Basel.	0.53	0.60	0.51	0.71	0.62	0.03	0.73	0.54
BIT [174]	0.66	0.75	0.69	0.82	0.77	0.05	0.82	0.72
ECNU [162]	0.73	0.74	0.74	0.85	0.81	0.33	0.85	0.77
Top	0.73	0.75	0.75	0.85	0.83	0.34	0.85	0.77

Table 5.7 Results obtained in terms of Pearson's correlation over three runs for the six sub-tracks in comparison to the baseline, the best two performing systems, BIT and ECNU, and the top obtained correlation in each track. The primary score represents the weighted mean correlation with human judgment on tracks 1-5. In **boldface**, the best results that we obtained in each column. The first row shows the average length of sentences in words. For cross-lingual tracks, we mention the average length of sentences for both languages.

Run1 into a MLP[9] neural network. **Run3** provides an average over an ensemble of the models from Run1 and Run2, and two additional regression methods[10]: Linear Regression (LR) and Regression Support Vector Machine (RegSVM). To evaluate our preliminary pre-testing models, we perform 10-fold cross-validation.

We report our results in Table 5.7. In boldface, we highlight the best result that we achieved in each track. **Baseline** results are provided by the organizers. They simply measure the word overlap between each pair of sentences by calculating the cosine similarity between their binary-weighted words vector representations. We also included the **top** obtained result in each track, and the results by the best two performing systems ECNU [162] and BIT [174]. The best performing system is submitted by the ECNU team [162]. They first translate all sentences to English, and then use an ensemble of four deep neural network models, and three feature-engineered regression methods with features based on: word alignments, summarization, machine translation evaluation metrics, kernel similarities of bags of words, bags of dependencies, n-gram overlap, edit distances, longest common prefix/suffix/substring, tree kernels, and pooled word embeddings. The second best results are achieved by the BIT team [174]. They train a linear regression model with WordNet, weighted word embeddings and alignment features.

According to the results, we can make the following observations:

[9] Hidden layers = 2, Learning rate = 0.4, momentum = 0.2

[10] We used the WEKA [173] implementation with default parameters, if not mentioned otherwise

- Our results outperform the baseline provided by the task organizers for most tracks by a large margin. Note that the baseline achieves an average correlation of 53.7, ranking 23^{rd} overall out of 44 system submissions that participated in all tracks.
- The ensemble outperforms the individual ensemble members.
- Results obtained in monolingual, especially English, are markedly higher than in cross-lingual tracks. This is due to the noise introduced by the automatic translation.
- Results of Track 4b appear to be significantly worse compared to other tracks results. In addition to the challenges introduced by inaccurate translations, the difficulty of this track lies in providing longer sentences with less informative surface overlap between the sentences compared to other tracks. The average lengths of sentences, in words, are given in Table 5.7 (first row).

5.4.4 Community Question Answering and Answer Triggering

The task of Community Question Answering (CQA) is to re-rank a list of possible answers according to their relevance with respect to a given natural language question [116]. In contrast to IR, where the retrieval scope could be a mix of relevant and irrelevant documents, in CQA, we assume that all answers are related to the given question to a certain degree. From the output perspective, rather than retrieving full documents, QA applications retrieve precise and accurate answers only.

Answer Triggering (AT) is a subtask of QA [178] that was formulated rather recently. Besides extracting correct answers from a set of candidate answers (i.e., answer selection), answer triggering verifies whether a correct answer exists in the first place.

In this section, we provide a summary of our experiments on CQA and AT. We show that the integration of domain-specific graph features and dependency graph-based alignment features leads to more precise and accurate results on benchmark datasets compared to state-of-the-art models. We separate the discussion of QA and AT experiments into two subsections. We have omitted many implementation details, focusing only on the improvement of the results obtained after incorporating DDGs-based features.

Community Question Answering (CQA)

Our experiments for Community Question Answering (CQA) are part of our participation in the SemEval-2017 Task 3 of CQA Subtask A for the English language [116]. Given a question and its first ten answers, the task is to rank these answers according to their relevance to the given question. The data was provided by the task organizers. It was crawled from the Qatar Living forum, and organized as a set of seemingly independent question–answers threads. The answers are annotated as "Good", "PotentiallyUseful" or "Bad" with respect to the question that started the thread. The idea is to have the "Good" answers ranked above the "PotentiallyUseful" and "Bad" answers. To resolve the task, we develop a Support Vector Machine (SVM)-based classifier and use its confidence score for ranking. In the training/development phase, we initially define a generic baseline trained on traditional **string similarity features**, then we perform ablation tests on five sets of features removing one group of features at a time, and observe its contribution to the model.

The **embeddings features** represent the distance between the sentence embedding vectors of a question and a given answer. Sentence embeddings are composed by averaging the weighted word embeddings[11] of the words contained in each sentence. **DDGs features**[12] represent the similarity-based (limited to dependency structure, expansion features and PoSLCS), dependency graph coverage and alignment-based features described in Sections 5.3.1, 5.3.2 and 5.3.3, respectively. **Topic features** represent the distance between LDA topic-based vectors of a question and an answer. Finally, **keyword features** represent the shared keywords and named entities between a question and an answer. Results on development and test sets are shown in Table 5.8. The ablation test reveals that embeddings and DDGs feature sets have the greatest impact on the reranking performance. Eliminating one of both feature sets would decrease the Mean Average Precision (MAP) on average by $\approx 3.50\%$.

In response to the task instructions, we submitted one primary run, to be used for the official ranking of participants. The other two contrastive runs, are scored and released but not officially ranked. Out of 85 submitted runs, our primary system achieved a MAP score of 86.88, ranking $3rd$ among 22 teams.

Answer Triggering (AT)

This subsection presents a hybrid deep learning model for AT, which combines several dependency graph-based alignment features, namely graph edit distance, graph-based similarity and dependency graph coverage, with dense vector representation of the Q-A pair from a Convolutional Neural Network (CNN). Graph edit distance defines the cost of the least expensive sequence of edit operations that are needed to transform one graph, in our case a dependency parsing graph, into another. Final optimal costs are calculated using the assignment algorithm [115]. The motivation behind this hybrid model is to combine the strength of CNN in capturing the semantic similarity between a question and an answer,

[11] We use pre-trained Google word embedding model: https://code.google.com/archive/p/word2vec/

[12] Trained using Google News corpus and the unannotated dataset provided by the task organizers

	Development Set 2017 - Feature Ablation						
	MAP	**AvgRec**	**MRR**	**P**	**R**	**F1**	**Accuracy**
All Features	**65.50**	**84.86**	**71.96**	**58.43**	**62.71**	**60.50**	**72.54**
All - string features	65.53	84.90	72.19	57.84	62.71	60.18	72.17
All - embedding features	62.11	81.23	69.00	53.03	53.42	53.23	68.52
All - DDGs features	61.85	81.06	69.80	54.46	54.52	54.49	69.47
All - topic features	65.15	84.79	72.37	59.02	61.98	60.47	72.83
All - keyword features	65.73	84.65	71.94	57.98	62.59	60.20	72.25
Baseline	**53.84**	**72.78**	**63.13**	**-**	**-**	**-**	**-**

	Test Set 2017 - Submitted Runs						
	MAP	**AvgRec**	**MRR**	**P**	**R**	**F1**	**Accuracy**
Primary	**86.88**	**92.04**	**91.20**	**73.37**	**74.52**	**73.94**	**72.70**
Contrastive run 1	86.35	91.74	91.40	79.42	51.94	62.80	68.02
Contrastive run 2	85.24	91.37	90.38	81.22	57.65	67.43	71.06
Baseline	**72.61**	**79.32**	**82.37**	**-**	**-**	**-**	**-**

Table 5.8 Feature ablation results on development set, and official scores attained by our primary and contrastive submissions to the SemEval-2017 Task 3 SubTask A. From left to right, columns represent the measures we used to evaluate our performance: Mean Average Precision (MAP), Average Recall (AvgRec), Mean Reciprocal Rank (MRR), Precession (P), Recall (R), F1 (with respect to the Good class), and Accuracy.

and the ability of dependency graph-based features to examine their structural overlap. An overview illustration of our Joint-CNN model is shown in Figure 5.5.

