university of groningen

eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Master's Thesis

The Erasmus Mundus European Masters in

Language and Communication Technologies (EMLCT)

Faculty of Arts, University of Groningen

Faculty of Informatics, University of Basque Country

Using Linguistic Features and External Lexical Resources for Improving Sentence-level Sentiment Classification

By

PINAR ARSLAN

0707.pinar@gmail.com

Thesis Submission on

15th, August 2017

Supervisor: Malvina Nissim (University of Groningen)

Co-supervisor: Oier Lopez De Lacalle (University of Basque Country)

Table of Contents

List of Figures

List of Tables

Acknowledgements

I owe my deepest gratitude to my supervisors Malvina Nissim and Oier Lopez De Lacalle for their guidance and assistance throughout my thesis. This thesis could not have been completed without their support.

I would also like to thank Hessel Haagsma for his helpful explanations for my questions. He is a very enthusiastic and talented person to share knowledge.

Studying in the Erasmus Mundus Masters Program in Language and Communication Technologies has gained me great knowledge and people. Thanks to my professors and my colleagues in San Sebastian and Groningen, I enjoyed learning new skills and knowledge. My LCT experience will for sure be remembered in all of my present and future achievements nicely.

I would like to extend my deepest gratitude to my other half, Çeçe, who always believed in me and supported me. Without his love and encouragements, I could not have completed this thesis. I am also grateful to have great parents who made me feel happy with their sincere supports.

Abstract

This study aims to ascertain whether certain grammatical and lexical features improve sentence level sentiment analysis (i.e. classifying sentences with the polarity labels: positive, neutral and negative). We performed a sentence-level binary (i.e. positive or negative) sentiment classification on three datasets. We built four baseline systems (i.e. Support Vector Machine-based, Naïve Bayes-based, lexicon-based and majority-class baselines). Our research questions are: (i) whether modal verbs and other linguistic features (e.g. negation words, adversative conjunctions), and (ii) external lexical resources (e.g. SentiWordNet) contribute to the performance of our models built by a non-linear Support Vector Machine with radial basis function kernel. Our models were either cross-validated (i.e. 10-fold) or trained and tested on different datasets. Our findings show that the best accuracy scores were obtained with the aid of cross-validation. We also obtained high accuracy scores when the models were trained on large-scale datasets and tested on different datasets. Our models outperformed the baseline systems. We unveiled that unigram models with the features represented at word level performed satisfactorily. Modal verbs did not have a positive impact on the performance of our models. The adversative conjunctions improved the performance of our models only for a large-scale dataset. The negation words increased the accuracy score of our models for the cross-validated datasets, which were either middle or large-scale. The SentiWordNet sentiments were helpful in sentence-level sentiment analysis when training and test sets were in different domains.

Keywords: Sentence-level Sentiment Classification, Opinion Mining

Chapter 1

1. Introduction

Sentiment Analysis, also referred to as opinion mining, is used to explore opinions of individuals on specific entities. Introduction of Web 2.0 (DiNucci, 1999) platforms including blogs, discussion forums and any other sorts of social media, rendered the access to large amount of information online possible. This has led to an increased speed with which information spreads over the Internet. Thus, consumers are now able to share their experiences and opinions about products and/or services much more easily on online platforms. The amount of opinions available online is informative to both companies and consumers. Consumers' opinion about a particular product is important when one wants to purchase the particular product. Keeping track of what customers think about a product also plays a major role in strategies companies develop. However, given the amount of opinions and reviews available, manually processing those data to determine the sentiment of each review is not time effective for consumers or companies. Therefore, a great deal has been invested on automatic sentiment classification systems. So far sentiment classification has been worked at different levels such as document level and sentence level. Diverse approaches have been tried on the sentiment classification tasks such as lexicon-based approaches and machine-learning approaches. Various lexical resources on sentiment have been used, see, for instance, Nissim and Patti (2016) who review many highly useful lexical resources to be used in the sentiment analysis, such as Opinion Lexicon, compiled by Bing Liu, and SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010). The bulk of the research in sentiment analysis has gone into producing lexica, into exploring features, into tuning algorithms, but in comparison little has been done towards exploring some more linguistic aspects that might intuitively play a role in how the sentiment is expressed and can be captured. Modality and negation, which are the two grammatical phenomena, have been an increasingly emerging area of the sentiment analysis research. In this thesis, we focus on sentence level binary sentiment classification (i.e. positive or negative). Our overarching research questions are (1) whether or not modal verbs (e.g. might, could) improve the accuracy of our Support Vector

Machine-based models implemented with a non-linear radial basis function kernel; (2) whether negation words (e.g. no, not), adversative conjunctions (e.g. but, yet, though) and SentiWordNet sentiments (e.g. positive, negative) in our models improve the sentiment accuracy. We investigated (3) whether the domain and/or size of datasets have an impact on the accuracy scores of the baseline systems as well as our proposed models, and whether the similarity or differences in the domain and size of the training and test sets affects the performance of our models for the test sets. To this end, we use three datasets in different domains and size used as training and test sets. Finally, we examined (4) whether different levels of feature representation (e.g. word level and part-of-speech tag level) and different n-gram models (e.g. unigram and high-order n-gram models) affect the sentiment accuracy of our models. We explain background and related work in Section 2, methods in Section 3, data and preprocessing in section 4, models in Section 5, experimental settings and results in Section 6 and discussion in Section 7 followed by conclusion in Section 8.

Chapter 2

2. Background and Related Work

In the following sections, sentiment analysis and the linguistic structures under investigation in this thesis including modality and negation are introduced. In section 2.1. sentiment analysis is explained and relevant research on the sentiment classification, which is based on supervised machine-learning approaches, are reviewed. In the section 2.2. modality and negation are briefly explained from both theoretical linguistic perspectives and computational perspectives where we mention how modality and negation can be utilized in sentiment analysis systems. In the section 2.3. some open issues about sentiment analysis are discussed.

2.1. Sentiment Analysis

Sentiment Analysis has been extensively studied recently. Sentiment Analysis determines whether or not a text expresses an opinion. The texts expressed an opinion are classified as subjective polarity while the factual texts are classified as objective polarity. Sentiment Analysis determines whether the subjective text expresses a positive or negative opinion of people about various entities such as a product, a service or a person. The opinion can be sometimes neutral too, though this is often confounded with objective in the subjective and objective distinction. Sentiment analysis can be administered at (i) document level and (ii) sentence level as well as (iii) sub-sentence level (e.g. words and phrases). In the following subsections, we explain related work about these levels of sentiment analysis.

2.1.1. Document-level Sentiment Analysis

In a document level sentiment classification, the objective is to classify the whole document as positive, negative or neutral.

Pang, Lee, and Vaithyanathan (2002) who worked on binary sentiment classification (i.e. positive or negative) at document level used movie reviews (i.e. 700 positive and 700 negative movie reviews) from the Internet Movie Database (IMDb) archive. Pang et al. (2002) represented each word situated

between a negation word and the upcoming punctuation marker with the label 'NOT_' (e.g. I do not NOT_like NOT_this NOT_fake NOT_news) in their models. They ran their models built by Naïve Bayes (NB), Support Vector Machines (SVMs) and maximum entropy (ME) applied to the 3-fold cross-validated movie reviews. The accuracy scores of their random-choice and two human-selected unigram baselines were between 50% and 69%. A unigram model (with negation tagging) built by the SVMs returned the highest accuracy score of 82.9%. The negation tagging was claimed to have a negligible but on average positive, effect (Pang et al., 2002). They conclude that the unigram model with the negation tagging had a positive impact on the document-level sentiment analysis.

Tsutsumi, Shimada, and Endo (2007) also studied the binary sentiment classification (e.g. positive or negative) of the movie review documents from IMDb used by Pang et al. (2002). Tsutsumi et al. (2007) ran their models built by SVMs, ME and score calculation techniques employed on the 3-fold cross-validated movie reviews for the document level sentiment classification task. Tsutsumi et al. (2007) used part-of-speech tags by giving weighted scores for adjectives. For feature selection, they applied a chi-square test so that they eliminated words with low reliability. They labeled for negation words such as 'not' and 'never'. They employed three methods containing two voting processes 'naïve voting', 'weighted voting' and integration with SVMs in order to combine the three classifiers (i.e. SVMs, ME and score calculation). The model built by the multiple classifiers with the integration method with SVMs got the accuracy score of 87.1%. Their findings indicate that using multiple classifiers would give a good accuracy score for a document-level sentiment analysis when they were combined with even a simple weighting method.

Moraes, Valiati, and Neto (2013) studied document-level sentiment analysis by comparing the application of commonly used learning techniques SVM and NB with Artificial Neural Networks (ANN) applied to movie reviews (Pang et al., 2002) and collections of product reviews obtained from amazon.com, which were GPS, books and cameras. Each dataset comprising 2000 documents with labeled 1000 positive and 1000 negative sentences was 10-fold cross-validated. Their main focus was to compare non-linear radial basis

function (rbf) SVM with ANN classifier. They removed stopwords and used a stemmer (e.g. Snowball stemmer) in the preprocessing step. Information gain filtered the features that were unigram-based and weighted by the Term Frequency Inverse Document Frequency (TF-IDF). The information gain (IG) decreased the running time of SVM and increased the practicality of ANN training. Accuracy scores of the model built by the ANN with 1000 terms selected in the context of balanced data were 87.3% for the GPS dataset, 81.8% for the books dataset, 90.3% for the cameras dataset, and 86% for the movies dataset. When the training datasets were highly unbalanced, the SVM performed better than the ANN except in the Movies reviews (Pang et al., 2002). Their results show that the SVM was less prone to be affected by 'less informative terms' (i.e. noisy terms) than ANN in the unbalanced data while the ANN outperformed the SVM for most tests performed on four datasets in the context of balanced data, which shows that ANN is a useful approach in the task of sentiment learning.

2.1.2. Sentence-level Sentiment Analysis

In sentence-level sentiment analysis, the focus is to classify each sentence in a review as positive, negative or neutral.

Meena and Prabhakar (2007) worked on sentence-level sentiment classification using car reviews data containing 5000 positive and 5000 negative pre-labeled sentences gathered from diverse car review sites (e.g. motortrend.com, wardsautoworld.com). The authors used POS-Tagging, word dependencies and dependency trees for each sentence in order to conduct a number of linguistic analyses. For example, they applied conjunction analysis to see if conjunctions influenced the overall semantic orientation of a sentence. That is, when a conjunction was found in a sentence, all of the phrases and words holding a polarity in the sentence were detected and then their polarity was measured by applying default polarity calculation (DPC) method along with the aid of General Inquirer (GI) word list or WordNet so that they revealed the sentiment of the words. When they could not find words in the GI list, the synonyms of the words were sought in WordNet and then the sentiments of synonyms were found on the GI word list. They took into account the impact of negation words as well. The accuracy score of their best system was 78%. They

conclude that both conjunction analysis and General Inquirer (GI) with WordNet improved the accuracy of the sentiment classification.

Hu and Liu (2004) performed sentence-level sentiment classification using 100 reviews for each of the five products, digital cameras (e.g. 'Canon G3' and 'Nikon coolpix 4300'), cellular phone, mp3 player and DVD player) gathered from amazon.com and C|net.com by using a feature-based summarization system in which they explored the features of each product (e.g. how much a cell phone weights, etc.). The authors focused on only adjectives. They categorized the polarity of each adjective as positive or negative. To do so, the authors used a set of seed adjectives with pre-labeled polarities so as to be able to extend the existing set by using the set of adjective synonym as well as the set of adjective antonym in Wordnet (Fellbaum, 1998). Then the authors applied a bootstrapping technique. They also detected adjacent adjectives (i.e. effective opinions) modifying a frequent feature that was usually either a noun or noun phrase. Both adjectives and effective opinions played an important role for their system to guess the semantic orientation of sentences holding an adjective. In the prediction of semantic orientation of an opinion sentence, Hu and Liu (2004) followed three different approaches: When a sentence had more positive opinion words than negative ones, then the polarity of the sentence would become positive or negative in the other way round. When the number of positive and negative opinion words was same, the authors considered the average semantic orientation of effective opinions of the product features in the sentence. This was because effective opinions carried the most related sentiment for a feature. For all other cases, they used contextual information to find out the semantic orientation of a sentence because the reviewers were considered to explain their opinions about one of the products in a few sequential sentences. The accuracy score of their proposed model was 84%. They conclude that utilizing WordNet to cover the semantic orientations of more adjectives has been useful in the sentence-level sentiment classification task.

Liu, Yu, Liu, and Chen (2014) investigated modality in sentence-level sentiment analysis by applying both binary classification (i.e. positive or negative) and three-class classification (i.e. positive, negative or neutral). The authors used 1635 customer review sentences with four categories of products

(e.g. vacuum cleaner, cellphone, mattress, hair care) out of 9152 sentences. Sentences without a modal form were discarded. The authors used a unigram-based Support Vector Machine. The authors used Delta TF-IDF, a technique used for weighting word scores before classification (Martineau & Finin, 2009). To find the semantic orientation of the sentences with modalities, Liu et al. (2014) used the Stanford log-linear part-of-speech tagger on the preprocessed dataset. The authors' main focus was to extract modal verbs and opinion-bearing words following the modal verbs (e.g. should laugh), which the authors called 'modality sequence'. Then, they categorized the modalities as subjunctive mood, deontic modality and dynamic modality. For negation words, a binary feature was used so that they found out if a sentence contains a negation word or not adjacent to an opinion-bearing word by setting the window size to four words. Additionally, the authors paid attention to adversative conjunctions combining two clauses. They found out the semantic orientation of the whole sentence by reversing the polarity of prior clause. Some punctuation markers were also used as features. The accuracy score of two-way classification (i.e. binary classification) obtained was 76% while the accuracy score of three-way classification was 61%. The authors conclude that the accuracy scores of the unigram-based model significantly outperformed the scores of three baseline methods (e.g. lexicon-based, Standard SVM, Naïve Bayes). Additionally, Liu et al. (2014) explained that the two-way classification had higher performance than the three-way classification since the three-way classification was more difficult than two-way classification in general. After all, they expressed that both modality sequence and modality categories were useful in the sentiment analysis for the sentences with modalities.

Conducting sentiment analysis at sentence level, Jain, Colaco, and Rodrigues (2016) investigated sentences containing a modal verb to find out whether or not the modal verbs influence the binary-sentiment classification (e.g. positive or negative) of sentences. They ran their model built by a decision list classifier data employed on Amazon customer reviews of four products (e.g. jewelry, watches, cell phones, software). They used 1000 sentences out of 5000 sentences as a training set since only the 1000 sentences contained a modal verb. Jain et al. (2016) removed stop words before they used a POS-tagging (i.e.

Stanford part-of-speech tagger) to the dataset. The authors followed the rules about the modality created by Liu et al. (2014). Upon detecting the sentences containing modal verbs (e.g. should, could), Jain et al. (2016) aimed to find out the sequences of modal verbs (e.g. should laugh) and modal categories (e.g. deontic or dynamic modalities) with the aid of categorization rules by Liu et al. (2014). Jain et al. (2016) took into account Delta TF-IDF scores before classification. As a test set, 100 random sentences related to any of the four products were used. The accuracy score of their proposed model was 72%. They conclude that their proposed model covering the modality was effective to determine whether the semantic orientation of a sentence was positive or negative.

2.1.3. Sub-sentence-level Sentiment Analysis

For sub-sentence-level sentiment analysis, the focus is to classify the semantic orientations of sub-sentences as positive, negative or neutral. There is less information at the sub-sentence level than sentence level or document level.

Zirn, Niepert, Stuckenschmidt, and Strube (2011) employed sub-sentence-level binary sentiment classification. With the aid of the discourse parser HILDA (Duverle & Prendinger, 2009), the review text was split into discourse segments constructing the entities their system classified. They used a subset of the Multi-Domain Sentiment Dataset containing product reviews collected from amazon.com. They involved 120 product-related reviews comprising 7376 segments. They used various pre-compiled sentiment lexicons such as SentiWordNet (SWN) (Baccianella et al., 2010) which is an external lexical resource assigning sentiment scores to positive, negative and neutral sentiments for WordNet (Fellbaum, 1998) synsets. The discourse parser HILDA (Duverle & Prendinger, 2009) was also used in order to find out the discourse relations between segments. Specifically, Zirn et al. (2011) differentiated the relations ‘contrast’ and ‘ncontrast’, which was used for all other relations, by modeling these two relations in Markov logic framework. Markov logic was utilized to integrate sentiment scores from various sentiment lexicons (e.g. SentiWordNet, Unigram Lexicon) and explicitly use information about the structure of text to determine the sentiment of text segments. The authors also

modeled neighborhood relations in which they showed which segment was preceded by which neighboring segment in Markov logic formulation (MLN_neigborhood). Then, structural information about neighboring segments was found out. In general, they aimed to learn weights for the formulas in the Markov logic model. Their best model was the one built on neighboring relations in Markov logic formulation, which gave 69% accuracy score. This result was a bit unexpected for them since Zirn et al. (2011) thought that discourse relations (i.e. contrast relation) would have greater relation with the polarity of segments than neighboring relations. The authors conclude, "The relation between contrast relation and the change of polarity is not as close as they had expected" (Zirn et al., 2011, p. 343). Their main conclusion is the use of general structures detected in the text contributed to the results.

Chen et al. (2015) performed sub-sentence-level polarity classification (i.e. positive, neutral, negative) as well as subjectivity classification (i.e. subjective or objective) using a joint framework for sentiment analysis at sub-sentence level with Markov logic. By using the powerful statistical-relational representation language (i.e. Markov logic), which first order logic was combined with Markov networks, the authors conducted the experiments on the 10-fold cross-validated dataset of Chinese sentiment corpus (Tan & Zhang, 2008) with three categories 'hotel', 'book' and 'notebook computer'. Chen et al. (2015) annotated the dataset at sub-sentence level. 300 documents comprising 4642 sub-sentences were used for the sentiment analysis task. The authors applied word segmentation and part-of-speech tagger for the dataset in the preprocessing step. Their joint model was compared to four various models (e.g. baseline system, Base_MLN, Pipeline_MLN). The results of the joint model outperformed the four models. The accuracy score was 63.43% for the polarity classification task. The authors claims that they can easily integrate all kinds of useful features, such as adjective, adverb, negation, and sentiment scores, into their joint framework, which is considered as an advantage of their proposed model.

2.2. Modality and Negation

In this section, we describe two concepts 'modality' and 'negation' from different perspectives: theoretical linguistic perspective (2.2.1.) and computational perspective (2.2.2.).

2.2.1. Brief Theoretical Linguistic Perspective

In this section, we present some brief linguistic perspective about 'modality' and 'negation' providing details about the interaction between them.

Morante and Sporleder (2012, p. 224) describes modality as "a grammatical category that allows the expression of aspects related to the attitude of the speaker towards her statements in terms of degree of certainty, reliability, subjectivity, sources of information, and perspective". In the analysis of modality some binary distinctions have been made such as 'realis' and 'irrealis' or 'non-modal' and 'modal' or 'declarative' and 'non-declarative' or 'subjunctive' and 'indicative' or 'factual' and 'non-factual' or 'real' and 'unreal'. The binary system of Realis and Irrealis is a typical characteristic of mood (Palmer, 1986). According to Mithun (2001, p. 173), "the realis portrays situations as actualized, as having occurred or actually occurring, knowable through direct perception" whereas "the irrealis portrays situations as purely within the realm of thought, knowable only through imagination". A similar description, "realis refers to the real world, and irrealis refers to possible worlds", is given by Roberts (1990, p. 367).

English has a set of modal verbs (e.g. can, could, may, might, must, will, would, should, shall, ought (to)). Jespersen (1924, p. 329) makes a distinction between "categories containing an element *will*" and "categories containing *no element of will*". Palmer (1986) names these elements as *propositional modality* and *event modality*. According to Palmer (1986, p. 8), propositional modality is "concerned with the speaker's attitude to the truth-value or factual status of the proposition" while event modality "refers to events that are not actualized, events that have not taken place but merely potential". To show the difference between the two-modality types, Palmer (1986, p. 7) gives the following examples.

1) a. Kate *must* be at home now. (Propositional modality)
   b. Kate *must* come in now. (Event modality)

The examples in (1a) and (1b) show the modal verb 'must' can be seen in two different types of modality: propositional (1a) paraphrased as 'It is necessarily the case that Kate is at home now' and event modality (1b) paraphrased as 'It is necessary for Kate to come in now'. Palmer suggests that propositional modality, includes two sub-categories, which are *epistemic* and *evidential*. He defines epistemic modality as "speakers express their judgments about the factual status of the proposition" while according to this point of view, evidential is referred to as a modal form that is used "to indicate the evidence that speakers have for its factual status" (Palmer, 1986, p. 8). Although Palmer takes evidential to be a modal category, De Haan (1999) takes evidential to be a deictic category but not primarily modal category. Also, Lyons (1977, p. 793), for instance, expresses that epistemic modality deals with "the speaker's opinion or attitude towards the proposition that the sentence expresses or the situation that the proposition describes". These two kinds of propositional modality are illustrated in (2a) and (2b).

2) a. This dishwasher *may* not worth € 750. (Epistemic)
   b. Students *must* have submitted their homework. (Evidential)

In (2a), the speaker expresses judgments about the price of a dishwasher. Thus, (2a) has an epistemic connotation as it directly involves the speaker's attitude. In (2b), by contrast, the speaker makes assumptions on whether that students submitted their homework or not. Therefore, (2b) is evidential rather than epistemic since it relates to the information source.

Palmer (1986, p. 9) distinguishes two types of event modality, which are *deontic* and *dynamic.* Deontic modality is referred to as "the conditioning factors are external to the relevant individual". Dynamic modality, by contrast, indicates, "the factors are internal to the individual". Deontic modality is related to extrinsic obligation or permission while dynamic modality is associated with

intrinsic ability or willingness as illustrated in (3a)-(3d), taken from (Palmer, 1986, p. 10).

3) a. John *may/can* come in now. (permission)
   b. John *must* come in now. (obligation)
   c. John *can* speak French. (ability)
   d. John *will* do it for you. (willingness)

The examples above in 3a and 3c show that the modal verb 'can' might own more than one interpretation: either permission (3a) or ability (3c), which make predictability of modal categories hard in terms of writing grammatical rules for modal verbs and corresponding modal categories. Palmer (1986, p. 10) also explains that the category 'ability' in dynamic modality needs to be interpreted widely since a dynamic modal verb (e.g. can) may "indicate not merely ability, but also the possibility in a more general sense" as he illustrated in (4a)-(4c).

4) a. He *can* go now. (Deontic: I give permission)
   b. He *can* run a mile in five minutes. (Dynamic: he has the ability)
   c. He *can* escape. (Dynamic: the door's not locked)

The terms used for the types of the modal systems (e.g. epistemic, deontic and dynamic) which are taken from the work on 'modal logic' written by Von Wright (1951, pp. 1-2) might contain some overlapping modal verbs. For instance, the examples above in (4a)-(4c) illustrate that the modal verb 'can' belongs to more than one modal category: either deontic (4a) or dynamic (4b and 4c). Namely, one modality marker (e.g. can) may be used in multiple senses.

Another grammatical category, negation, reverses the truth value of a proposition (Morante & Sporleder, 2012, p. 224). Some grammatical devices are used to express different types of negation in most languages such as 'clausal negation', which entire propositions are negated, and 'constituent negation', which constituents of clauses are negated (Morante & Sporleder, 2012). According to van der Wouden (1997), negation can also be expressed by a variety of grammatical ways such as verb semantics (e.g. hate) , noun semantics

(e.g. happiness), prepositions (e.g. without), conjunctions (e.g. nevertheless). Negation might also emerge as an affix (e.g. **im**polite, speech**less**).

A different concept, 'negative polarity' interrelates with the grammatical category 'negation'. Israel (2004, p. 701) describes polarity "as such polarity encompasses not just the logical relation between negative and affirmative propositions, but also the conceptual relations defining contrary pairs like hot-cold, long-short, and good-bad". Although negation can flip the polarity of an expression, the bond between negation and polarity is not straightforward. Negation alters polarity not only from positive to negative but also negative to neutral or positive as exemplified in (5a) and (5b) by Morante and Sporleder (2012, p. 251).

5) a. This is *not* a *good* camera. (positive to negative)
   b. This is *by no means* a *bad* camera. (negative to positive)

In (5a), the negator 'not' changes the polarity of expression "camera's being good" and the polarity of this sentence becomes negative. In (5b), the negator 'by no means' changes the polarity of the expression "camera's being bad" and the polarity of this sentence becomes positive (or neutral depending on the context used).

In the following section we briefly explain how modality and negation are used in computational linguistics field, especially in sentiment analysis.

2.2.2. Brief Computational Perspective

Taking into account information about modality and negation brings a noticeable contribution to many applications such as machine translation (Baker et al., 2014), trustworthiness detection (Su, Huang, & Chen, 2010) and sentiment classification (Liu et al., 2014). Baker et al. (2014) investigated modality and negation in the machine translation tasks. They constructed a modality annotation scheme, modality lexicon and two automated modality taggers. Their goal was firstly to detect modalities because they thought that modalities could make distinctions between actualized events and non-actualized events or between belief and certainties or between positive instances of entities and

negative ones. Secondly, three components of modality were detected, which were a trigger (i.e. word(s) expressing modality), a target (i.e. the thing that the modality scopes over) and a holder (i.e. experiencer of the modality). Baker et al. (2014) focused on the scope of modality and negation and the interaction between modality and negation with some linguistic simplifications and some menu choices for modality annotation (e.g. 'H is able to [make P true/false', 'H is not able to [make P true/false'). Their focus was on factivity-related eight modalities which some of them might also be tagged for sentiment or evidentiality. The three of the eight modalities 'success', 'ability' and 'requirement' are illustrated in (6a)-(6c).

6) a. Success: does H succeed in P?
   b. Ability: can H do P?
   c. Requirement: does H require P?

Their publicly available English modality lexicon[1] produced semi-automatically was created to annotate a corpus with modality information automatically. The lexicon entries consisted of several elements: a string of one or more words such as can and might, a part of speech for each word (e.g. MD for can), a modality type (e.g. Able, NotAble, Belief, NotBelief), a head word (or trigger) (e.g. hope in hope for) and one or more subcategorization codes from syntactic codes provided in Longman's Dictionary of Contemporary English (Paul, 1978). The modality lexicon and the modality taggers (e.g. structure-based tagger) were used to improve machine translation output (Baker et al., 2014).

Negation and modality play an important role in sentiment analysis tasks (Liu et al., 2014). Negation is named as a 'polarity influencer', an element that has a power to flip the polarity of an expression (Wilson, Wiebe, & Hoffmann, 2009, p. 402). Similar to negation, modal verbs can also change the polarity of an expression as illustrated in (7a) and (7b).

[1] Website of modality lexicon: http://www.umiacs.umd.edu/~bonnie/ModalityLexicon.txt

7) a. The woman *should* have been happy.

b. The woman has *not* been happy.

The examples above show that modality in (7a) and negation in (7b) change the polarity of expressions. In (7a), the sentence containing 'should', base form of verb 'have', past participle of the verb 'be' and an opinion bearing adjective 'happy' holds a negative polarity. Although it has no negative element, the modal verb 'should' used with past participle form of verb has changed the polarity of the expression the woman's being happy which is positive. In (7b), the sentence holds a negative polarity since the negation marker 'not' changes the polarity of expression the woman's being happy'.

In terms of investigation of modality and negation in sentiment analysis, different approaches have been presented. Pang et al. (2002) used negation as a prefix attached right before words in a bag-of-words model (e.g. NOT_good, NOT_bad). Similar approach was used by Das and Chen (2001) adding a negation tag (i.e. –-n) for negation words. According to Pang et al. (2002), adding such additional features like negation did not bring to a great extent contribution to sentiment analysis tasks due to increased feature space and data sparseness. Kennedy and Inkpen (2006) used negation elements such as 'not' and 'never' as polarity shifters, reversing the semantic orientation of a particular word. Wilson, Wiebe, and Hoffmann (2005) presented specific negation features in their phrase-level sentiment analysis work. Wilson et al. (2005, p. 352) explained their approach towards negation: "Negated is a binary feature that captures whether the word is being locally negated: its value is true if a negation word or phrase is found within the four preceding words or in any of the word's children in the dependency tree, and if the negation word is not in a phrase that intensifies rather than negates (e.g., not only). The negated subject feature is true if the subject of the clause containing the word is negated". Differently, negation models have been used as a strength determiner of opinions (Choi & Cardie, 2010; Popescu & Etzioni, 2005). Modality features have been made use in sentiment analysis work as well. Liu et al. (2014) focused on modality sequence so that they extracted a modal verb (e.g. should) and an opinion bearing word (e.g. win) from sentences. They made use of modality categories of the detected

modalities (e.g. subjunctive mood, deontic modality and dynamic modality). They took into account not only modality but also negation. For negation words, they used a binary feature to find out if the sentence contained a negation word or not (see section 2.1.2. for more explanation). Jain et al. (2016) also considered modality as a powerful polarity shifter that might reverse the polarity of sentences. They adopted exactly the same approach towards modalities in sentiment analysis as Liu et al. (2014).

2.3. Open Issues

Previous studies in the field of sentiment analysis have more dominantly investigated document-level sentiment analysis. Exploring the polarity of a document seems to be easier than seeking to find the polarity at sentence or sub-sentence level. When a text contains more words, there will be more useful clues to be used in the sentiment analysis tasks. This thesis focused on sentence-level sentiment analysis. When the polarity of a sentence is accurately predicted, there is no doubt that the performance of document-level sentiment analysis tasks will show an improvement as well.