The standard CNN model takes a sentence as an input, maps it to a matrix of fixed dimension, where each row represents a word vector, performs convolution followed by pooling, and classifies the sentence into one of the predefined classes by a softmax classifier [81]. Joint-CNN is an advancement where both a question and its candidate answer are used as input to the model. The convolution and pooling operations for the question and its answer are performed separately. However, the outputs of both are later concatenated and passed to a fully connected softmax layer whose output is the probability distribution over the classes "Trigger" or "Non-Trigger".

We train a standard CNN with two layers of convolution on top of word vectors obtained from an unsupervised neural language model[13]. We leverage mini-batch gradient descent (SGD), and backpropagation algorithm [69] to compute the gradients and update the weights. The Ada-delta [181] update rule is used to tune the learning rate. Next, max-pooling is applied and the sentence vectors for Q and A are generated. The outputs of the pooling layers p_Q and p_A are concatenated as $p = p_Q \otimes p_A$, and passed to a fully connected softmax layer that computes the probability scores over the two classes "Trigger" and "Non-Trigger". These

[13] Trained by Mikolov et al. [105] on 100 billion words of Google News, and are publicly available through: https://code.google.com/p/word2vec/

	Train	Dev	Test
# Questions	2,118	296	633
# Question w/o Answer	1,245	170	390
# Answers	1,040	140	293

Table 5.9 Statistics and data splits of the WIKIQA dataset.

Joint-CNN probability scores in addition to the dependency graph-based alignment features are used as features to train a logistic regression model. The optimal hyperparameters[14] are determined on the development data.

To train and evaluate our model, we use the WIKIQA dataset [178]. It consists of questions collected from the Bing query logs and candidate answers extracted from the summary paragraphs of the associated Wikipedia pages. The dataset includes questions for which there exists no correct answer, see Table 5.9. Lexical similarity between Q-A pairs in the WIKIQA dataset is also lower as compared to other answer selection datasets. In some cases, there is no lexical overlap at all, which makes the AT task even more challenging.

The results are summarized in Table 5.10. We run experiments with different feature map sizes and various ablation modalities. Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) are used to evaluate the answer sentence selection performance by evaluating the relative ranking of the candidate answers of each question. Following Yang et al. [178], we use the standard precision, recall and F-score measures to evaluate the AT classification performance. We compare our results to those obtained by four baselines: BM-25, embeddings features from both word2vec and GloVe, n-gram coverage up-to trigrams, and two recent works that have used the same dataset [78, 178]. Yang et al. [178] combine word matching features with CNN sentence semantic method. Jurczyk et al. [78] propose a CNN architecture with additional hidden layer and incorporate subtree matching mechanism to measure the contextual similarity between a Q-A pair.

As shown in Table 5.10, our Joint-CNN-100-FMap with graph-based features shows the best performance on all measures except recall on both the development and test sets. Although embedding features (GloVe) and Jurczyk et al. [78] achieve relatively higher recall, however, they yield a substantially lower precision as compared to our best model, which means a higher rate of false positives. Our best model Joint-CNN-100-FMap with graph-based features achieves superior results of 6.17%, 4.21% and 3.41% over the Joint-CNN model without graph-based alignment features with respect to F-score, MAP and MRR respectively.

[14] feature maps size (fMap) = 50, 100 or 150 , drop-out rate = 0.5, maximum epochs = 50, learning rate = 0.2, filter window size = 3, 4, α = 7, β = 5, λ= 2, m = 2

	Test set					Development set				
Model	**MAP**	**MRR**	**Precision**	**Recall**	**F-score**	**MAP**	**MRR**	**Precision**	**Recall**	**F-score**
Baselines										
BM-25	47.12	48.89	21.04	24.43	22.60	45.69	47.01	20.32	26.19	22.88
N-gram Coverage	51.02	53.49	24.47	28.59	25.24	52.89	49.09	25.73	29.11	27.31
Embeddings Feat. (w2v)	43.23	44.11	13.37	34.16	19.21	44.29	43.95	14.55	35.87	20.70
Embeddings Feat. (GloVe)	49.28	52.77	16.12	40.74	23.10	49.87	49.01	17.56	39.19	24.25
Our Models										
Joint-CNN (50-FMap)	62.18	63.22	28.90	31.28	30.04	61.23	65.20	33.01	26.98	29.69
Joint-CNN (150-FMap)	63.69	65.35	25.07	39.51	30.67	61.52	62.45	25.29	34.13	29.05
Joint-CNN (100-FMap)	63.72	65.67	23.21	48.15	31.33	62.20	62.50	25.33	**46.03**	32.68
+ Graph Edit Distance	65.40	67.08	33.18	30.04	31.53	64.87	65.22	32.53	36.28	34.30
+ Graph Similarity	66.48	68.28	25.00	43.21	31.67	68.10	67.44	34.85	36.51	35.66
+ Graph Coverage	**67.93**	**69.08**	**35.69**	39.51	**37.50***	**69.17**	**69.40**	**39.83**	37.30	**38.52***
State-of-the art (Answer Triggering)										
CNN-cnt+All [178]	–	–	28.34	35.80	31.64	–	–	–	–	–
CNN$_3$: max + emb+ [78]	–	–	29.43	**48.56**	36.65	–	–	–	–	–

Table 5.10 Evaluation results of answer selection and answer triggering on the development and test sets of the WIKIQA dataset. **FMap** denotes the size of the feature map. *A t-test confirms a statistically significant improvement in the AT performance over the baseline methods and two state-of-the-art models ($p < 0.05$).

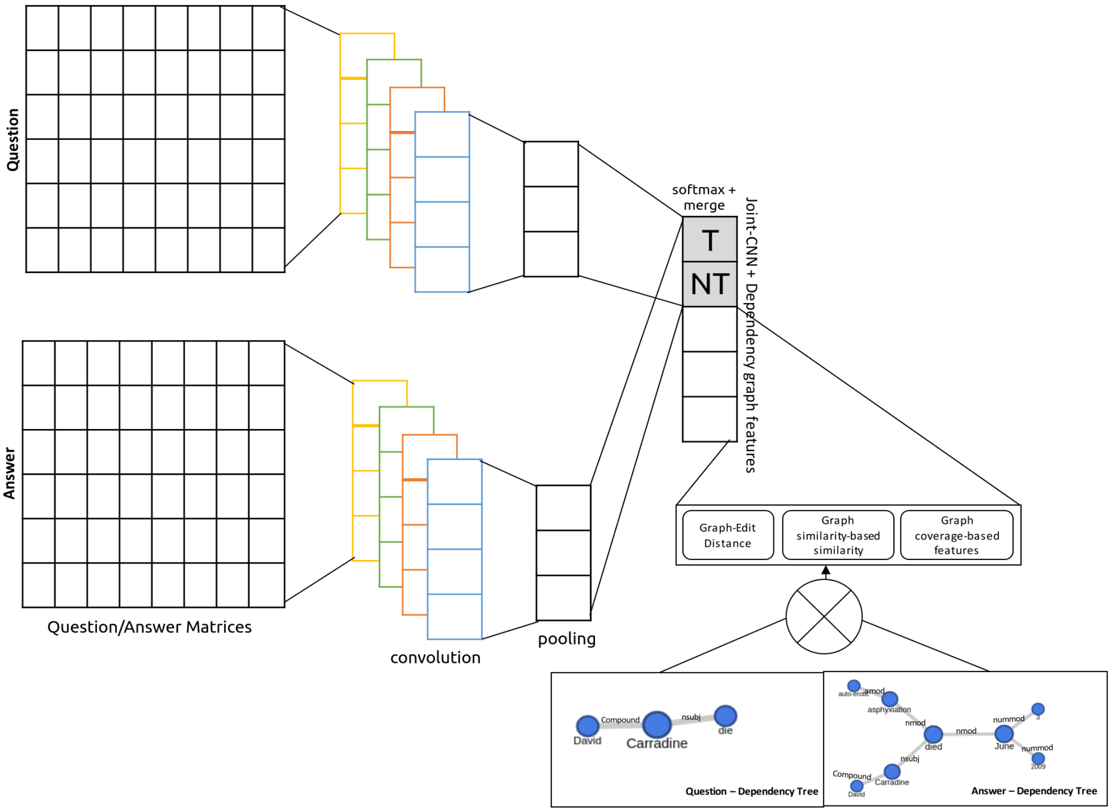


Fig. 5.5 Proposed Joint-CNN model architecture for answer triggering. The architecture combines mainly two components, the joint representation of a Q-A pair and their dependency graph-based features. Both components work independently by taking a question and an answer as input. The joint vector representation is then fed to a logistic regression classifier to predict the final label, either "Trigger" or "Non-Trigger".

Chapter 6

Conclusion

6.1 Summary

This dissertation is divided into two parts. The first part (Chapters 2 and 3) introduced a new generic and entirely data-driven approach to identify the most dominant concepts from multi-domain document collection and represent them as structured cohesive graphs, later called Domain Dependency Graphs (DDGs). The challenges were mainly on processing unstructured noisy text, and turning it into a structured representation, which incorporates enough structural and semantic information about the text, with less or no reliance on an external knowledge base. DDGs can encode structural (through dependency parsing) and shallow semantic information (through topic modeling), which are both necessary for future text processing tasks. To demonstrate the effectiveness and quality of our extracted DDGs, in the second part (Chapters 4 and 5), we performed an extrinsic evaluation by incorporating statistical DDGs-based features to improve the performance of two different Natural Language Processing (NLP) tasks, namely Aspect-Based Sentiment Analysis (ABSA) (covered in Chapter 4) and Semantic Textual Similarity (STS) (covered in Chapter 5). In both chapters, we started by introducing each task, reviewing the state-of-the-art and discussing related works. Following, we highlight the contents and outcome of each chapter:

Chapter 1 introduced the dissertation and presented a thorough discussion of classical text representation models, like the Bag-of-Words (BoW) model, and their notable weakness in capturing structural and semantic aspects of text content. We also discussed some recent structured representations, in particular graph-based approaches, and analyze their contribution and limitation. Based on that discussion, we gave further insights into possible improvements by summarizing our research contributions. We showed how a combination of a graph-based model together with topic modeling could lead to a superior text model representation. This combination forms the core contribution of this dissertation and later called Domain Dependency Graphs (DDGs). The main idea is to aggregate individual dependency relations between domain-specific content words, weigh them with Tf-Idf, and produce a DDG per topic by selecting the highest-ranked words and their dependency relations. The chapter closed with providing the outline for this dissertation and listing our related publications.

Chapter 2 presented DDGs, their mathematical definition, and further details on how they were constructed. First, we used LDA to extract dominant topics behind a corpus of documents. Then, source-side dependency structures of documents per topic were aggregated into one coherent DDG, which maintains the inter-topic cohesiveness together with the structural aspect of the text. Finally, an extra level of term and dependency weighting approach ensured the extraction of highly domain-specific words and relations.

Chapter 3 presented DDG_{viz}, an open-source web-based tool that supports interactive exploration and visualization of DDGs. DDG_{viz} enables users to filter, analyze and search generated DDGs by adjusting various parameters and configurations. The chapter described the back-end and front-end components of the DDG_{viz} and the detailed implementation of each component. It also showed several demonstrations of multiple NLP tasks and options.

Chapter 4 presented both supervised and unsupervised approaches to tackle several tasks in ABSA. We created a novel unsupervised graph-rule mining approach, which incorporates high-level structural linguistic information to accurately identify/extract the most compelling aspects of different entities and the corresponding sentiment word toward each extracted aspect from unstructured user-generated reviews. By comparing the extracted corresponding sentiment word against a sentiment lexicon, each extracted aspect-sentiment pair is then classified into one of the following polarity classes: "positive", "negative" or "neutral". To resolve the limitation and scarceness of existing sentiment lexicons, we introduced LexiExp, a tool for expanding existing sentiment lexicons based on the notion of distributional thesaurus. We also showed how LexiExp estimates polarities for the newly expanded lexicon using statistical co-occurrence calculation. The same tasks of aspect extraction and sentiment classification, in addition to a third task of Entity#Attribute detection (classifying the extracted expression to one or more predefined categories), were formulated again as supervised sequence labeling and classification problems. DDGs-based features, like DDGs top domain words and identified aspects, as well as distributional semantics and LexiExp features, were integrated into the feature space to train supervised models for the different subtasks. At the end of the chapter, we described an approach that uses DDGs-based relation co-occurrence statistics to calculate the association between explicit aspects and their corresponding opinion words and utilize it for implicit aspects identification. To evaluate, we experimented on four datasets from different sources, namely Amazon customer reviews dataset [100], Hu and Liu customer reviews benchmark dataset [70], the SemEval-2016 ABSA benchmark dataset [132], and sentiment analysis for Indian languages (SAIL) subtask benchmark dataset [128]. For each of the four datasets, we explained the preparation and preprocessing pipeline, the experimental setup, and finally discussed the obtained results. We compared our results to state-of-the-art approaches and performed feature ablation tests to examine the contributions of our DDGs-based features to the overall performance. We may summarize the overall results of this chapter as follows: (1) our unsupervised rule-based opinion expression extraction approach achieve better performance than state-of-the-art frequency-based approaches in terms of recall and precision, (2) using expanded lexicons by LexiExp improve the performance of sentiment polarity classification, (3) the combination of DDGs-based and LexiExp features contributes significantly to the performance of our supervised models for all tasks.

In **Chapter 5**, we proposed a supervised and unsupervised models for learning textual similarity, which can identify and score textual similarity between a pair of texts, and we presented a way of combining both models into an ensemble. We examined the impact of using dependency graph similarity and coverage features extracted from DDGs, and leveraged supervised machine learning techniques to improve semantic similarity predictions. We also introduced an approximate sub-graph alignment approach to find a sub-graph in the candidate text dependency graph that is similar to a given query text dependency graph, allowing for graph structural differences, as well as node gaps and mismatches, where a certain word in one dependency graph cannot be mapped to any word in the query text graph. To ensure that our method is generalizable and potentially adaptable to different languages, domains, topics and various text lengths, we conducted experiments using different corpora on four NLP tasks, namely document re-ranking, automatic short answer scoring, multilingual and cross-lingual STS, question answering and answer triggering. Results indicated that our approach provides better or comparable performance to baseline and recent approaches.

6.2 Future Works

During the course of this study, we have pushed many important but non-urgent ideas for the sake of time restrictions. The extensive error analysis we performed together with the time we spend "looking at the data", has improved our understanding of our failures and opened up our eyes to new and interesting ideas that we believe would be fruitful. In the following subsection, we highlight the potential follow-ups of our work as well as a brief outlook on future work we deem promising.