Lexicon-based approaches are usually compared to machine learning approaches by using the former approach in a baseline system and the latter approach in a proposed system (Liu et al., 2014). However, not many studies combine the strengths of lexicon-based and machine-learning approaches in the sentiment analysis tasks. Our study also incorporates lexicon-based approach to machine-learning approach to explore whether or not the two approaches together improve sentiment analysis results.

More linguistic analyses (e.g. negation, adversative conjunctions) have been performed in sentiment analysis tasks in recent years. Modality has been performed in sentence-level sentiment analysis tasks (Jain et al., 2016; Liu et al., 2014). However, these studies disregard the sentences without a modal verb. Also, they aim to find out the modal categories automatically by following the categorization rules about modality. This approach may not give accurate modal categories since a modal verb (e.g. can) can belong to more than one modality types (e.g. deontic and dynamic). Such a modality categorization requires more comprehensive semantics analysis so that correctly modal categories can be

labeled. As for our approach, we investigate a binary sentiment classification by not disregarding any sentences, which means we employ the sentiment analysis in the sentences with/without a modal verb. When a modal verb is detected in a sentence, we use either some modality tags taken from the English Modality Lexicon (Baker et al., 2014) or apply some generalization tags. Before we tag modal verbs, part-of-speech tagging is employed to avoid any ambiguous cases (e.g. 'can' has two word categories: noun and modal verb). From this point of view, our proposed approach avoids labeling modal verbs inaccurately but still it contains semantics in the tags used for modal verbs to some extent.

In the following sections, we explain methods (3), the datasets that we have used and preprocessing (4), models (5), experimental settings and results (6), discussion (7) and conclusion (8).

Chapter 3

3. Methods

This project investigates a sentence-level sentiment classification (or sentiment analysis) for three review-based datasets including 'Review' (Blitzer, Dredze, & Pereira, 2007), 'Amazon' (Kotzias, Denil, De Freitas, & Smyth, 2015) and 'YelpIMDB' (Kotzias et al., 2015) datasets. We conduct a binary sentiment classification task, which the polarity of a sentence is either positive or negative. We investigate some linguistic analyses for modal verbs, negation words and adversative conjunctions. Our linguistic analyses are based on some linguistic generalizations. What we mean with generalization is to label the linguistic elements such as modal verbs and adversative conjunctions with some normalized and generalized labels. For the modal verbs, we use different degrees of feature generalizations. Some of these labels have a close relation to semantics while some of them have no strong relation with semantics. Our main research question is whether or not the labels used for the modal verbs improve the sentiment analysis results. We also aim to see the contributions of the labels used for other linguistic features (e.g. negation, adversative conjunctions) on the accuracy scores of the sentiment analysis task. Additionally, we train and test our models on different datasets in this research. This is because we are interested to see whether using training sets, whose domains and sizes show differences or similarities compared to test sets would affect the performance of our models for the test sets.

In some of our proposed models, we incorporate machine learning into a lexicon-based approach. Specifically, a Support Vector Machine (SVM) (Cortes & Vapnik, 1995) is used as a machine-learning learning technique. SentiWordNet v3.0 (Baccianella et al., 2010) is utilized as a publicly available lexical resource. In our unigram-based models, we use three different levels of feature representations: word level (e.g. good), part-of-speech-tag level (e.g. JJ), word along with part-of-speech-tag level (e.g. good_JJ) since we aim to find out which feature representation improves the accuracy of our models at most. Our proposed models differ in a few aspects from one another. Different n-gram ranges are applied on our models so that we find out which n-gram models give

the best accuracy score for the sentiment analysis task. We use several tools and resources to build our models. For example, SentiWordNet v3.0 is used to find out the semantic orientations of opinion-bearing words as a lexical resource. Natural Language Tool Kit (NLTK) (Loper & Bird, 2002) is utilized to carry out some linguistic analyses such as part-of-speech-tagging. Additionally, we use scikit-learn (Pedregosa et al., 2011), a free software machine-learning library for the Python programming language in order to conduct machine learning classifiers.

Finally, we compare our proposed models to more than one baseline. Some baselines are built with supervised machine learning classifiers and some of them are focused on the frequency of sentiment labels in the datasets so that we obtain multiple accuracy scores from the baseline systems. As well as the comparison of our proposed models to the baseline systems, we compare one of the best models with the rest of the proposed models for each of the three datasets. Furthermore, we uncover the contribution of each of the labels used for the linguistic features (e.g. modal verbs, negation) on the accuracy score of the best model with the model comparison strategy.

In the following sections, we elaborately explain our approaches, models, baseline systems, datasets and so on. In section 4, we present what datasets we have used and how we have preprocessed them. In section 5, we uncover what baseline systems we have created, how we have represented features and what components are comprised in our models. Experimental settings and results are reported in section 6. We discuss our findings and related work in section 7. We conclude our study in the section 8.

Chapter 4

4. Data and Preprocessing

In this section we introduce three review-based datasets used for the sentence-level binary sentiment classification task. We show the number of words, modal verbs, sentences, and sentences with a modal verb and an adversative conjunction within positive and negative reviews for each dataset. We compare the distributions of positive and negative review sentences across modality for each dataset. We explore whether or not the modal verbs have any association with the polarities: positive or negative polarity. Upon introducing each dataset in the section 4.1, we briefly explain how we preprocess our datasets in the section 4.2.

4.1. Datasets

We conduct several datasets. The first one is called "Review dataset" which is a subset of a larger collection of customer reviews collected from Amazon.com (Blitzer et al., 2007). The second dataset is "Amazon review dataset" which is a small collection of customer reviews from Amazon.com (Kotzias et al., 2015). The third dataset is the concatenation of two datasets: Yelp and IMDB datasets extracted from the reviews of restaurants and movies, respectively (Kotzias et al., 2015). All of the three datasets are based on reviews and well balanced in terms of the distribution of positive and negative classes. The domains of the review-based datasets show differences. For instance, the Amazon and Review datasets contain product-related reviews while the YelpIMDB dataset comprises movie and restaurant-related reviews. Another difference among the datasets is that they are in different size. 'Review' dataset comprises the highest number of sentences in comparison with the two other datasets. Our intention is to use these different-sizes datasets since we wanted to see to what extent the data size brings a contribution to accuracy scores of our models.

Although we do some analyses on different datasets such as 'Subtask A training dataset of SemEval 2016 Task 4 (Nakov, Ritter, Rosenthal, Sebastiani, & Stoyanov, 2016)' and 'SemEval 2015 Task 11 Training dataset (Ghosh et al., 2015)', we did not want to use them in this sentiment classification task. This is

because they are not well balanced in terms of the distribution of positive and negative classes. SemEval 2015 Dataset contains 6419 negative sentences and 432 positive sentences. Similarly, SemEval 2016 Dataset comprises 762 negative sentences and 3540 positive sentences. In this research, we are interested in having well-balanced datasets offering linguistically rich features such as modal verbs and adversative conjunctions. Therefore, we use the three datasets (i.e. the Review, Amazon and YelpIMDB datasets) for the sentence-level sentiment classification task by considering the distribution of the sentiment classes (e.g. positive or negative) and linguistically richness of these datasets.

4.1.1. Review Dataset

The first dataset, 'Review dataset', is a subset of a larger collection of customer reviews collected from Amazon.com (Blitzer et al., 2007). The original dataset has 25 categories, while our subset, preprocessed by Richard Johansson, has six. This dataset contains 6000 documents manually labeled as 'positive' or 'negative'. It comprises the customer reviews of six categories: 'music', 'health', 'DVD', 'software', 'books' and 'camera'. Upon splitting each document into sentences, we obtain 47457 sentences. Since the sentiment labels are annotated at document level, we assume that every sentence in a document carries the same polarity label as the label of the document. In Table 1 we summarize the number of documents, sentences and words as well as the number of sentences with a modal verb (e.g. can, may, could) and an adversative conjunction (e.g. but, however, yet, although, still) for the Review dataset.

Table 1: *Count of documents, sentences, words, sentences with modality and an adversative conjunction for the Review dataset*

Sentiment Orientation	Documents	Sentences	Words	Sentences with modality	Sentence with conjunction
Positive	2968	22843	401136	4289	3902
Negative	3032	24614	407736	4945	4073
Sum	6000	47457	808872	9234	7975

The numerical values in the Table 1 show that the 'Review' dataset contains a similar amount of positive and negative labels (i.e. 22843 positive and 24614 negative sentences). The numbers of positive and negative sentences with a modal verb do not show a significant difference. We also investigate each of the modal verbs available in the positive and negative Review sentences. In Table 2 we show not only the raw counts of the modal verbs but also the proportions of the modal verbs. The proportions are shown inside parentheses. To obtain the proportion of the modal verbs, we divide the raw counts of the modal verbs by the total number of the modal verbs in a sentiment class (e.g. the modal verb 'would' in 'positive' class: 1051 /5228 = 0.20).

Table 2: *Absolute numbers of frequency of each modal verb along with their proportions inside parentheses for the Review dataset*

Modals	**Positive**	**Negative**	**Sum**
May	262 *(0.05)*	240 *(0.04)*	502 *(0.04)*
Might	131 *(0.03)*	233 *(0.04)*	364 *(0.03)*
Will	1300 *(0.25)*	1319 *(0.22)*	2619 *(0.23)*
Would	1051 *(0.20)*	1574 *(0.26)*	2612 *(0.23)*
Can	1456 *(0.28)*	1384 *(0.23)*	2625 *(0.23)*
Could	477 *(0.09)*	716 *(0.12)*	1193 *(0.11)*
Should	317 *(0.06)*	415 *(0.07)*	732 *(0.06)*
Ought	8 *(0.001)*	6 *(0.001)*	14 *(0.001)*
Must	221 *(0.04)*	213 *(0.04)*	434 *(0.04)*
Shall	5 *(0.001)*	6 *(0.001)*	11 *(0.001)*
Need	0 *(0.00)*	1 (0.0002)	1 *(<0.001)*
Sum	5228	6107	11335

The distributions of the proportions of the positive and negative sentences are not significantly different in the use of modal verbs (X^2 (10) = 0.023, $p > 0.05$) for the Review dataset. Still, some of the modal verbs seem to be biased towards one of the two polarities: positive or negative polarity. The modal verbs 'would' (X^2 (1) = 28.03, $p < 0.01$) and 'could' (X^2 (1) = 14.32, $p < 0.01$) show higher association with the negative polarity. The modal verbs 'can' (X^2 (1) = 27.49, $p <$

0.01) and 'will' (X^2 (1) = 12.72, $p < 0.01$) seem to be significantly biased towards the positive polarity. We also want to visualize how the raw counts of the modal verbs are distributed in the positive and negative sentences for the Review dataset so that we once again see the modal verbs showing higher association with a specific polarity (e.g. positive or negative) on a chart. Figure 1 displays the distributions of the positive and negative Review sentences across the modal verbs.

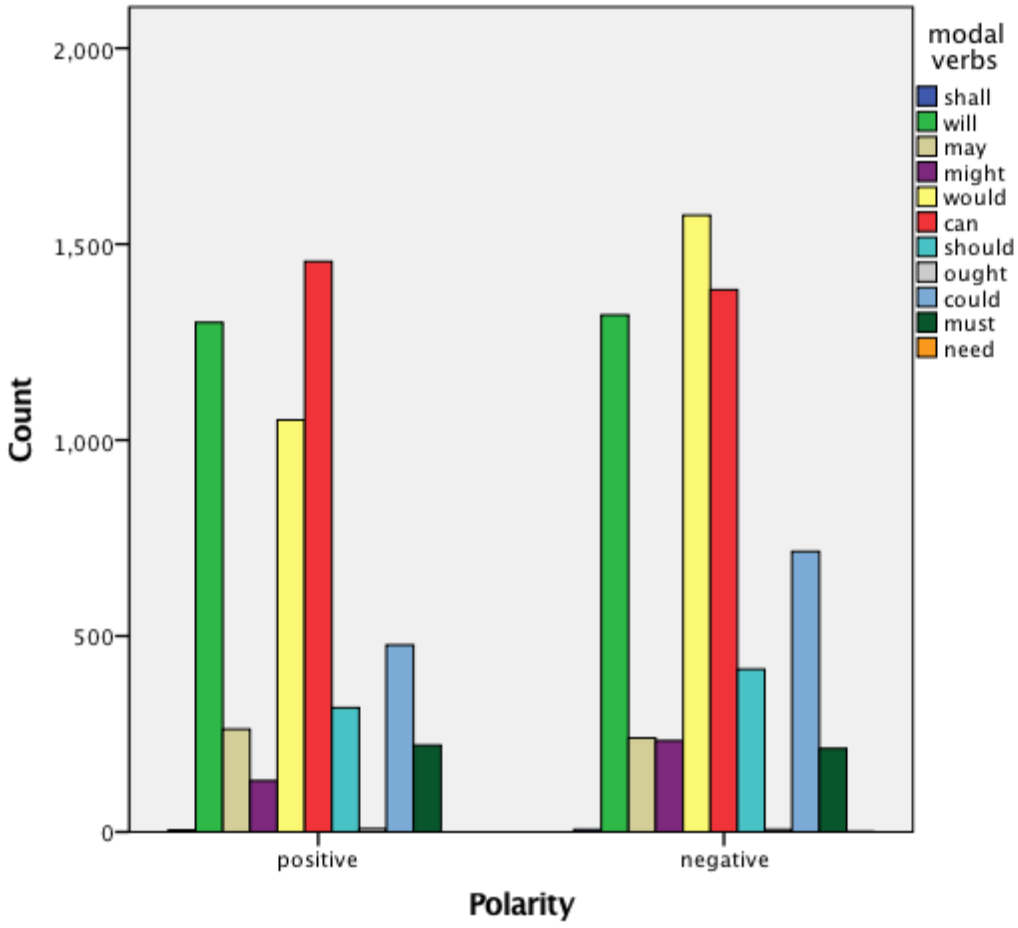

Figure 1: *The distribution of the positive and negative sentences across the modal verbs for the Review dataset*

This visualization of the distributions of the raw counts of the modal verbs also shows which modal verbs (e.g. could, would, can and will) have higher association with which polarities (e.g. positive or negative).

4.1.2. Amazon Dataset

The second dataset, Amazon dataset, extracted from the reviews of products contains 1000 sentences with manually labeled 500 positive and 500 negative labels (Kotzias et al., 2015). Because of the same amount of polarity classes (500 positive, 500 negative), we can consider this dataset as a balanced dataset. In this dataset, each sentence has a sentiment score, either 0 (for negative) or 1 (for positive). Table 3 displays the number of sentences and words as well as the number of sentences containing a modal verb (e.g. can, could) and an adversative conjunction (e.g. but, yet, still) in the Amazon dataset.