6.2.1 Domain Dependency Graphs (DDGs)

- The weighting scheme used for our DDGs described in Section 2.2.5 is quite basic. We hope to increase our weighting capability by extracting not only unique or distinctive nodes/relations, but also highly connected influential nodes that are located at the core of a given DDG. For future work, it is worth investigating more advanced ranking and filtering techniques such as eigenvector centrality [120, 104], PageRank [29] or core decomposition techniques [140, 150].

- Topic modeling, LDA in particular, reveals the hidden semantic structure underlying a collection of documents. However, this kind of semantic information is still shallow. To capture the complete semantics of the whole text, we have to consider many issues like co-references, pronouns, discourse relations, etc. We also believe that further integration of frame semantics theory and FrameNet [13] linguistic resource would increasingly enrich our DDGs representation. This can be verified by performing more experiments on semantic role labeling and learning semantic templates.

- We plan to provide a more precise evaluation of our DDGs. Throughout this dissertation, we only performed an extrinsic evaluation, while intrinsic evaluation, which measures the topical and graph coherence was not considered [41]. Additional important evaluation for the scalability and efficiency of our approach is needed. We have not yet analyzed how much parsing and topic modeling output quality affect the overall performance of the tasks build on top of our DDGs, e.g., our unsupervised rule-based OTE extraction approach.

- To date, our approach has been used to extract DDGs from a static corpus. We assume that a whole corpus is available before we start our process. However, in real cases, this may not always be possible, such as data is arriving in batches and it is constantly changing. For example, social media, such as Twitter and Facebook, generates large amounts of news feeds every hour. We think it would be valuable to find a way to integrating newly arriving text batches into available DDGs incrementally, without the need for restarting the process from scratch. This addition will enable us to automatically learn changes and shifts in topics represented by DDGs and track information genealogy [153].

6.2.2 *DDG_{viz}*

- We plan to conduct real-case and larger-scale scenario experiments with users to assess and provide a precise and realistic evaluation of the front end visualization.

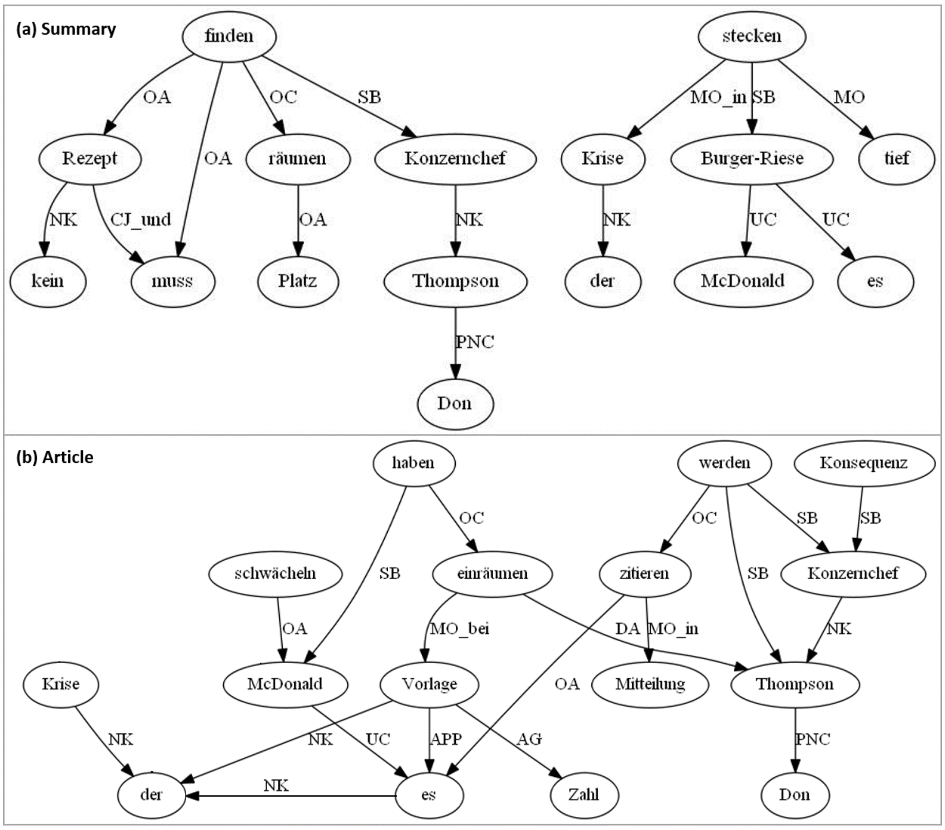


Fig. 6.1 Approximate sub-graph (bottom segment (b)) of a complete articles DDG when the full summary of an article is used as a query (top segment (a)).

- Future work will also explore the following possible extensions to the front-end interface: (a) provide additional statistical information about the graph; (b) disambiguate nodes with different word senses; (c) allow users to self-load a corpus and define the preprocessing pipeline by setting a specific config file.

- The resulted visualization of the "Search for Similarity" option shows only the subgraph of a DDG that approximately matches the query given by the user. We hope to improve the visualization output of this option by showing the detailed alignment between two dependency graphs, especially when a given query is not a list of keywords, but rather a full sentence. Figure 6.1 illustrates the approximate sub-graph (bottom segment (b)) of articles aggregate DDG when the full summary of one article is used as a query (top segment (a))[1]. The same example was used to demonstrate the keyword-based query DDG searching option in Chapter 3, see Figure 3.5.

[1] Original Article: spiegel.de/1015570

6.2.3 Aspect-Based Sentiment Analysis

- Our error analysis showed that most of the misclassifications of our unsupervised lexicon-based sentiment polarity classification approach are due to misclassified sentiment words expansions by LexiExp, in the first place. Words may also carry different polarities when used in different domains or with different aspects within the same domain, which we have not yet clearly resolved through our LexiExp pipeline. When using the same background corpus for probability distribution calculation, the same word will always be classified to the same polarity. Basically, the integration of DDGs-based topic features improves our supervised sentiment polarity classification accuracy, especially in cases where a word's polarity is domain-dependent, however, domain-dependent sentiment words within the same domain are still challenging, e.g., in the laptop domain "small" is negative when describing the storage aspect whilst positive when describing the size aspect, similarly, "cold food" vs. "cold drink" in restaurants domain. In the future, LexiExp could be integrated with a supervised domain-adaption approach that utilities DDGs to learn polarity shifts [177]. By learning separate projections of word embeddings for each domain (i.e., DDG), LexiExp will be able to capture shifts in polarities of individual words across domains (i.e., DDGs).

- Another direction is to develop our unsupervised rule-based OTE extraction approach to identify conditional sentences, e.g., "The camera works fine when the room is dark"; suggestion and presupposition, e.g., "phone is bit quieter than I would like" or "a larger compact-flash card won't hurt"; and comparative sentences, e.g., "I think the sound quality of this Nokia phone is better than my Samsung phone".

- Although we have already made slight progress in identifying implicit aspect, however, several issues remain unresolved. We have already discussed many of these issues at the end of Section 4.5.4. An additional issue is to identify implicit aspects that cannot be replaced by explicit aspects based on co-occurrence analysis, e.g., functional and tasty refers to functionality and taste, but phrases like "functional functionality" and "tasty taste" are not commonly used, thus, difficult to identify using our co-occurrence analysis approach.

6.2.4 Semantic Textual Similarity (STS)

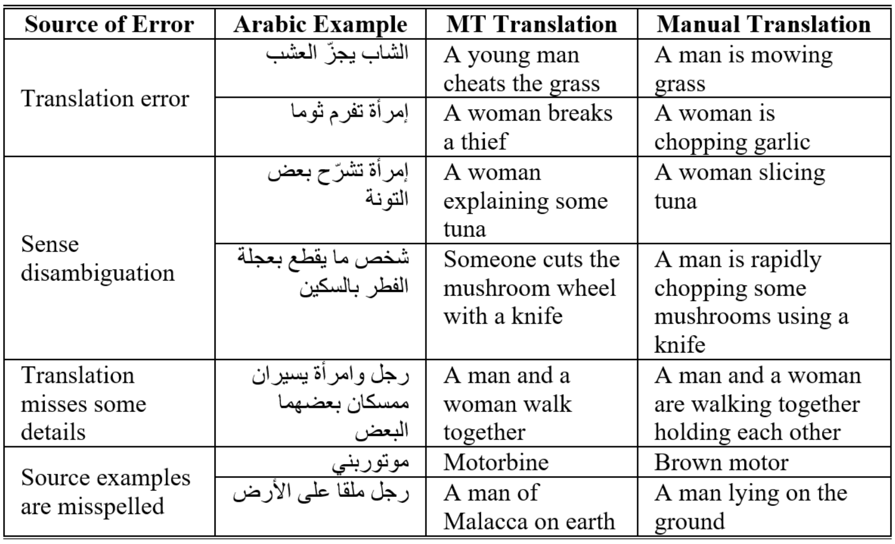
- For cross-lingual and non-English monolingual tracks, we used machine translation (MT) to translate text to English, given the high-availability and high-quality language processing tools and resources for the English language. However, the poor quality of

MT has introduced subtle errors. As is evident in Table 5.7, non-English monolingual tracks were less affected, but not completely unaffected, since the translations for both sentences are being compared. On the other hand, cross-lingual tracks results were significantly affected by translation quality. According to the task organizers, the Pearson correlation between the MT scores and the gold STS scores is 0.41 [7], which shows that translation quality measures and STS are only moderately correlated. In Table 6.1, we provide common sources of translation errors, and examples of Arabic sentences, from our previous SemEval experimentation, that were inaccurately or wrongly translated to English by the machine. Common translation error sources include, but not limited to: (1) *translation error*, in case the machine provides unjustifiable false translations; (2) *sense disambiguation*, where the machine fails to determine the correct meaning or sense of a word in a context; (3) *information lost*, when the target translation does not completely reflect the original input; (4) *misspelled or unstructured input*, if a translation input is not well-processed and pre-edited, it may lead to a poor or inaccurate translation.

Post-editing and manual translation are expensive and sometimes impossible due to disfluency. Following the success of deep learning approaches, in the future, we would like to explore possible approaches to learn multilingual word and sentence representations, and exploit rich multilingual knowledge bases, in particular BabelNet [119], to filling the gaps of low quality MT, resource-poor languages, Out-of-Vocabulary words (OOV) and sense disambiguation. BabelNet utilizes entity and word senses information from Wikipedia with and WordNet, in addition to many other resources such as Wiktionary and Wikidata to create a wide-coverage, multilingual semantic network of concepts.

- Our proposed approaches for approximate sub-graph matching and word alignment still demand further improvements, which include, capturing phrasal semantics, aligning multilingual terminology of cross-lingual pairs, finding optimal alignments effectively, considering advanced linguistic aspects for alignment like Part-of-Speech tags, entity types, etc. For languages like German, further consideration is required for handling splitting points of German compounds.

- The creation of solid semantic representations of entire text snippets is a central but challenging task. In Section 3.2.2, we used Skip-Thoughts to generate representations of sentences in the interests of performing sentence-sentence similarities to eliminating redundant sentences from the final summarization output. Since no official evaluation was made for the multi-document summarization task, it was excluded from the



Source of Error	Arabic Example	MT Translation	Manual Translation
Translation error	الشاب يجزّ العشب	A young man cheats the grass	A man is mowing grass
	إمرأة تفرم ثوما	A woman breaks a thief	A woman is chopping garlic
Sense disambiguation	إمرأة تشرّح بعض التونة	A woman explaining some tuna	A woman slicing tuna
	شخص ما يقطع بعجلة الفطر بالسكين	Someone cuts the mushroom wheel with a knife	A man is rapidly chopping some mushrooms using a knife
Translation misses some details	رجل وامرأة يسيران ممسكان بعضهما البعض	A man and a woman walk together	A man and a woman are walking together holding each other
Source examples are misspelled	موتوربني	Motorbine	Brown motor
	رجل ملقا على الأرض	A man of Malacca on earth	A man lying on the ground

Table 6.1 Sources and examples of machine translation errors of Arabic sentences from our SemEval STS Shared task experiments.

STS Chapter 5. In the future, we could complete an official evaluation of the text summarization approach. Based on this evaluation, we would examine the usage of Skip-Thoughts representations of phrases or intermediate nodes of dependency trees in order to improve our word-phrase and phrase-phrase similarity prediction.

In summary, our work opens up many interesting future research possibilities. DDGs-based features serves as general textual features that can be utilized in other NLP tasks like keyword extraction and domain information extraction.

References

[1] Abbasi, A., Chen, H., and Salem, A. (June 2008). Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. *ACM Transactions on Information Systems*, 26(3):12:1–12:34.

[2] Afzal, N., Wang, Y., and Liu, H. (2016). MayoNLP at SemEval-2016 Task 1: Semantic Textual Similarity based on Lexical Semantic Net and Deep Learning Semantic Model. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 674–679, San Diego, California.

[3] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., and Wiebe, J. (2015). SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pages 252–263, Denver, Colorado.

[4] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. (2014). SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 81–91, Dublin, Ireland.

[5] Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., and Wiebe, J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California.

[6] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 Shared Task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia.

[7] Agirre, E., Cer, D., Diab, M., Lopez-Gazpio, I., and Specia, L. (2017). SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

[8] Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada.

[9] Akhtar, M. S., Kohail, S., Kumar, A., Ekbal, A., and Biemann, C. (2017). Feature Selection Using Multi-objective Optimization for Aspect Based Sentiment Analysis. In Frasincar, F., Ittoo, A., Nguyen, L. M., and Métais, E., editors, *Natural Language Processing and Information Systems*, NLDB'17, pages 15–27, Liège, Belgium.

[10] Albalate, A. and Minker, W. (2013). *Semi-Supervised and Unsupervised Machine Learning: Novel Strategies*. John Wiley & Sons.

[11] Amiri, H., Resnik, P., Boyd-Graber, J., and Daumé III, H. (2016). Learning Text Pair Similarity with Context-sensitive Autoencoders. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL'16, pages 1882–1892, Berlin, Germany.

[12] Arun, R., Suresh, V., Veni Madhavan, C. E., and Narasimha Murthy, M. N. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. In Zaki, M. J., Yu, J. X., Ravindran, B., and Pudi, V., editors, *Advances in Knowledge Discovery and Data Mining*, pages 391–402, Berlin, Heidelberg. Springer Berlin Heidelberg.

[13] Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90, Montreal, Quebec, Canada.

[14] Balinsky, H., Balinsky, A., and Simske, S. (2011). Document sentences as a small world. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2583–2588, Anchorage, Alaska.

[15] Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

[16] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't Count, Predict! A Systematic Comparison of Context-counting vs. Context-predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1 of *ACL'14*, pages 238–247, Baltimore, Maryland.

[17] Bekoulis, G. and Rousseau, F. (2016). Graph-Based Term Weighting Scheme for Topic Modeling. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 1039–1044, Barcelona, Spain.

[18] Benikova, D., Yimam, S. M., Santhanam, P., and Biemann, C. (2015). GermaNER: Free Open German Named Entity Recognition Tool. In *GSCL*, pages 31–38, Duisburg-Essen, Germany.

[19] Biemann, C. (2009). Unsupervised Part-of-Speech Tagging in the Large. *Research on Language and Computation*, 7(2):101–135.

[20] Biemann, C. (2011). *Structure Discovery in Natural Language*. In Hirst, G., Hovy, E. and Johnson, M. (Series Eds.): Theory and Applications of Natural Language Processing. Springer, Heidelberg Dordrecht London New York.