Table 3: *Count of sentences, words and sentences with modality and an adversative conjunction in the Amazon Dataset*

Sentiment orientation	Sentences	Words	Sentences with modality	Sentences with conjunction
Positive	500	4937	48	31
Negative	500	5285	72	48
Sum	1000	10222	120	79

We detect the modal verbs and then we calculate their raw counts as well as their proportions. We obtain the proportion of a modal verb after dividing the absolute frequency of the modal verb by the total number of the modal verbs in a sentiment class (e.g. the modal verb 'may' in 'negative' class: 3/76 = 0.04). Table 4 shows the absolute frequencies of the modal verbs for the Amazon dataset. The proportions are given inside parentheses.

Table 4: *Absolute numbers of frequency of each modal verb along with their proportions inside parentheses for the Amazon dataset*

Modals	**Positive**	**Negative**	**Sum**
May	0*(0)*	3*(0.04)*	3*(0.02)*
Might	0*(0)*	3*(0.04)*	3*(0.02)*
Will	5*(0.10)*	13*(0.17)*	18*(0.14)*
Would	18*(0.38)*	18*(0.24)*	36*(0.29)*
Can	13*(0.27)*	16*(0.21)*	29*(0.23)*
Could	8*(0.17)*	18*(0.22)*	26*(0.21)*
Should	1*(0.02)*	4*(0.05)*	5*(0.04)*
Must	3*(0.06)*	1*(0.01)*	4*(0.03)*
Sum	48	76	124

Although the modal verbs seem more skewed to the negative sentiment at first sight, the proportions of positive and negative sentences do not significantly differ across the modal verbs (X^2 (7) = 0.19, $p > 0.05$). We also want to visualize how the raw counts of the modal verbs are distributed in the positive and negative sentences for the Amazon dataset. Figure 2 displays the distributions of the positive and negative sentences across the modal verbs.

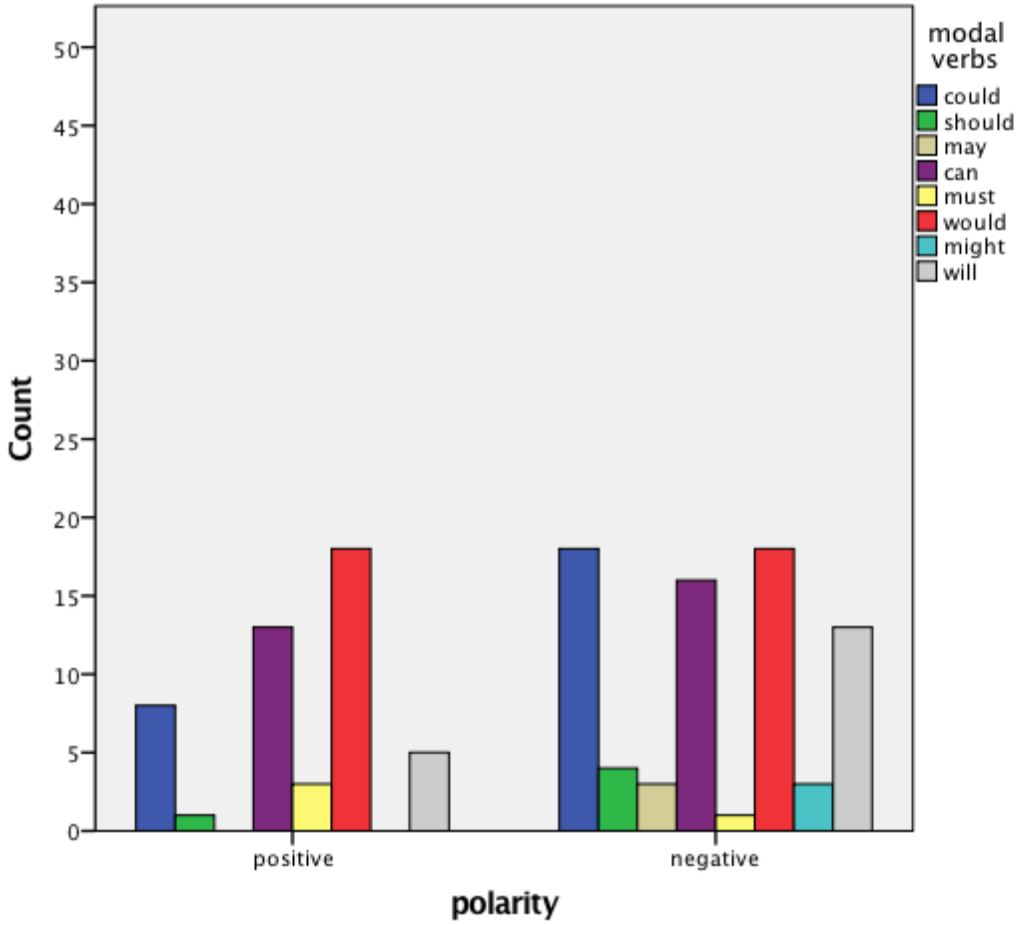

Figure 2: *The distribution of the positive and negative sentences across the modal verbs in the Amazon dataset*

Although some modals (e.g. could, will) seem to be biased towards one of the polarities (e.g. negative polarity), none of the modal verbs have any significantly higher association to any polarities: positive or negative.

4.1.3. YelpIMDB Dataset

The third dataset, the concatenation of Yelp and IMDB datasets, extracted from the reviews of restaurants and movies, comprises 2000 sentences (i.e. 1000 positive and 1000 negative sentences) in total (Kotzias et al., 2015). Each sentence has a sentiment score, either 0 (for negative) or 1 (for positive). The numbers of sentences and words as well as the number of sentences with a modal verb and an adversative conjunction in the YelpIMDB dataset are illustrated in Table 5.

Table 5: *Count of sentences, words, sentences with modality and an adversative conjunction in the YelpIMDB Dataset*

Sentiment orientation	Sentences	Words	Sentences with modalities	Sentences with conjunction
Positive	1000	12674	109	111
Negative	1000	12534	167	112
Sum	2000	25208	276	223

Upon detecting each modal verb available in the YelpIMDB dataset, we measure the absolute frequency of the modal verbs as well as the proportions of the modal verbs in the positive and negative sentences. Table 6 illustrates the absolute frequencies of the modal verbs in for the YelpIMDB dataset. The proportions are given inside parentheses. We divide the absolute numbers of the modal verbs by the total number of the modal verbs in a sentiment class so that we obtain the proportions of the modal verbs (e.g. the modal verb 'can' in 'positive' class: 28/116 = 0.24).

Table 6: *Absolute frequencies of the modal verbs along with their proportions inside parentheses in the YelpIMDB Dataset*

Modals	Positive	Negative	Sum
May	4*(0.03)*	5*(0.03)*	9*(0.03)*
Might	2*(0.02)*	2*(0.01)*	4*(0.01)*
Will	41*(0.35)*	42*(0.23)*	83*(0.28)*
Would	10*(0.09)*	47*(0.26)*	57*(0.19)*
Can	28*(0.24)*	37*(0.20)*	65*(0.22)*
Could	17*(0.15)*	28*(0.15)*	45*(0.15)*
Should	5*(0.04)*	13*(0.07)*	18*(0.06)*
Must	8*(0.07)*	5*(0.03)*	13*(0.04)*
Ought	1*(0.01)*	0*(0)*	1*(0.003)*
Need	0*(0)*	1*(0.01)*	1*(0.003)*
Shall	0*(0)*	1*(0.01)*	1*(0.003)*
Sum	116	181	297

Although it seems the greater occurrence of the modal verbs is in negative sentences, the proportions of the positive and negative sentences do not significantly differ in the use of modal verbs (X^2 (10) = 0.16, $p > 0.05$). The modal verb 'would' (X^2 (1) = 8.74, $p < 0.01$) is the only one modal verb showing higher association with a polarity label, which is the negative polarity. We visualize how the raw counts of the modal verbs are distributed in the positive and negative sentences for the YelpIMDB dataset in Figure 3.

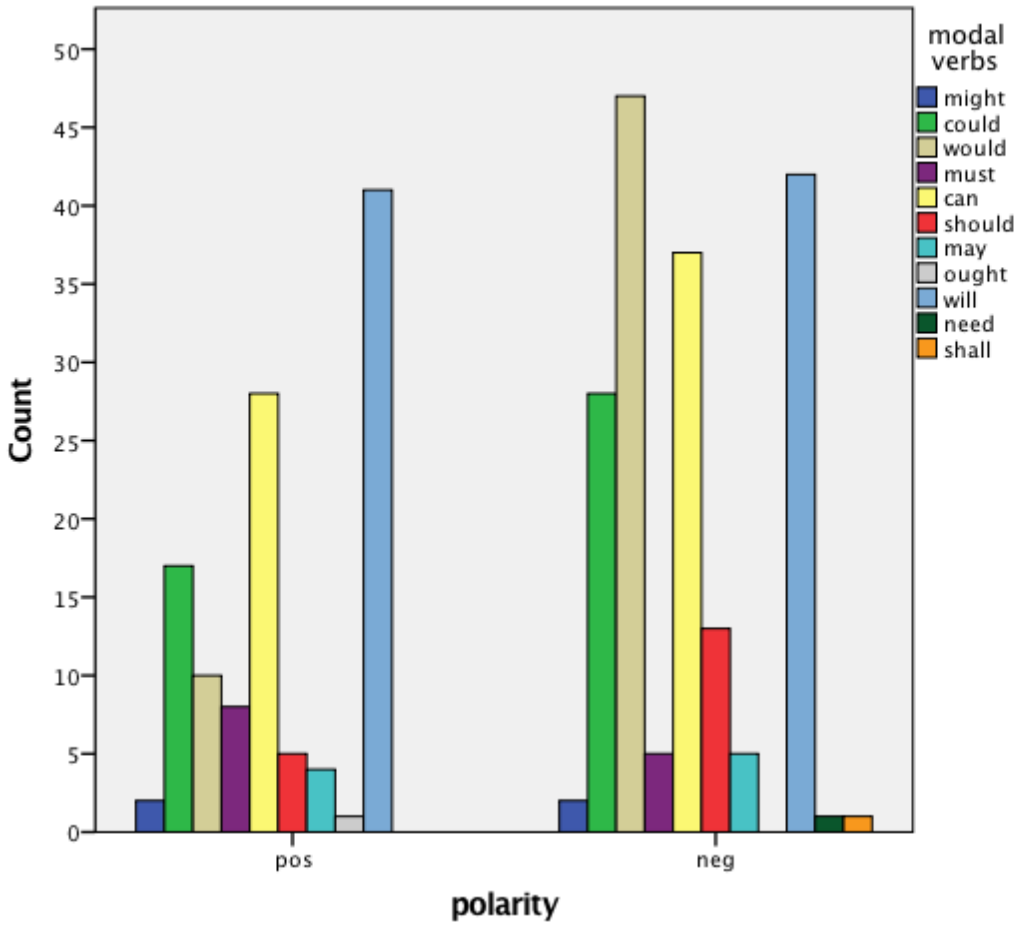

Figure 3: The distribution of the positive and negative sentences across the modal verbs for the YelpIMDB dataset

The distribution of the positive and negative sentences across the modal verbs once again shows that the modal verb 'would' has higher association with the

negative polarity. The rest of the modal verbs are not biased towards any of the two polarities (i.e. positive or negative).

4.2. Preprocessing

For each of the three datasets we normalize and prepare the datasets before classification. Since the first dataset 'Review' comprises document-based reviews labeled with 'positive' and 'negative' sentiment labels, we apply sentence splitting on this dataset by using NLTK (Loper & Bird, 2002). Since the Amazon and YelpIMDB datasets (Kotzias et al., 2015) are sentence-based reviews, there is no need to apply sentence splitting on them. We remove all whitespaces and punctuation markers for all the datasets. We lowercase all of the words. These preprocessing steps are applied to each dataset. We present the overview of the three preprocessed datasets in Table 7. This overview shows the number of documents, sentences, words, and modal verbs for each dataset.

Table 7: *Count of documents, sentences, words, and modal verbs in each dataset*

Datasets	Polarity	Documents	Sentences	Words	Modal verbs
Review	Positive	2968	22843	401136	5228
	Negative	3032	24614	407736	6107
	Sum	6000	47457	808872	11335
Amazon	Positive	x	500	4937	48
	Negative	x	500	5285	76
	Sum	x	1000	10222	124
YelpIMDB	Positive	x	1000	12674	116
	Negative	x	1000	12534	181
	Sum	x	2000	25208	297

Chapter 5

5. Models

Our baseline systems and our models are explained in this section. We present how we extract, select and represent features in our proposed models. We also report the labels we use as a feature.

5.1. Baseline Systems

We build four baseline systems. The two of the baseline systems are based on a Support Vector Machine (SVM) and Naïve Bayes (NB). The third baseline 'majority-class' takes into account the distribution of the sentiment classes (e.g. positive or negative) in the datasets. Lastly, we use a lexicon-based baseline system that takes into account of the semantic orientations of words. We run baseline systems using sentences, which are removed from punctuation markers and are lowercased.

In the following subsections, we give the details about each of the baseline systems. Specifically, we explain a Support Vector Machine classifier and a Naïve Bayes classifier briefly. We shortly explain the differences between the Support Vector Machine-based baseline system and the Support Vector Machine classifier used in our models. Moreover, we introduce the lexical resource 'SentiWordNet' (Baccianella et al., 2010) used in the lexicon-based baseline system as well as some of our models.

5.1.1. Support Vector Machine (SVM)-based Baseline

Support Vector Machine (SVM) is a vector space based machine-learning method that aims to find a decision boundary between two classes that are separated with a maximum margin. The SVM tries to find a hyperplane separating positive sentences from negative ones with a maximum margin in our case.

All of the sentences are converted into a matrix of Term Frequency Inverse Document Frequency (TF-IDF) features before we use the linear SVM-based baseline. With the linear SVM, linear separators separate the different classes (e.g. positive and negative) in the search space at foremost. According to Cortes and Vapnik (1995), "in the support-vector networks algorithm one can

control the trade-off between complexity of decision rule and frequency of error by changing the parameter C*".* Joachims (2002, p. 40) explains, "The factor *C* is a parameter that allows one to trade off training error vs. model complexity. A small value for *C* will increase the number of training errors, while a large *C* will lead to a behavior similar to that of a hard-margin SVM." We keep the default setting of C parameter for the linear SVM-based baseline system, which is unigram-based, in order to avoid both overfitting and underfitting issues.

In non-linear SVMs, there is a non-linear decision surface that the data instances are projected non-linearly. In our models, we use a non-linear radial basis function (rbf) SVM with a default penalty parameter (i.e. C factor) and an intermediate gamma value (i.e. 0.7). C and Gamma are the two powerful parameters for a non-linear support vector machine (SVM) with a radial basis function kernel. Gamma is the free parameter of the radial basis function. We choose the intermediate gamma value 0.7 in our models since we do not want to have either high or low bias. Because we aim to avoid overfitting and underfitting, we allow some amounts of training errors by using the default setting of C parameter in our models.