[21] Biemann, C. and Riedl, M. (2013). Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. *Journal of Language Modelling*, 1(1):55–95.

[22] Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G. A., and Reynar, J. (2008). Building a Sentiment Summarizer for Local Service Reviews. In *WWW Workshop on NLP in the Information Explosion Era*, pages 14–23, Beijing, China.

[23] Blanco, R. and Lioma, C. (2012). Graph-based Term Weighting for Information Retrieval. *Information retrieval*, 15(1):54–92.

[24] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003a). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[25] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003b). Latent Dirichlet Allocation. *the Journal of machine Learning research*, 3:993–1022.

[26] Bohnet, B. (2010). Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China.

[27] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D^3 Data-Driven Documents. *IEEE Transactions on Visualization & Computer Graphics*, 17(12):2301–2309.

[28] Boyd-Graber, J., Mimno, D., and Newman, D. (2014). Care and feeding of topic models: Problems, diagnostics, and improvements. *Handbook of mixed membership models and their applications*, pages 225–255.

[29] Brin, S. and Page, L. (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

[30] Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

[31] Brun, C., Perez, J., and Roux, C. (2016). XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281, San Diego, California.

[32] Brychcín, T. and Svoboda, L. (2016). UWB at SemEval-2016 Task 1: Semantic Textual Similarity using Lexical, Syntactic, and Semantic Information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 588–594, San Diego, California.

[33] Buckley, C., Singhal, A., Mitra, M., and Salton, G. (1995). New Retrieval Approaches Using SMART: TREC 4. In Harman, D. K., editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 25–48, Gaithersburg, Maryland.

[34] Castillo, E., Cervantes, O., Vilariño, D., and Báez, D. (2016). UDLAP at SemEval-2016 Task 4: Sentiment Quantification Using a Graph Based Representation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 109–114, San Diego, California.

[35] Chambers, N. and Jurafsky, D. (2011). Template-based Information Extraction Without the Templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ACL-HLT '11, pages 976–986, Portland, Oregon.

[36] Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 288–296, Vancouver, British Columbia, Canada.

[37] Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., and Lin, C.-J. (2010). Training and Testing Low-degree Polynomial Data Mappings via Linear SVM. *Journal of Machine Learning Research*, 11:1471–1490.

[38] Chen, Y. and Skiena, S. (2014). Building Sentiment Lexicons for All Major Languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL'14, pages 383–389, Baltimore, Maryland.

[39] Chow, T. W. S., Zhang, H., and Rahman, M. K. M. (2009). A New Document Representation Using Term Frequency and Vectorized Graph Connectionists with Application to Document Retrieval. *Expert Systems with Applications*, 36(10):12023–12035.

[40] Chowdhury, G. G. (2010). *Introduction to Modern Information Retrieval*. Facet Publishing, 3rd edition.

[41] Christensen, J., Mausam, Soderland, S., and Etzioni, O. (2013). Towards Coherent Multi-Document Summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia. Association for Computational Linguistics.

[42] Cimiano, P., Unger, C., and McCrae, J. (2014). Ontology-Based Interpretation of Natural Language. *Synthesis Lectures on Human Language Technologies*, 7(2):1–178.

[43] Cortes, C. and Vapnik, V. (1995). Support Vector Machine. *Machine learning*, 20(3):273–297.

[44] Cozza, V. and Petrocchi, M. (2016). MIB at SemEval-2016 Task 4a: Exploiting lexicon based features for Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 133–138, San Diego, California.

[45] Crochemore, M. and Vérin, R. (1997). Direct Construction of Compact Directed Acyclic Word Graphs. In *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching*, CPM '97, pages 116–129, London, UK.

[46] Das, A. and Bandyopadhyay, S. (2010). SentiWordNet for Indian Languages. *Asian Federation for Natural Language Processing, China*, pages 56–63.

[47] Das, D. and Martins, A. F. (2007). A Survey on Automatic Text Summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4(192-195):57.

[48] Dechter, R. and Pearl, J. (1988). Network-Based Heuristics for Constraint-Satisfaction Problems. In *Search in Artificial Intelligence*, pages 370–425. Springer New York, New York, NY.

[49] Dehkharghani, R., Saygin, Y., Yanikoglu, B., and Oflazer, K. (2016). SentiTurkNet: A Turkish Polarity Lexicon for Sentiment Analysis. *Language Resources and Evaluation*, 50(3):667–685.

[50] Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271.

[51] Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. (2004). *Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools*, pages 127–148. Springer-Verlag, Berlin.

[52] Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

[53] Esuli, A. and Sebastiani, F. (2007a). Pageranking WordNet Synsets: An Application to Opinion Mining. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 7 of *ACL'07*, pages 442–431, Prague, Czech Republic.

[54] Esuli, A. and Sebastiani, F. (2007b). SentiWordNet: A High-Coverage Lexical Resource for Opinion Mining. *Evaluation*, 17:1–26.

[55] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of machine learning research*, 9(Aug):1871–1874.

[56] Ferrer i Cancho, R., Capocci, A., and Caldarelli, G. (2007). Spectral Methods Cluster Words of the Same Class in a Syntactic Dependency Network. *International Journal of Bifurcation and Chaos*, 17(07):2453–2463.

[57] Frey, B. J. and Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976.

[58] Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'13, pages 758–764, Atlanta, Georgia.

[59] Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

[60] Ghiassi, M., Skinner, J., and Zimbra, D. (2013). Twitter Brand Sentiment Analysis: A Hybrid System Using n-gram Analysis and Dynamic Artificial Neural Network. *Expert Systems with Applications*, 40(16):6266 – 6282.

[61] Giannakopoulos, G., Karkaletsis, V., Vouros, G., and Stamatopoulos, P. (2008). Summarization System Evaluation Revisited: N-gram Graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5:1–5:39.

[62] Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 759–765, Istanbul, Turkey.

[63] Gu, Y., Yang, Z., Zhou, J., Qu, W., Wei, J., and Shi, X. (2016). A Fast Approach for Semantic Similar Short Texts Retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 89–94, Berlin, Germany.

[64] Gupta, D., Kohail, S., and Bhattacharyya, P. (2018). Combining Graph-based Dependency Features with Convolutional Neural Network for Answer Triggering. In *The 19th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'18, Hanoi, Vietnam.

[65] Hagen, M., Potthast, M., Büchner, M., and Stein, B. (2015). Webis: An Ensemble for Twitter Sentiment Detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pages 582–589, Denver, Colorado.

[66] Hagen, M., Völske, M., Göring, S., and Stein, B. (2016). Axiomatic Result Re-Ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 721–730, Indianapolis, Indiana.

[67] Halliday, M. A. and Hasan, R. (1976). *Cohesion in English*. Longman's, London.

[68] Hamdan, H. (2016). SentiSys at SemEval-2016 Task 4: Feature-Based System for Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 190–197, San Diego, California.

[69] Hecht-Nielsen, R. (1989). Theory of the Backpropagation Neural Network. In *International 1989 Joint Conference on Neural Networks*, pages 593–605 vol.1. Washington, D.C.

[70] Hu, M. and Liu, B. (2004a). Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA.

[71] Hu, M. and Liu, B. (2004b). Mining Opinion Features in Customer Reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 755–760, San Jose, California.

[72] i Cancho, R. F. and Solé, R. V. (2001). Two Regimes in the Frequency of Words and the Origins of Complex Lexicons: Zipf s Law Revisited*. *Journal of Quantitative Linguistics*, 8(3):165–173.

[73] Jahren, B. E., Fredriksen, V., Gambäck, B., and Bungum, L. (2016). NTNUSentEval at SemEval-2016 Task 4: Combining General Classifiers for Fast Twitter Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 103–108, San Diego, California.

[74] Jakob, N. and Gurevych, I. (2010). Extracting Opinion Targets in a Single- and Cross-domain Setting with Conditional Random Fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045, Cambridge, Massachusetts.

[75] Jansen, B. J. and Spink, A. H. (2009). Investigating Customer Click Through Behaviour with Integrated Sponsored and Nonsponsored Results. *International Journal of Internet Marketing and Advertising*, 5(1/2):74–94.

[76] Jin, W., Ho, H. H., and Srihari, R. K. (2009). A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 465–472, Montréal, Canada.

[77] Jones, K. S. and Willett, P. (1997). *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

[78] Jurczyk, T., Zhai, M., and Choi, J. D. (2016). SelQA: A New Benchmark for Selection-Based Question Answering. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 820–827, San Jose, California.

[79] Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S., and Finin, T. (2014). Meerkat Mafia: Multilingual and Cross-Level Semantic Textual Similarity Systems. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 416–423, Dublin, Ireland.

[80] Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. (2008). *Visual Analytics: Scope and Challenges*, pages 76–90. Springer Berlin Heidelberg, Berlin, Heidelberg.

[81] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar.

[82] Kiritchenko, S., Zhu, X., Cherry, C., and Mohammad, S. (2014). NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 437–442, Dublin, Ireland.

[83] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought Vectors. In *Advances in neural information processing systems*, pages 3294–3302, Montréal, Canada.

[84] Kohail, S. (2015). Unsupervised Topic-specific Domain Dependency Graphs for Aspect Identification in Sentiment Analysis. In *Proceedings of the Student Research Workshop associated with RANLP*, pages 16–23, Hissar, Bulgaria.

[85] Kohail, S. and Biemann, C. (2017). Matching, Re-ranking and Scoring: Learning Textual Similarity by Incorporating Dependency Graph Alignment and Coverage Features. In *18th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'17, pages 377–390, Budapest, Hungary.

[86] Kohail, S., Salama, A. R., and Biemann, C. (2017). STS-UHH at SemEval-2017 Task 1: Scoring Semantic Textual Similarity Using Supervised and Unsupervised Ensemble. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 175–179, Vancouver, Canada.

[87] Kozareva, Z., Riloff, E., and Hovy, E. (2008). Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proceedings of the 46th. Annual Meeting of the Association for Computational Linguistics: Human Language. ACL-08: HLT*, pages 1048–1056, Columbus, Ohio.

[88] Kumar, A., Kohail, S., Ekbal, A., and Biemann, C. (2015). IIT-TUDA: System for Sentiment Analysis in Indian Languages using Lexical Acquisition. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 684–693, Hyderabad, India.

[89] Kumar, A., Kohail, S., Kumar, A., Ekbal, A., and Biemann, C. (2016). IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1129–1135, San Diego, California.

[90] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, Williamstown, MA.

[91] Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An Introduction to Latent Semantic Analysis. *Discourse processes*, 25(2-3):259–284.

[92] Li, F., Pan, S. J., Jin, O., Yang, Q., and Zhu, X. (2012). Cross-domain Co-extraction of Sentiment and Topic Lexicons. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 410–419, Jeju Island, Korea.

[93] Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

[94] Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

[95] Maks, I., Izquierdo, R., Frontini, F., Agerri, R., Vossen, P., and Azpeitia, A. (2014). Generating Polarity Lexicons with WordNet Popagation in 5 Languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1155–1161, Reykjavik, Iceland.

[96] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland.

[97] Markov, A. and Last, M. (2005). Efficient Graph-Based Representation of Web Documents. In *Proceedings of the 3rd International Workshop on Mining Graphs, Trees and Sequences*, MGTS'05, pages 51–62, Porto, Portugal.

[98] Markov, A., Last, M., and Kandel, A. (2008). The Hybrid Representation Model for Web Document Classification. *International Journal of Intelligent Systems*, 23(6):654–679.

[99] Marneffe, M., Maccartney, B., and Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 449–454, Genoa, Italy.

[100] McAuley, J. and Leskovec, J. (2013). Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172, Hong Kong, China.

[101] Mihalcea, R. (2004). Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Barcelona, Spain.

[102] Mihalcea, R., Corley, C., Strapparava, C., et al. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780, Boston, Massachusetts.

[103] Mihalcea, R. and Radev, D. (2011). *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, 1st edition.

[104] Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing Order into Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 404–411, Barcelona, Spain.

[105] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, Lake Tahoe, Nevada.

[106] Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

[107] Mimno, D., Wallach, H., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing Semantic Coherence in Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Edinburgh, UK.

[108] Miura, Y., Sakaki, S., Hattori, K., and Ohkuma, T. (2014). TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 628–632, Dublin, Ireland.

[109] Mohammad, S. (2012). #Emotional Tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada.

[110] Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a Word-Emotion Association Lexicon. *Computational Intelligence*, 29(3):436–465.

[111] Mohler, M., Bunescu, R., and Mihalcea, R. (2011). Learning to Grade Short Answer Questions Using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 752–762, Portland, Oregon.

[112] Mohler, M. and Mihalcea, R. (2009). Text-to-text Semantic Similarity for Automatic Short Answer Grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 567–575, Athens, Greece.

[113] Moscati, V. (2006). *The Scope of Negation*. PhD dissertation, University of Siena, Italy.

[114] Mueller, J. and Thyagarajan, A. (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2786–2792, Phoenix, Arizona.

[115] Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

[116] Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K. (2017). SemEval-2017 Task 3: Community Question Answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada.

[117] Nandi, T., Biemann, C., Yimam, S. M., Gupta, D., Kohail, S., Ekbal, A., and Bhattacharyya, P. (2017). IIT-UHH at SemEval-2017 Task 3: Exploring Multiple Features for Community Question Answering and Implicit Dialogue Identification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 90–97, Vancouver, Canada.

[118] Navigli, R. and Lapata, M. (2009). An Experimental Study of Graph Gonnectivity for Unsupervised Word Sense Disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692.

[119] Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network. *Artif. Intell.*, 193:217–250.

[120] Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

[121] Nielsen, F. Å. (2011). A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages:*, pages 93–98, Heraklion, Greece.

[122] Nivre, J. (2005). Dependency Grammar and Dependency Parsing. *MSI report*, 5133(1959):1–32.

[123] Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Paris, France.

[124] Okazaki, N. (2007). CRFsuite: A Fast Implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/.

[125] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia.

[126] Padó, S. and Lapata, M. (2007). Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

[127] Panchenko, A. (2014). Sentiment index of the Russian speaking Facebook. In *In Proceedings of International Conference on Computational Linguistics. Dialogue 2014*, pages 506–517, Moscow, Russia.

[128] Patra, B. G., Das, D., Das, A., and Prasath, R. (2015). Shared Task on Sentiment Analysis in Indian Languages SAIL Tweets - An Overview. In *Proceedings of the Third International Conference on Mining Intelligence and Knowledge Exploration - Volume 9468*, MIKE 2015, pages 650–655, Hyderabad, India.

[129] Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '14, pages 1532–1543, Doha, Qatar.

[130] Phan, X.-H. and Nguyen, C.-T. (2007). GibbsLDA++ is a C/C++ Implementation of Latent Dirichlet Allocation (LDA). http://gibbslda.sourceforge.net.

[131] Pilehvar, M. T. and Navigli, R. (2015). From Senses to Texts: An All-in-one Graph-based Approach for Measuring Semantic Similarity. *Artificial Intelligence*, 228:95–128.

[132] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). SemEval-2016 Task 5 : Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California.

[133] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado.

[134] Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval-2014)*, pages 27–35, Dublin, Ireland.

[135] Preece, J., Rogers, Y., and Sharp, H. (2001). *Beyond Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., New York, NY, USA.