5.1.2. Naïve Bayes (NB)-based Baseline

Our second baseline system is built by Naïve Bayes, which is the simplest and most commonly used classifier. This classifier uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label (Medhat, Hassan, & Korashy, 2014). We apply the NB classifier for the three datasets after all of the sentences in the datasets are converted into a matrix of Term Frequency Inverse Document Frequency (TF-IDF) features. In the end, we obtain the accuracy scores of the NB-based baseline system that is unigram-based.

5.1.3. Majority-class Baseline

In a majority-class system, the most frequent label in a training set determines the labels of the test set. Then, the majority-based labels are compared to the original labels of the test set. We implement a dummy classifier from Scikit-learn to know the majority-based accuracy scores. We choose 'most_frequent' option as a strategy in the classifier for each dataset.

5.1.4. Lexicon-based Baseline

Knowing the semantic orientations of opinion-bearing words in a sentence might have an impact on the overall polarity of the sentence. For this reason, we build a lexicon-based baseline to see to what extent lexicon-based approach contributes to the accuracy scores of the sentence-level sentiment classification task.

In this lexicon-based baseline, we use a publicly available lexical resource 'SentiWordNet' (Baccianella et al., 2010) which is a continuation of WordNet (Fellbaum, 1998). SentiWordNet (SWN) v3.0 comprises 38,182 non-neutral words. Each synset of WordNet is associated with a sentiment label that is positive, negative or objective. Figure 4 shows how the SentiWordNet represents the opinion-related properties of a term sense (Baccianella et al., 2010).

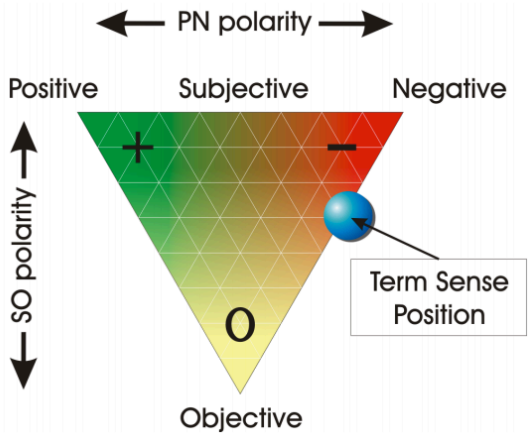

Figure 4: *The opinion-related properties of a term sense represented by the SentiWordNet*

The SentiWordNet is also considered as an effective tool for opinion mining applications for several reasons. Firstly, it owns a wide coverage since each of the three labels Objective, Positive, Negative is assigned to all WordNet synsets. Secondly, it is fine-grained and this characteristic is achieved by determining the labels by means of numerical scores (Baccianella et al., 2010).

We use SentiWordNet v3.0 and a Python library, called Sentlex in order to carry out this lexicon-based baseline system. We obtain the Sentlex implementation from a github repository[2]. The SentiWordNet focuses on opinion-bearing words (i.e. content words: verbs, adverbs, nouns, and

[2] https://github.com/bohana/sentlex

adjectives). Each word receives a sentiment value with the use of the SentiWordNet. The average of all of the assigned sentiment values determines the sentiment value of each sentence. When the average value assigned as a positive sentiment is higher than the one assigned as a negative sentiment in a sentence, the sentiment of the sentence is considered as positive. The other way round is considered as negative. When the average values assigned to positive and negative sentiments are zero, then the polarity label of the sentence becomes neutral. Every sentence gains a SentiWordNet (SWN) label in this way. We compare the labels (e.g. positive, negative) predicted by the lexicon-based baseline system to the correct polarity labels of the sentences so that we find out the accuracy score of the lexicon-based baseline system.

5.2. Feature Extraction, Selection and Representation

Selecting and representing features well play an important role in the sentiment analysis tasks. We use some generalization labels for some words in our models and then we convert the labeled features into a feature vector. We implement Term Frequency Inverse Document Frequency (TF-IDF) vectorizer since we want to indicate relative importance of features via Scikit-learn (Pedregosa et al., 2011). Some words are very frequent in a large dataset (e.g. 'the', 'is', 'a') and they do not give sufficiently meaningful information about the actual contents of datasets. To prevent such frequent words from shadowing more interesting terms with rare frequencies in a given text, the count features might be re-weighted by using the TF-IDF transform. TF-IDF Vectorizer is equivalent to CountVectorizer followed by tfidfTransformer. Tf means **term-frequency** while TF–IDF means term-frequency times **inverse document-frequency**:

$$\text{tf-idf}(t,d) = \text{tf}(t,d) \times \text{idf}(t).$$

Term frequency shows the number of times a term occurs in a document. An *inverse document frequency* factor diminishes the weight of terms occurring very frequently in the document set and increases the weight of terms occurring rarely in the given document. TF-IDF is mostly used for text features. In this sentiment classification task, we use the term frequency weights to indicate the relative importance of features for each dataset and model, including baseline systems.

In the following subsections, we report the labels we use for linguistic features and how we represent them elaborately.

5.2.1. Negation Words

The presence of negation words such as 'no' and 'not' changes polarity of words adjacent to negation words. For example, the sentence *this offer is not exciting* contains a positive opinion bearing word '*exciting*'. However, the author of this sentence expresses a negative opinion on this offer. Without the negation word, the polarity of this sentence would be positive. Considering the power of negation words on the semantic orientation of a sentence, we represent the negation words (e.g. not, no, never) with the label 'NEG'. We either replace the negation words (e.g. not) with the negation label 'NEG' or add both the negation words and the negation labels (e.g. not, NEG) in our models. For instance, the negation word 'not' in the sentence *he will **not** buy this phone* is shown as either '*NEG'* or *'not'* and *'NEG'* in our models.

5.2.2. Adversative Conjunctions

Adversative conjunctions are sometimes known as contrasting conjunctions and they express comparisons or contrasts. The most common adversative conjunctions are *but, yet, still, however, although, despite, though* and *nevertheless*. They play an important role in the semantic orientations of sentences. For example, the sentence *she is an angry **but** honest girl* comprises an adversative conjunction '*but*' that connects two words with opposite meanings and semantic orientations. When we consider the prior adjective 'angry', the sentence would carry a negative polarity. However, posterior adjective 'honest' following the contrasting conjunction 'but' is a positive content word and it changes the semantic orientation of this sentence from negative to positive. Since adversative conjunctions might have a strong impact on the polarity classification, we want to pay an extra attention to the conjunctions: "but", "however", "although", "despite", "though", "yet", "nevertheless" and "still". We represent these conjunctions with the label "ADVERSATIVE" and then we convert them into a feature vector. Similar to the label used for the negation words, the 'ADVERSATIVE' label is used either as a placeholder or as an

additional feature in our models. For instance, the conjunction '*but*' in the sentence *she is an angry but honest girl* would be represented as either 'ADVERSATIVE' or 'but' and 'ADVERSATIVE'.

5.2.3. Numbers

We label all of the digits available in each dataset with the generalization label 'NUMBER' in some of our models. Namely, all numbers are represented with the unvarying label (i.e. NUMBER). We can consider this labeling as normalization. This label is used as either a placeholder or an additional feature meaning we add both digits and the label 'NUMBER'. For instance, the number "*70"* in the sentence *he paid 70 euros for business lunch* would be labeled as either *'NUMBER'* or '*70'* and *'NUMBER'*.

5.2.4. Modal Verbs

Modal verbs carry a significant role in the sentiment analysis tasks. Although they are not opinion bearing words, they might have weights to change the semantic orientations of sentences. We detect modal verbs in each dataset by using part-of-speech tagger in NLTK (Loper & Bird, 2002). We have noticed that the part-of-speech tagger has not detected some of the negated modal verbs correctly due to several factors such as incorrectly written forms of the modal verbs (e.g. *couldnt). In these cases, we have normalized these modal verbs by separating the negation affix from the stem (e.g. could not). We have applied this normalization for each dataset in order to calculate the number of modal verbs and the number of sentences containing a modal verb in each dataset correctly. We have re-calculated the numbers of words, sentences, modal verbs and sentences with a modal verb and an adversative conjunction in each modified dataset. We have carried out statistics once again after modifying each dataset (e.g. *couldnt becomes could not). The section 4 'Datasets and Preprocessing' introduces the dataset correctly by taking into account the recent modifications. Additionally, we have re-run some of our models on the modified datasets to see whether or not the accuracy scores of the models change. Since we do not obtain any differences in terms of the accuracy scores of the models, we rely on the

automatic detection of part-of-speech tagging for measuring the accuracy scores of our models.

We use three progressively augmented generalization labels over modal verbs in terms with relation to semantics. We briefly explain three different degrees of generalization over the modal verbs in the sections from 5.2.5 to 5.2.7.

5.2.5. MODALNAMES

The first generalization over the modal verbs has the least relation to semantics. The modal verbs available in the datasets are merely uppercased. For example, the modal verb '*would*' in the sentence *this book would be excellent* is represented with the label 'WOULD'.

5.2.6. MODAL

We use a global label for each modal verb available in the datasets. Instead of labeling each modal verb differently, we label all of the modal verbs with the same label, which is 'MODAL' For instance, the modal verbs '*should*' and '*can*' in the sentences *he should eat pizza* and *she can order Döner* are shown with the label 'MODAL'. Namely, we replace all of the modal verbs with the global label 'MODAL'.

5.2.7. MD$function

In the last degree of generalization over the modal verbs, the labels have some relation to semantics to some extent. We use English Modality Lexicon built by Baker et al. (2014) for labeling some of the modal verbs. This modality lexicon is not created for a sentiment analysis task but a machine translation task. The lexicon comprises six modal verbs that we can use in the sentiment classification task, which are 'may', 'might', 'can', 'could', 'must' and 'should'. As for other modal verbs, we uppercase them as expressed in the Section 5.2.5.

Each of the six modal verbs contains a label in the modality lexicon. Specifically, three labels are used in total: the label 'MD$Belief' is used for the modal verbs 'may' and 'might', 'MD$Able' is the label used for the modal verbs 'can' and 'could'. Finally, 'MD$Require' is the label used for the modal verbs 'should' and 'must'. These generalization labels used for the modal verbs (e.g.

MD$Able for the modal verb 'can' and 'could') are used as either a placeholder or an additional feature in some of our models. For example, the modal verb '*should*' in the sentence *waiters should be kind* is labeled as either *'MD$Require'* or *'should'* and *'MD$Require'*.

From a theoretical linguistic perspective, some of the modal verbs (e.g. can and may) may belong to more than one modal category, which creates ambiguous cases. For example, the modal verb 'may' belongs to the deontic modal category in the sentence *She may eat now* expressing *a permission to eat* while it belongs to the epistemic modality in the same sentence *She may eat now* expressing *the speaker is not certain if she is eating now*. From a computational perspective, such ambiguities make detecting and labeling the modal categories (e.g. deontic, epistemic) automatically difficult. Our generalization labels used for the modal verbs do not cause any ambiguity since we do not distinguish the categories of the modal verbs in our labels (e.g. deontic). Another ambiguity cases can be stemmed from the word categories to be assigned to the words available in the datasets. For example, the word 'can' might have two part-of-speech tags: noun or modal verb. To avoid such ambiguous cases, we use the part-of-speech tagger (NLTK) so that we are able to detect and label the modal verbs correctly.

5.2.8. Use of Part-of-Speech Tags (POS Tags)

We apply three different levels of feature representations to our models. First of all, we perform a word-level representation, which we keep the words as words. For example, the sentence *Samsung TV is a disappointment* is shown as *'samsung', 'tv', 'is', 'a', 'disappointment'*. Secondly, we use a part-of-speech tag (POS tag)-level representation, which we replace words with their POS tags. For instance, the same sentence would be represented with the POS tags like this: *'NNP', 'NN', 'VBZ', 'DT', 'NN'*. The final level of a feature representation is to use words along with their POS tags. The same sentence would be represented like 'samsung_NNP', 'tv_NN', 'is_VBZ', 'a_DT', 'disappointment_NN'.

5.2.9. N-Gram Features

We apply different n-gram ranges (e.g. unigram, bigram, skipgram) for our models so that we find out which n-gram model performs best. We specify the n-gram-ranges inside the TF-IDF vectorizer that combines all the options of CountVectorizer and TfidfTransformer in a single model as stated in Sklearn[3]. We use a word-based n-gram to represent the features. Unigrams contain independent words such as '*interesting*", which we would know the sentiment of the independent opinion bearing word 'positive'. Bigrams containing pair of word are used as a feature vector. For example, the pair of word *'not interesting'*, which carries a negative meaning, would be represented together. Table 8 shows how different n-gram ranges are represented for the sentence '*this book is not interesting'*.

Table 8: *N-gram representations of the sentence 'this book is not interesting'*

N-gram feature selections	Feature representations
Unigram-based	*'this', 'book', 'is', 'not', 'interesting'*
Bigram-based	*'this book', 'book is', 'is not', 'not interesting'*
Trigram-based	*'this book is', 'book is not', 'is not interesting'*
Fourgram-based	*'this book is not', 'book is not interesting'*
Fivegram-based	*'this book is not interesting'*

Some collocations are unable to come together with the use of simple n-grams because some additional words are available between frequent word combinations. To reach more word combinations, we also use skip-grams via nltk.util.skipgrams and sklearn Tf-Idf vectorizer. We can consider n-grams as skip-grams with slop 0. For the skip-grams, we parameterize looseness by specifying the slop that is the number of additional words. We extract high order n-grams; bigram, trigram, fourgram and fivegram with the slop value '2', which means there might be two skipped words between word combinations. Table 9 illustrates how the sentence '*this book is not interesting*' is represented in 2-slop high order n-grams.

[3] http://scikit-learn.org/stable/modules/feature_extraction.html

Table 9: *Skip-gram representations of the sentence 'This book is not interesting'*

Skip-gram feature selection	**Feature representations**
2-slop bigram-based	*'this book', 'this is', 'this not', 'book is', 'book not', 'book interesting', 'is not', 'is interesting', 'not interesting'*
2-slop trigram-based	*'this book is', 'this is not', 'this not interesting', 'book is not', 'book not interesting', 'is not interesting'*
2-slop fourgram-based	*'this book is not', 'this is not interesting'*
2-slop fivegram-based	*'this book is not interesting'*

5.3. Our Proposed Models

In this section, we introduce the details of our proposed models. We create several models differing in a few aspects from one another. In this way, we want to find out the contributions of features, parameters, and resources on the accuracy scores of our models using the three datasets: 'Review', 'Amazon' and 'YelpIMDB'. Our models are built by a non-linear radial basis function 'Support Vector Machine (SVM)' classifier with the default setting of C parameter and intermediate gamma value 0.7. In each model, linguistic features such as modal verbs, adversative conjunctions and negation words are represented with the generalization-based labels. Not all of our proposed models comprise the SentiWordNet labels (e.g. positive, negative, neutral) and the label used for digits (i.e. NUMBER).

5.3.1. Model 1: SVM_MODALNAMES

In this model, we use labels for the linguistic features that are negation words, adversative conjunctions and modal verbs. As explained in the Section 5.2.5. MODALNAMES, the modal verbs are uppercased in this model. Instead of keeping these linguistic features (i.e. negation words, adversative conjunctions and modal verbs) as they are, we use their labels as a feature. Apart from the labeled features, we represent other words at three different levels: a word level, a part-of-speech tag (POS tag) level and a word along with part-of-speech-tag level (see Section 5.2.8.) for the unigram-based Model 1. To illustrate how the

sentence *but he should never say never* would be represented at different levels by the unigram-based Model 1, we added the examples from i to iii.

i. **Word-level unigram-based model**
   'ADVERSATIVE', 'he', 'SHOULD', 'NEG', 'say', 'NEG'
ii. **Part-of-speech tag (POS tag)-level unigram-based model**
   'ADVERSATIVE', 'PRP', 'SHOULD', 'NEG', 'VB', 'NEG'
iii. **Word and POS tag-level unigram-based model**
   'ADVERSATIVE', 'he_PRP', 'SHOULD', 'NEG', 'say_VB', 'NEG'

5.3.2. Model 2: SVM_MODAL

This model is similar to the first model *SVM_MODALNAMES* in terms of containing labels for the linguistic features such as negation words, adversative conjunctions and modal verbs. Similar to the Model 1, this model contains different levels of feature representation such as word level (e.g. smart), POS tag level (e.g. JJ) and both word and POS tag level (e.g. smart_JJ). Different n-gram ranges are again applied to the unigram-based Model 2. What is different from the Model 1 is the label used for the modal verbs. In this model, we use a global label 'MODAL' for all modal verbs (see Section 5.2.5.). We want to show how the unigram-based Model 2 would represent the sentence *but he should never say never* at different levels in the examples from iv to vi.

iv. **Word-level unigram-based model**
   'ADVERSATIVE', 'he', 'MODAL', 'NEG', 'say', 'NEG'
v. **Part-of-speech tag (POS tag)-level unigram-based model**
   'ADVERSATIVE', 'PRP', 'MODAL', 'NEG', 'VB', 'NEG'
vi. **Word and POS tag-level unigram-based model**
   'ADVERSATIVE', 'he_PRP', 'MODAL', 'NEG', 'say_VB', 'NEG'

5.3.3. Model 3: SVM_MD$function

This model contains labels for the linguistic elements such as negation words, adversative conjunctions like the Models 1 and 2. As for the labels used for the modal verbs, we are inspired by English Modality Lexicon built by Baker et al.

(2014). As stated in the section 5.2.7 (i.e. MD$function), we label six modal verbs, which are 'can', 'could', 'may', 'might', 'must' and 'should' as they are represented in the English Modality Lexicon and the rest of modal verbs is represented in uppercase (see Section 5.2.6.). We run the unigram-based Model 3 at the three levels (i.e. word level, POS tag level, word and POS tag level) for the three datasets. To give an idea how the unigram-based Model 3 would represent the features of the sentence *but he should never say never* at different levels, see the examples from vii to ix.

vii. Word-level unigram-based model

'ADVERSATIVE', 'he', 'MD$Require', 'NEG', 'say', 'NEG'

viii. Part-of-speech tag (POS tag)-level unigram-based model

'ADVERSATIVE', 'PRP', 'MD$Require', 'NEG', 'VB', 'NEG'

ix. Word and POS tag-level unigram-based model

'ADVERSATIVE', 'he_PRP', 'MD$Require', 'NEG', 'say_VB', 'NEG'

5.3.4. Model 4: SVM_MD$function+SWNlabels

This model comprises the same labels assigned for the negation words, adversative conjunctions and modal verbs as the Model 3. Besides these linguistic features, this model labels opinion bearing content words (i.e. noun, verb, adjective, and adverb) with the SentiWordNet (SWN) sentiment labels. If a content word follows a modal verb, one of the three SWN sentiment labels 'POSITIVE', 'NEGATIVE' and 'NEUTRAL' is assigned to the content word. These sentiment labels are used as a placeholder. Namely, the content words are represented with their corresponding SWN polarity labels as a feature. The unigram-based Model 4 represents the features at three levels (i.e. word level, POS tag level, word and POS tag level). High-order n-grams and skip-grams are also applied to this model when the features are represented only at the word level. The examples from x to xii illustrate how this model would represent the features of the sentence *but he might hate cats* in different n-gram ranges and at word level.

x. **Word-level unigram-based model**

'ADVERSATIVE', 'he', 'MD$Belief', 'NEGATIVE', 'cats'

xi. **Word-level bigram-based model**

'ADVERSATIVE, he', 'he, MD$Belief', 'MD$Belief, hate', 'hate, cats'

xii. **Word-level 2-skip bigram-based model**

'ADVERSATIVE, he', 'ADVERSATIVE, MD$Belief', 'ADVERSATIVE, hate', 'he, MD$Belief', 'he, hate', 'he, cats', 'MD$Belief, hate', 'MD$Belief, cats', 'hate, cats'

The section xii shows the feature representation of 2-skip word-level bigram-based model for the sentence *but he might hate cats*. 2-skip bigram-based model means that there might be maximum two words to be skipped for creating bigram features. For example, adversative conjunction 'but' can be accompanied with the subject pronoun 'he', the modal verb 'might' and the negative opinion bearing word 'hate' because we are allowed to skip maximally 2 words to have more bigrams. When we use skip-grams to represent the features, we obtain more features as an output.

5.3.5. Model 5: SVM_MD$function_SWNlabels[additional feature]-for-all-content-words

In this model, we again label the same linguistic features that are negation words, adversative conjunctions and modal verbs. We again label the modal verbs by taking into account the labels from the English modality lexicon (Baker et al., 2014). A new label 'NUMBER' is used for digits occurring in the sentences. The external lexicon database 'SentiWordNet' is used for all content words no matter where the content words are situated in a sentence. Different levels of feature representations (i.e. word level, POS tag level and word and POS tag level) are applied to the unigram-based Model 5. The sentiment labels are not added as a placeholder. However, they are added as an additional feature, which means that a word such as *'hates*' is added as a feature and its sentiment label 'NEGATIVE' is added as another feature for this word. In the examples from xiii to xv, we illustrate the unigram-based feature representations (i.e. word level,

POS tag level, word and POS tag level) of the sentence *should he love cats* in this model.

xiii. Word-level unigram-based model

'MD$Require', 'he', 'love', 'POSITIVE', 'cats', 'NEUTRAL'

xiv. POS tag-level unigram-based model

'MD$Require', 'PRP', 'VB', 'POSITIVE', 'NNS', 'NEUTRAL'

xv. Word and POS tag-level unigram-based model

'MD$Require', 'he_PRP', 'love_VB', 'POSITIVE', 'cats_NNS', 'NEUTRAL'

The Model 5 produces more features than the Model 4 because we add the sentiment labels for each content word independently from their positions in a sentence. Also, the sentiment labels are additional features. For example, in example xiii, both sentiment label 'POSITIVE' and content word 'love' are represented in the output as separate features. Similarly, the word 'cats' in xiv is represented with multiple features. The first feature is for its word category 'NNS' and the second feature is used for showing its SWN sentiment label 'NEUTRAL'.

5.3.6. Model 6: SVM_MD$function_SWN_all-labels[additional feature]

This model is exactly the same as the Model 5 in terms of the labels used for the linguistic features and digits as well as the SentiWordNet labels. What is different from the Model 5 is that we add all of the labels as additional features in this model. Namely, we do not use the labels as a placeholder. For instance, the adversative conjunction 'although' would be represented with two features: 'ADVERSATIVE' and 'although'. Different levels of the feature representations are also applied to the unigram-based Model 6 (i.e. word level, POS tag level and word and POS tag level). In the examples from xvi to xviii, we illustrate the unigram-based feature representations of the sentence *should he love cats* in this model.

xvi. **Word-level unigram-based model**

'should', 'MD$Require', 'he', 'love', 'POSITIVE', 'cats', 'NEUTRAL'

xvii. **POS tag-level unigram-based model**

'MD', 'MD$Require', 'PRP', 'VB', 'POSITIVE', 'NNS', 'NEUTRAL'

xviii. **Word and POS tag-level unigram-based model**

'should_MD', 'MD$Require', 'he_PRP', 'love_VB', 'POSITIVE', 'cats_NNS', 'NEUTRAL'

As seen in the examples, the Model 6 produces more features than the Model 5. We seek to find out to what extent adding more features contributes the accuracy scores of our models. In the following section, we explain our experimental settings and results.

Chapter 6

6. Experimental Settings and Results

In this section, we firstly explain all of our experimental settings elaborately. Then the results of our study will be given.

6.1. Experimental Settings

In this section, we explain how we evaluate our baseline systems and our proposed models. We also express how we train and test our models using the Amazon, YelpIMDB and Review datasets. . Recapping that the 'Review' dataset comprises 47457 sentences with 22843 positive and 24614 negative sentences. The 'Amazon' dataset contains 1000 sentences with 500 positive and 500 negative sentences. The 'YelpIMDB' dataset has 2000 sentences with 1000 positive and 1000 negative sentences. If the resulting score is a fraction, it is rounded to the nearest whole number – ½ or more is rounded up (e.g. 0.5397 becomes 0.54); less than ½ is rounded down (e.g. 0.5322 becomes 0.53).

As stated in the previous sections, we used four baseline systems, which are the majority-class-based, the lexicon-based, the linear Support Vector Machine-based and Naïve Bayes-based. We used the full datasets for the lexicon-based baseline system. We cross-validated (i.e. 10-fold) the datasets for the rest of the baseline systems. That is, each dataset was split into 10 equal sized subsets. From these split subsets, one subset was chosen as the validation data for testing the model, and the other remaining subsets were used as a training data. The cross-validation process was performed 10 times in order for each of the data subsets to be used once as the validation data. Apart from the four baseline systems, we reported six models, which were based on different n-gram ranges. Specifically, we used the six models as unigram models, four of them as high-order n-gram models and one of them as high-order skip-gram model (i.e. 2-skip). A Support Vector Machine (SVM) classifier was applied to each model. The SVM can be used with different kernels (e.g. linear and non-linear 'rbf') and penalty rates (e.g. C = 10, 100). To find out which kernel (i.e. linear vs non-linear 'rbf') and penalty rate (i.e. C = 1, 10, 100, 1000) would be a plausible choice for our models, we selected one of the unigram-based models, 'Model 1:

SVM_MODALNAMES' with the features represented at three levels (i.e. word level, part-of-speech-tag 'POS tag' level, word along with POS tag level) in order to see the contributions of the 'linear' and 'non-linear rbf' kernels and penalty rates increasing from 1 to 1000 on the accuracy scores. Since we worked on split datasets in the very beginning of our study, the Model 1 was ran on the split Review dataset, which means 75% of the dataset were used as a training set and 25% were used as a development set. Then, the Amazon dataset was used as a test set. Only for this comparison (i.e. linear vs non-linear kernel, C = 1 to C = 10, 100, 1000), which we aimed to find out better kernel and penalty rate, we used split datasets.

For all of the six unigram-based models [4] including the Model 1: SVM_MODALNAMES, each of the datasets was cross-validated and built by the non-linear radial basis function 'rbf' Support Vector Machine with the default setting of C factor and the intermediate level of gamma value '0.7'. The unigram models were represented at the three levels (e.g. word level, POS tag level, word along with POS tag level).

In order to find out which n-gram range would improve the performance of our proposed models, we selected one of our proposed models: Model 4: SVM_MD$function+SWNlabels. We investigated the Model 4 with high-order n-grams and high-order skip-grams. We ran the Model 4 built by the linear Support Vector Machine with the default setting of C factor, due to time efficiency, for the 10-fold cross-validated Review dataset.

Among our proposed models, we identified the best performing models with the aid of McNemar's test. Then one of the best models was selected in order to investigate the contribution of the labels used for linguistic and lexical features on the performance of the selected model again with the use of McNemar's test. Additionally, we wanted to see the contribution of running the best model on training sets whose domains and sizes showed similarity or differences compared to test sets on the accuracy score of the test sets. For this purpose, we cross-validated datasets before we used them as training sets and

[4] ***The unigram models:*** Model 1: SVM_MODALNAMES; Model 2: SVM_MODAL; Model 3: SVM_MD$function; Model 4: SVM_MD$function+SWNlabels; Model 5: SVM MD$function_SWNlabels[additional feature]-for-all-content-words; Model 6: SVM MD$function_SWN_all-labels[additional-feature]

then we ran the best model on another dataset as a test set different from the training sets. All the features were represented at the word level.

6.2. Results

In this section, we present the results from the baseline systems and our proposed models for sentiment analysis involving binary classification of positive and negative sentences for the three datasets: Review, Amazon and YelpIMDB.

6.2.1. Baseline Systems

Table 10 demonstrates the accuracy scores of the four baseline systems, which were the majority-class, the lexicon-based, the linear Support Vector Machine-based and Naïve Bayes-based, applied to the Review, YelpIMDB and Amazon datasets.

Table 10: *The accuracy scores of three baseline systems (i.e. the linear SVM-based, Naïve Bayes-based and majority-class baselines) applied to the 10-fold cross-validated datasets 'Amazon', 'YelpIMDB' and 'Review' and the accuracy scores of the lexicon-based baseline system applied to the three datasets without any cross validation*

Baselines	**Datasets**		
	Amazon	**YelpIMDB**	**Review**
Support Vector Machine-based Baseline	0.52	0.50	0.54
Naïve Bayes-based Baseline	0.52	0.50	0.54
Majority-class Baseline	0.50	0.50	0.52
Lexicon-based Baseline	0.70	0.69	0.54

We investigated the number of correct predictions in each baseline system for each dataset. McNemar statistical test results have showed that the linear SVM-based and NB-based baselines, which both give exactly the same number of correct and incorrect predictions, performed better than the majority-class ($p < 0.01$) and the lexicon-based baseline ($p < 0.01$) systems for the Review dataset.

As for the Amazon dataset, the lexicon-based baseline system better performed compared to both the linear SVM-based and Naïve Bayes-based baseline systems (both systems output same scores, McNemar's test, $p < 0.01$) and the majority-class baseline system (McNemar's test, $p < 0.01$). As for the YelpIMDB dataset, the lexicon-based baseline system outperformed both the linear SVM-based and the NB-based baseline systems (both systems output same scores, McNemar's test, $p < 0.01$) and the majority-class baseline system (McNemar's test, $p < 0.01$).

6.2.2. Linear and Non-Linear Kernels with Differential Penalty Rates

Table 11 demonstrates the accuracy scores of the unigram-based Model 1: SVM_MODALNAMES built by the Support Vector Machine using a linear kernel with different C factors (C = 1, 10, 100 and 1000) for the Review dataset used as a development set and the Amazon test set. The features were represented at the three levels (e.g. word level).