[136] Prettenhofer, P. and Stein, B. (2010). Cross-language Text Classification Using Structural Correspondence Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1118–1127, Uppsala, Sweden.

[137] Radev, D. R. and Mihalcea, R. (2008). Networks and Natural Language Processing. *AI magazine*, 29(3):16–28.

[138] Ramachandran, L., Cheng, J., and Foltz, P. (2015). Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106, Denver, Colorado.

[139] Remus, R. (2012). Domain Adaptation Using Domain Similarity- and Domain Complexity-Based Instance Selection for Cross-Domain Sentiment Analysis. In *IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012)*, pages 717–723, Brussels, Belgium.

[140] Rousseau, F., Kiagias, E., and Vazirgiannis, M. (2015). Text categorization as a graph classification problem. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1702–1712, Beijing, China.

[141] Rubenstein, H. and Goodenough, J. B. (1965). Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.

[142] Ruppert, E., Klesy, J., Riedl, M., and Biemann, C. (2015). Rule-based Dependency Parse Collapsing and Propagation for German and English. In *GSCL*, pages 58–66, Duisburg-Essen, Germany.

[143] Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., and Andruszkiewicz, P. (2016). Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for Diversity; Combining Recursive Autoencoders, WordNet and Ensemble Methods to Measure Semantic Similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 602–608, San Diego, California.

[144] Saha, S., Dhamecha, T. I., Marvaniya, S., Sindhgatta, R., and Sengupta, B. (2018). Sentence Level or Token Level Features for Automatic Short Answer Grading?: Use Both. In *International Conference on Artificial Intelligence in Education*, pages 503–517, London, UK.

[145] Salameh, M., Mohammad, S., and Kiritchenko, S. (2015). Sentiment after Translation: A Case-Study on Arabic Social Media Posts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777, Denver, Colorado.

[146] Sandhaus, E. (2008). The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

[147] Schäfer, R. (2015). Processing and querying large web corpora with the COW14 architecture. In Bański, P., Biber, H., Breiteneder, E., Kupietz, M., Lüngen, H., and Witt, A., editors, *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3)*, pages 28 – 34, Lancaster, UK.

[148] Schenker, A., Last, M., Bunke, H., and Kandel, A. (2003). Clustering of Web Documents Using a Graph Model. In *Web Document Analysis: Challenges and Opportunities*, pages 3–18. World Scientific, London, UK.

[149] Schuster, S. and Manning, C. D. (2016). Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Language Resources and Evaluation (LREC)*, pages 23–28, Portorož, Slovenia.

[150] Seidman, S. B. (1983). Network Structure and Minimum Degree. *Social Networks*, 5(3):269 – 287.

[151] Severyn, A. and Moschitti, A. (2015). Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, Santiago, Chile.

[152] Shao, Y. (2017). HCTI at SemEval-2017 Task 1: Use Convolutional Neural Network to Evaluate Semantic Textual Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133, Vancouver, Canada.

[153] Shaparenko, B. and Joachims, T. (2007). Information Genealogy: Uncovering the Flow of Ideas in Non-hyperlinked Document Databases. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 619–628, San Jose, California.

[154] Singhal, A., Salton, G., and Buckley, C. (1996). Length Normalization in Degraded Text Collections. In *Proceedings of Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 149–162, Las Vegas, Nevada.

[155] Sparck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.

[156] Steyvers, M. and B Tenenbaum, J. (2005). The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive science*, 29:41–78.

[157] Stone, P. J. and Hunt, E. B. (1963). A Computer Approach to Content Analysis: Studies Using the General Inquirer System. In *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*, AFIPS '63 (Spring), pages 241–256, Detroit, Michigan.

[158] Sultan, M. A., Bethard, S., and Sumner, T. (2014). DLS@CU: Sentence Similarity from Word Alignment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 241–246, Dublin, Ireland.

[159] Sultan, M. A., Bethard, S., and Sumner, T. (2015). DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015*, pages 148–153, Denver, Colorado.

[160] Sultan, M. A., Salazar, C., and Sumner, T. (2016). Fast and Easy Short Answer Grading with High Accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'16, pages 1070–1075, San Diego, California.

[161] Talley, E. M., Newman, D. R., Mimno, D., Herr, B. W., Wallach, H. M., Burns, G. A. P. C., Leenders, A. G. M., and Mccallum, A. (2011). Database of NIH Grants using Machine-learned Categories and Graphical Clustering. *Nature Methods*, 8:443–444.

[162] Tian, J., Zhou, Z., Lan, M., and Wu, Y. (2017). ECNU at SemEval-2017 Task 1: Leverage Kernel-based Traditional NLP Features and Neural Networks to Build a Universal Model for Multilingual and Cross-lingual Semantic Textual Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, Vancouver, Canada.

[163] Titov, I. and McDonald, R. (2008). A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *Proceedings of the 46th. Annual Meeting of the Association for Computational Linguistics: Human Language. ACL-08: HLT*, pages 308–316, Columbus, Ohio.

[164] Tjong Kim Sang, E. F. and Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*, ConLL '00, pages 127–132, Lisbon, Portugal.

[165] Toh, Z. and Wang, W. (2014). DLIREC: Aspect Term Extraction and Term Polarity Classification System. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 235–240, Dublin, Ireland.

[166] Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*, ECML 2001, pages 491–502, Freiburg, Germany.

[167] Tutubalina, E. (2015). Target-Based Topic Model for Problem Phrase Extraction. In *Advances in Information Retrieval - 37th European Conference on IR Research*, ECIR 2015, pages 271–277. Vienna, Austria.

[168] Vázquez, S. and Bel, N. (2012). A classification of adjectives for polarity lexicons enhancement. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3557–3561, Istanbul, Turkey.

[169] Wan, X. and Xiao, J. (2008). Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860, Chicago, Illinois.

[170] Wei, C.-P., Chen, Y.-M., Yang, C.-S., and Yang, C. (2010). Understanding What Concerns Consumers: A Semantic Approach to Product Feature Extraction From Consumer Reviews. *Information Systems and e-Business Management*, 8:149–167.

[171] Weston, J., Bordes, A., Chopra, S., and Mikolov, T. (2015). Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *CoRR*, abs/1502.05698.

[172] White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., and Wagstaff, K. (2001). Multidocument Summarization via Information Extraction. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–7, San Diego, California. Association for Computational Linguistics.

[173] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[174] Wu, H., Huang, H., Jian, P., Guo, Y., and Su, C. (2017). BIT at SemEval-2017 Task 1: Using Semantic Information Space to Evaluate Semantic Textual Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84, Vancouver, Canada.

[175] Wu, Z. and Palmer, M. (1994). Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Las Cruces, New Mexico.

[176] Yang, S., Lu, W., Yang, D., Yao, L., and Wei, B. (2015a). Short Text Understanding by Leveraging Knowledge into Topic Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1232–1237, Denver, Colorado.

[177] Yang, Y. and Eisenstein, J. (2015). Putting Things in Context: Community-specific Embedding Projections for Sentiment Analysis. *arXiv preprint arXiv:1511.06052*, 4(3).

[178] Yang, Y., Yih, W. T., and Meek, C. (2015b). WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 2013–2018, Lisbon, Portugal.

[179] Yi, J., Nasukawa, T., Bunescu, R., and Niblack, W. (2003). Sentiment Analyzer: Extracting Sentiments About a Given Topic Using Natural Language Processing Techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 427–434, Melbourne, Florida.

[180] Yu, J., Zha, Z.-J., Wang, M., and Chua, T.-S. (2011). Aspect Ranking: Identifying Important Product Aspects from Online. Consumer Reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505, Portland, Oregon.

[181] Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701.*

[182] Zhai, Z., Liu, B., Xu, H., and Jia, P. (2011). Clustering Product Features for Opinion Mining. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011)*, pages 347–354, Hong Kong, China.