Table 11: *The accuracy scores of the unigram-based Model 1: SVM_MODALNAMES built by the linear Support Vector Machine with the four different C factors (i.e. from 1 to 1000) for the Review development set and the Amazon test set with the features represented at the three different levels (e.g. word level)*

C factors and Levels of Feature Representation	**Split dataset** Review (75% training set, 25% development set)	**Test set** Amazon
C=1		
Word-level	**0.70**	0.78
POS tag-level	0.57	0.65
Word and POS tag-level	0.69	0.78
C=10		
Word-level	0.68	0.75
POS tag-level	0.57	0.65
Word and POS tag-level	0.67	0.75
C=100		
Word-level	0.66	0.72
POS tag-level	0.57	0.63

Word and POS tag-level	0.65	0.69
C=1000		
Word-level	0.65	0.71
POS tag-level	0.48	0.50
Word and POS tag-level	0.64	0.67

The Model 1 built by the linear Support Vector Machine with the default setting of C factor employed on the Review development set with the features represented at the word level performed better than when the C factor was set to 10 (McNemar's test, $p < 0.01$), to 100 (McNemar's test, $p < 0.01$) or to 1000 (McNemar's test, $p < 0.01$). Table 12 demonstrates the accuracy scores of the Model 1: SVM_MODALNAMES built with the non-linear 'rbf' Support Vector Machine classifier with the four different C factors (i.e. from 1 to 1000) for the Review development set and the Amazon test set with the features represented at the three different levels (i.e. word level, POS tag level and word and POS tag level).

Table 12: *The accuracy scores of the Model 1: SVM_MODALNAMES built by the non-linear 'rbf' Support Vector Machine with the four different C factors (i.e. from 1 to 1000) for the Review development set and the Amazon test set represented at the three different levels (e.g. word level)*

C factors and Levels of Feature Representation	**Split dataset** Review[5] (75% training set, 25% development set)	**Test set** Amazon
C=1		
Word-level	0.71	0.82
POS tag-level	0.58	0.66
Word and POS tag-level	0.70	0.81
C=10		
Word-level	0.72	0.80
POS tag-level	0.57	0.64

[5] This column shows the accuracy scores of the Review dataset used as a development set.

Word and POS tag-level	0.71	0.80
C=100		
Word-level	0.72	0.80
POS tag-level	0.56	0.62
Word and POS tag-level	0.71	0.80
C=1000		
Word-level	0.72	0.80
POS tag-level	0.55	0.63
Word and POS tag-level	0.71	0.80

The Model 1 built by the non-linear 'rbf' Support Vector Machine with the default setting of C factor employed on the Review development set with the features represented at the word level did not perform better than when the C factor was 10 (McNemar's test, $p < 0.01$) and than when the C factors were 100 or 1000 (McNemar's test, $p < 0.01$). However, the Model 1: SVM_MODALNAMES built by the non-linear SVM with the default setting of C factor applied to the Review development set performed better than the Model 1: SVM_MODALNAMES built by the linear SVM with the default setting of C factor SVM (McNemar's test, $p < 0.01$) when the features were represented at the word level. Taking into account the statistical results for the Model 1 employed on the split Review dataset, we ran our unigram-based models using the non-linear Support Vector Machine with the default setting of C factor.

6.2.3. Unigram Models

Table 13 shows the accuracy scores of the six unigram-based models built by the non-linear SVM with the default setting of C factor and the intermediate level gamma value '0.7' employed on the cross-validated Review dataset and the test sets 'Amazon' and 'YelpIMDB'. The Review dataset was cross-validated (10-fold). We used the Review dataset as a training set and then we ran the unigram models (Model 1: SVM_MODALNAMES; Model 2: SVM_MODAL; Model 3: SVM_MD$function; Model 4: SVM_MD$function+SWNlabels; Model 5: SVM MD$function_SWNlabels[additional feature]-for-all-content-words; Model 6: SVM MD$function_SWN_all-labels[additional-features]) using the Amazon and

YelpIMDB datasets as test sets. The features were again represented at the three levels (i.e. word level, POS tag level, word and POS tag level).

Table 13: *The accuracy scores of the unigram models[6] built by the non-linear 'rbf' SVM with the default setting of C factor and intermediate level gamma value '0.7' for the 10-fold cross-validated 'Review', 'Amazon' and 'YelpIMDB' datasets whose features were represented at the three levels (e.g. word level)*

Baseline Systems & Models & Levels of Feature Representation	**Datasets** *10-fold-Cross-validated*		
	Review	**Amazon**	**YelpIMDB**
Model 1			
Word-level	0.72	0.82	0.80
POS tag-level	0.58	0.67	0.61
Word and POS tag-level	0.71	0.82	0.79
Model 2			
Word-level	0.72	0.83	0.80
POS tag-level	0.58	0.67	0.60
Word and POS tag-level	0.71	0.82	0.79
Model 3			
Word-level	0.72	0.83	0.80
POS tag-level	0.58	0.67	0.61
Word and POS tag-level	0.71	0.82	0.79
Model 4			
Word-level	0.72	0.82	0.80
POS tag-level	0.58	0.67	0.61
Word and POS tag-level	0.71	0.82	0.79
Model 5			
Word-level	0.72	0.83	0.80
POS tag-level	0.60	0.76	0.68
Word and POS tag-level	0.71	0.82	0.78

[6] Model 1: SVM_MODALNAMES; Model 2: SVM_MODAL; Model 3: SVM_MD$function; Model 4: SVM_MD$function+SWNlabels; Model 5: SVM MD$function_SWNlabels[additional feature]-for-all-content-words; Model 6: SVM MD$function_SWN_all-labels[additional-feature]

Model 6	0.72	0.82	0.80
Word-level	0.60	0.76	0.69
POS tag-level	0.71	0.81	0.78
Word and POS tag-level			

For the 10-fold cross-validated 'Review' dataset, the Model 6 performed better than the Model 5 (McNemar's test, $p < 0.01$) when the features were represented at the word level. The Model 6 and Model 2, which the features were represented at the word level, did not differ from each other significantly (McNmear's test, $p > 0.05$) for the Review dataset.

For the 10-fold cross-validated Amazon and YelpIMDB datasets, the statistical test results showed that the Models 5 and 6 did not significantly differ from each other (McNemar's test, $p > 0.05$). In fact, the Model 6 did not show any significant difference with any of the models for the cross-validated Amazon and YelpIMDB datasets.

Considering these statistical test results, we selected the Model 6 to be compared with the baseline systems since it was one of the high-performing models for the Review dataset. For the 10-fold cross-validated 'Review' dataset, the Model 6 performed best in comparison with all the baseline systems (McNemar's test, *all ps* < 0.01). Similar to the Review dataset, for the 10-fold cross-validated Amazon and YelpIMDB datasets, the Model 6 outperformed all of the baseline systems (McNemar's test, *all ps* < 0.01).

6.2.4. High-Order N-Gram Models and Skip-Gram Models

The high-order n-grams (i.e. bigram to fivegram) and the high-order skip-grams (i.e. 2-skip bigram to 2-skip fivegram) implemented to the Model 4: SVM_MD$function+SWNlabels were applied to the 10-fold cross-validated Review dataset with the features at the word level in Table 14. We ran Model 4 built by the linear SVM with the default setting of C factor. Since we used more features than the unigram-based models especially in the Model 4 with skip-grams, using a linear SVM should be much faster compared to a non-linear SVM.

Table 14: *The accuracy scores of the Model 4 based on the high-order n-grams (i.e. bigram to fivegram) and the high-order skip-grams (i.e. 2-skip bigram to 2-skip fivegram) and built by the linear Support Vector Machine with the default setting of C factor for the 10-fold cross-validated Review dataset whose features were represented at word level*

Model 4 with the high-order n-grams and the high-order skip-grams	**Dataset** *10-fold cross-validated* Review
Model 4	
Bigram	**0.71**
Trigram	0.71
Fourgram	0.71
Fivegram	0.70
Model 4	
2-skip Bigram	0.67
2-skip Trigram	0.61
2-skip Fourgram	0.56
2-skip Fivegram	0.53

For the 10-fold cross-validated Review dataset, the bigram-based Model 4 performed better than the Model 4 with trigrams (McNemar's test, $p < 0.01$), the Model 4 with fourgrams (McNemar's test, $p < 0.01$) and the Model 4 with fivegrams (McNemar's test, $p < 0.01$). Similarly, the Model 4 with the 2-skip bigrams outperformed the Model 4 with the 2-skip trigrams (McNemar's test, $p < 0.01$), the Model 4 with the 2-skip fourgrams (McNemar's test, $p < 0.01$) and the Model 4 with 2-skip fivegrams (McNemar's test, $p < 0.01$). Furthermore, the bigram-based Model 4 performed better than the 2-skip bigram-based Model 4 (McNemar's test, $p < 0.01$). The bigram-based Model 4 built by the linear SVM performed worse than the unigram-based Model 4 built by the non-linear 'rbf' SVM classifier for the 10-fold cross-validated Review dataset (McNemar's test, $p < 0.01$) when the features were represented at the word level.

6.2.5. The Best Performing Model

The Model 6: SVM MD$function_SWN_all-labels[additional-feature] built by a non-linear 'rbf' SVM was selected as one of the best models although the significant difference between the Model 6 and the rest of the models was found only in the Review dataset. We investigated whether the Model 6 used with the labels (i.e. the labels used for the modal verbs, negation words, adversative conjunctions and SentiWordNet sentiments) and the Models 6 without the labels significantly would differ from each other.

In this section, we also wanted to see the contribution of running the best model on training sets whose domains and sizes showed similarity or differences compared to test sets on the accuracy score of the test sets. For this purpose, we cross-validated datasets before we used them as training sets and then we ran the best model on another dataset as a test set different from our training set. All the features were represented at the word level.

6.2.5.1. The 10-fold Cross-Validated Amazon Dataset and YelpIMDB Test Set

We cross-validated the Amazon dataset (i.e. 10-fold) for the Model 6. Table 15 shows the number of correct and incorrect sentences predicted by the Model 6 with and without the labels employed on the 10-fold cross-validated Amazon dataset. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also shown in the Table 15.

Table 15: *The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels and the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels for the 10-fold cross-validated Amazon dataset with the features represented at the word level*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	825	175	
Without the modality labels	827	173	$p > 0.05$
Without the negation label	826	174	$p > 0.05$
Without the SentiWordNet labels	830	170	$p > 0.05$
Without the label of adversative conjunctions	821	179	$p > 0.05$

For the 10-fold cross-validated Amazon dataset, the Model 6 used with all the labels (i.e. best model) and the Model 6 without the labels shown in the Table 15 did not significantly differ from each other.

After we cross-validated the Amazon dataset, it was used as a training set. Then we ran the Model 6 (i.e. best model) on the YelpIMDB dataset as a test set with the features represented at the word level. The best model returned the accuracy score of 0.72 for the YelpIMDB test set. Table 16 shows the number of correct and incorrect sentences predicted by the Model 6 with and without the labels employed on the YelpIMDB dataset as a test set. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also given in the Table 16.

Table 16: *The statistical test results for the comparison of the Model 6 with all the labels to the Model 6 without the labels as well as the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels applied to the YelpIMDB test set with the features represented at the word level.*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	1439	561	
Without the modality labels	1440	560	*p > 0.05*
Without the negation labels	1432	568	***p < 0.05***
Without the SentiWordNet labels	1399	602	***p < 0.01***
Without the label of adversative conjunctions	1440	560	*p > 0.05*

The statistical test results showed that the Model 6 used with the negation and SentiWordNet labels improved the performance of the Model 6 applied to the YelpIMDB test set. However, the labels used for the modal verbs and adversative conjunctions seem not to contribute to the number of correct predictions by the Model 6 for the YelpIMDB test set.

6.2.5.2. The 10-fold Cross-Validated YelpIMDB Dataset and Amazon Test Set

We cross-validated the YelpIMDB dataset (i.e. 10-fold) for the Model 6. Table 17 demonstrates the number of correct and incorrect sentences predicted by the Model 6 with and without the labels, which was run on the 10-fold cross-validated YelpIMDB dataset. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels are also given in Table 17.

Table 17: *The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels and the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels for the 10-fold cross-validated YelpIMDB dataset with the features represented at the word level*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	1608	392	
Without the modality labels	1605	395	*p > 0.05*
Without the negation label	1599	401	***p < 0.01***
Without the SentiWordNet labels	1615	385	***p < 0.05***
Without the label of adversative conjunctions	1609	391	*p > 0.05*

The statistical test results show that the negation labels improved the performance of the Model 6 applied to the cross-validated YelpIMDB dataset. The Model 6 without the SentiWordNet labels performed better than the Model 6 containing all of the labels. Using the labels for the modal verbs and adversative conjunctions did not contribute to the number of correct predictions by the Model 6 for the cross-validated YelpIMDB dataset.

After we cross-validated the YelpIMDB dataset, it was used as a training set. Then we ran the Model 6 (i.e. best model) on the Amazon dataset as a test set with the features represented at the word level. The best model returned the accuracy score of 0.82 for the Amazon test set. Table 18 exhibits the number of correct and incorrect sentences predicted by the Amazon dataset used as a test set for the Model 6 used with all the labels and the Model 6 without the labels. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also given in the Table 18.

Table 18: *The statistical test results for the comparison of the Model 6 with all the labels to the Model 6 without the labels as well as the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels employed on the Amazon test set with the features represented at the word level.*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	820	180	
Without the modality labels	820	180	*p > 0.05*
Without the negation labels	821	179	*p > 0.05*
Without the SentiWordNet labels	788	212	***p < 0.01***
Without the label of adversative conjunctions	819	181	*p > 0.05*

The statistical test results displays that the SentiWordNet labels increased the number of correct predictions in the Model 6 applied to the Amazon test set. However, the labels used for the modal verbs, negation words and adversative conjunctions had no contribution on the correct predictions in the Model 6 employed on the Amazon test set.

6.2.5.3. The 10-fold Cross-Validated Review Dataset, The Amazon and YelpIMDB Test Sets

The Review dataset was cross-validated (i.e. 10-fold) for the Model 6. Table 19 displays the number of correct and incorrect sentences predicted by the Model 6 with and without the labels applied to the 10-fold cross-validated Review dataset. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also given in the Table 19.

Table 19: *The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels and the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels for the 10-fold cross-validated Review dataset with the features represented at the word level*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	34121	13336	
Without the modality labels	34130	13327	***p < 0.01***
Without the negation label	34060	13397	***p < 0.01***
Without the SentiWordNet labels	34131	13326	***p < 0.01***
Without the label of adversative conjunctions	34108	13349	***p < 0.01***

As displayed in the Table 19, the labels used for the negation words and the adversative conjunctions contributed on the number of correct predictions in the Model 6 for the cross-validated Review dataset. However, the modality and SentiWordNet labels decreased the number of correct predictions in the Model 6 employed on the cross-validated Review dataset.

After we cross-validated the Review dataset, it was used as a training set. Then we ran the Model 6 (i.e. best model) on the Amazon dataset as a test set with the features represented at the word level. The best model returned the accuracy score of 0.84 for the Amazon test set. Table 20 illustrates the number of correct and incorrect sentences predicted by the Model 6 with and without the labels applied to the Amazon dataset used as a test set. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also given in the Table 20.

Table 20: *The statistical test results for the comparison of the Model 6 with all the labels to the Model 6 without the labels as well as the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels applied to the Amazon test set with the features represented at the word level.*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	840	160	
Without the modality labels	840	160	*p > 0.05*
Without the negation labels	841	159	*p > 0.05*
Without the SentiWordNet labels	844	156	*p > 0.05*
Without the label of adversative conjunctions	842	158	*p > 0.05*

The number of correct predictions in the Model 6 did not decrease when the labels were excluded from the Model 6 employed on the Amazon test set.

We also ran the Model 6 (i.e. best model) on the YelpIMDB dataset as a test set with the features represented at the word level after the Review dataset was used as a training set. The best model returned the accuracy score of 0.80 for the YelpIMDB test set. Table 21 illustrates the number of correct and incorrect sentences predicted by the Model 6 with and without the labels applied to the YelpIMDB dataset used as a test set. The statistical test results for the comparison of the Model 6 with the labels to the Model 6 without the labels were also given in the Table 21.

Table 21: *The statistical test results for the comparison of the Model 6 with all the labels to the Model 6 without the labels as well as the number of correct and incorrect sentences predicted by the Model 6: SVM MD$function_SWN_all-labels[additional-feature] with and without the labels which was run on the YelpIMDB test set with the features represented at the word level.*

Model 6 with and without the labels	**The Number of Predictions**		**McNemar's test statistics**
	Correct	**Incorrect**	
Model 6			
With all the labels	1602	398	
Without the modality labels	1602	398	*p > 0.05*
Without the negation labels	1602	398	*p > 0.05*
Without the SentiWordNet labels	1594	406	***p = 0.01***
Without the label of adversative conjunctions	1600	400	*p > 0.05*

The use of SentiWordNet labels increased the number of correct predictions in the Model 6 applied to the YelpIMDB dataset. However, the labels used for the modal verbs, negation words and adversative conjunctions did not increase the number of correct predictions in the Model 6 for the YelpIMDB test set.

The Model 6, which was run on the YelpIMDB test set used after the Model 6 was trained on the Review dataset, performed better than the same model, which was run on the YelpIMDB test set used after the Model 6 was trained on the Amazon dataset (McNemar's test, $p < 0.01$). Similarly, the Model 6, which was run on the Amazon test set used after the Model 6 was trained on the Review dataset, performed better than the Model 6 applied to the Amazon test set used after the Model 6 was trained on the YelpIMDB dataset (McNemar's test, $p < 0.01$).

6.2.6. The Summary of Results

The four baseline systems contributed on the accuracy scores for each dataset in a different extent. The features represented at the word level contributed to the accuracy scores to a great extent. The unigram-based Model 6 performed better compared to the models with unigrams, high-order n-grams, high-order skip-grams and the four baseline systems for each dataset. Table 22 displays the accuracy scores, precision, recall and f1-scores of the unigram-based Model 6 built by the non-linear 'rbf' Support Vector Machine with the default setting of C factor and the intermediate level gamma value '0.7' for each of the 10-fold cross-validated datasets with the features represented at the word level.

Table 22: *The precisions, recalls, f1-scores and accuracy scores of the unigram-based Model 6 built by the non-linear 'rbf' Support Vector Machine with the default setting of C factor and the intermediate level gamma value '0.7' for the 10-fold cross-validated datasets: Review, YelpIMDB and Amazon with the features represented at the word level*

Datasets	Precision	Recall	F1-Score	Accuracy
Review	0.72	0.72	0.72	0.72
YelpIMDB	0.80	0.80	0.80	0.80
Amazon	0.83	0.82	0.82	0.82

The labels used for the linguistic and lexical features had a differential impact on the accuracy score of the Model 6 (i.e. best model) depending on the data sizes and domains as well as how we trained and tested the model.

In the following section, we discuss these results and the previously carried out relevant studies.

Chapter 7

7. Discussion

In this thesis, we investigated sentence-level binary sentiment classification using three datasets including product reviews (i.e. the Amazon and the Review datasets) and movie/restaurant reviews (i.e. the YelpIMDB dataset). We explored the following research questions:

(1) Do modal verbs improve the sentiment accuracy of our models?

(2) Do negation words, adversative conjunctions and SentiWordNet sentiments contribute to the performance of our models?

(3) Does running our models using different training and test sets in different domains and size affect the accuracy scores for the test sets? Does the data size affect the accuracy scores of our different baseline systems in a different way?

(4) Do different levels of feature representation and different n-gram models have different impacts on the sentiment accuracy?

Our findings show that the Models 5 and 6, having been trained with the Review dataset, performed better than the rest of our models when tested with both the YelpIMDB and the Amazon datasets. These two models are likely to have reached high accuracies due to the SentiWordNet sentiments (e.g. positive, negative) used for the opinion bearing words as additional features. In other words, it seems that the SentiWordNet sentiments had a positive impact on the good performance of these two high-performing models.

Regarding our first question, we have found that the modal verbs did not increase the number of correct predictions in the Model 6 (i.e. one of the best models) when tested with the Amazon and YelpIMDB datasets. Indeed, we actually found that the modal verbs decreased the number of correct predictions in the Model 6 when tested with the Review dataset. That is, we found no positive contribution of modal verbs in our best performing models' predictions accuracy. Our findings are at odds with Liu et al. (2014) and Jain et al. (2016) who found that modal verb and opinion bearing verb combinations (e.g. 'would recommend') improved their models' sentiment prediction accuracy. However, it

should be noted that these authors trained their models and tested them with sentences that contain at least one modal verb. In other words, their data contains no sentence without modal verbs. It is therefore uncertain to what extent Liu, et al (2014) and Jain, et al.'s (2016) findings can be generalized to naturalistic language settings, as it is highly unlikely that all sentences in a natural dataset would include a modal verb. Unlike Liu, et al (2014) and Jain, et al (2016), we found modal verbs do not increase accuracy in sentiment predictions when the analyses are applied on naturalistic datasets, which contain several kinds of sentences including those with and without modal verbs.

Another downside regarding Liu, et al (2014) and Jain, et al.'s (2016) methodology is that they grouped modal verbs into subjunctive, deontic and dynamic categories. However, the extent to which this categorization is warranted for each given modal verb is unclear from their study. For example, Palmer (1986) explains that modal verbs might hold more than one interpretation. It is to say that a modal verb affiliated with deontic connotations in a context might refer to dynamic modal interpretations in some other contexts. Liu, et al (2014) and Jain, et al. (2016) have labeled deontic modals when a modal verb is used with a non-first-person subject (e.g. 'you should') but dynamic modals when a modal verb is used with a first-person subject (e.g. 'I would recommend'). It is unclear to us on which grounds this categorization is linguistically plausible as the interpretation of modal verbs are in fact mostly independent of the person subject they are used in combination with. For instance, according to the authors the sentence 'I can do it' can be classified as a dynamic modal but such a sentence can also be uttered as to mean 'being permitted'. It is well known that a modal form may expose different interpretations which often overlap in meaning. Such factors seem to have worked against the models built by Liu et al. (2014) and Jain et al. (2016). Our findings therefore do not support these authors' conclusions that modal verbs contribute to sentiment analysis.

Regarding our second question, in which we sought to find out the contribution of the negation words, adversative conjunctions and SentiWordNet sentiments, we found that the negation words had a positive impact on the performance of our high-performing model, the Model 6 especially for the cross-

validated datasets (i.e. the YelpIMDB and the Review datasets) except for the Amazon dataset. It seems to stem from the fact that the Amazon dataset was smaller in size compared to the YelpIMDB and the Review datasets. This is perhaps why we did not see any contribution of the negation words to the cross-validated Amazon dataset. Our findings support Pang et al. (2002) who showed negation tagging had a positive impact on the sentiment accuracy to some extent. In line with the authors, we found that negation tagging increased the performance of the Model 6 when it was cross-validated with YelpIMDB and Review datasets. However, we did not find a significant increase in model accuracy for each of the test sets when the Model 6 was trained by a different training set. Despite this discrepancy, we believe that including negation words in our analyses has improved the model performance when large-scale cross-validated datasets are used.

We discovered that the adversative conjunctions increased the number of correct predictions in the Model 6 only when the model was cross-validated with the Review dataset. Our findings point to the importance of using large amount of data (especially discourse data that present logical links between sentences) in order to measure sizable contributions of adversative conjunctions in sentiment analysis. This is sensible because adversative conjunctions (e.g. 'however') are often found in multiple-sentence discourse data. Our findings about the adversative conjunctions were in line with previous research findings (Meena & Prabhakar, 2007) indicating that analyzing conjunctions for sentiment classification undoubtedly improves polarity predictions. Findings of Meena and Prabhakar (2007) show that not only the adversative conjunctions but also the other conjunctions contributed to the sentiment accuracy. Similarly, Liu et al. (2014) found that the adversative conjunctions had a positive impact on the sentiment accuracy. In our study, the contribution of the adversative conjunctions was found for only the large-scale dataset 'Review', of which 17% contained sentences with an adversative conjunction. For sentences containing an adversative conjunction, Liu et al. (2014) reversed the polarities of prior clauses to find out the semantic orientations of sentences, which might have caused such a positive contribution of the adversative conjunctions for their small-scale dataset. It is worthwhile remembering that Liu et al. (2014) included

only the sentences with a modal verb into their model. Assuming that the sentences with modal verbs might contain adversative conjunctions, we believe that the small dataset used by Liu et al. (2014) should not be equivalent to our small or middle-scale datasets (i.e. the Amazon and the YelpIMDB datasets) containing the sentences with/without modal verbs.

We found that the SentiWordNet sentiments increased the number of correct predictions in the Model 6 for the test sets whose domains were different from the training sets while they decreased the performance of the Model 6 for the cross-validated datasets (i.e. the YelpIMDB and the Review datasets). These domain differences in our training and test sets seem to have worked in favour of the Model 6 regarding the utilization of the SentiWordNet sentiments. For the cross-validated datasets, the SentiWordNet sentiments caused a negative influence on the sentiment accuracy. The external lexical databases such as SentiWordNet to find out the sentiments of words have been used in previous research in the field of sentiment classification at different levels such as sentence level (Hu & Liu, 2004; Meena & Prabhakar, 2007) and sub-sentence level (Zirn et al., 2011). We agree with Meena and Prabhakar (2007) that the external lexical resources (e.g. General Inquire with WordNet) are useful in examining the polarity of words and in improving accuracy for sentence level sentiment prediction. Although Meena and Prabhakar (2007) used training and test sets in the same domain (i.e. the car reviews), the use of external lexical resource (i.e. the General inquire with WordNet) increased the performance of their proposed model. By contrast, the SentiWordNet sentiments decreased the performance of our best model when trained and tested with datasets of similar domains (i.e. the cross-validated YelpIMDB and Review datasets). Hu and Liu (2004) found that the use of lexical databases (e.g. WordNet) was helpful to extend the existing set of adjectives with polarity labels and then to find out the semantic orientations of sentences with adjectives. The model in Hu and Liu (2004) predicted the sentiments of sentences containing adjectives while our model predicted sentiments of sentences with/without adjectives. Zirn et al. (2011) found the use of various sentiment lexicons (e.g. SentiWordNet, Unigram Lexicon) effective to improve the performance of their proposed model as well.

Regarding our third question related to data domains and sizes, we found that the small-scale training sets (e.g. the Amazon dataset) decreased the accuracy scores for the test sets (e.g. the YelpIMDB dataset) while the large-scale training set (e.g. the Review dataset) increased the accuracy scores for the test sets (e.g. the Amazon and the YelpIMDB datasets). Our findings therefore unarguably demonstrate that the sentiment accuracy is sensitive to sample size. The domain and size differences seem to be important in determining contributions of linguistic features (e.g. the modal verbs, adversative conjunctions) and of external lexical resources on the sentiment accuracy scores as well. In the following, the relationship between sentiment accuracy and data size and domain are summarized for the linguistic features:

(i) Including the modal verbs did not increase the number of correct predictions no matter what the data sizes and domains were.

(ii) Using the negation words in sentiment analysis showed a substantial positive impact especially when the datasets were cross-validated.

(iii) Adversative conjunctions contributed to the model performance in the current study only when a large-scale dataset (i.e. the Review) was used.

As for the SentiWordNet sentiments, we found that they decreased the model performance when the training and test sets were in the same domain. Otherwise, they increased the sentiment accuracy of the model for the test sets whose domains were different from the training sets no matter the data sizes were.

As for the baseline systems, there seems to be a relation between the data size and the high performing baseline systems. For the largest dataset (i.e. the Review dataset), the highest accuracy score was obtained with the use of the linear Support Vector Machine-based and the Naïve Bayes-based baseline systems. For the small (i.e. the Amazon dataset) or middle-scale datasets (i.e. the YelpIMDB dataset), the lexicon-based baseline system gave the highest accuracy scores. This shows that more features (e.g. SentiWordNet sentiment labels) added to the large-scale dataset might not have been as useful as the small and/or middle-scale datasets.

In previous related research in this area (Moraes et al., 2013; Pang et al., 2002; Tsutsumi et al., 2007), they ran their models on the cross-validated datasets. It should be noted that our largest training set (i.e. the Review dataset) had polarity labels at document level. We split each document available in the Review dataset into sentences. The sentences within a document were assumed to have the same polarity label as the document. This overgeneralized labeling system could be considered initially as a limitation of the Review dataset. Despite this limitation, running our best model (i.e. the Model 6) on such a large dataset significantly contributed to performance of the model for the Amazon and YelpIMDB datasets used as test sets. Although the accuracy score of the Model 6 using the Review dataset as a training set (i.e. 0.72) was higher than the baseline accuracy scores for the Review dataset, they were not as high as the accuracy scores for the test sets Amazon (i.e. 0.84) and YelpIMDB (i.e. 0.80).

Regarding our fourth question, which we asked whether or not different levels of feature representation and different n-gram models have different impacts on the sentiment accuracy. We found that the unigram-based models, features of which were represented at word level, increased the sentiment accuracy scores for all of our datasets. It is important to note that POS tagging was a helpful aid for our unigram-based models. With the use of a POS tagger, we were easily able to detect opinion-bearing words (e.g. adjectives, verbs) and then we added those words along with the SentiWordNet sentiments (e.g. positive, negative) into the Model 6. Meena and Prabhakar (2007), Liu et al. (2014), Jain et al. (2016) and Chen et al. (2015) seem to believe that the part-of-speech taggers were a useful aid for sentiment classification tasks since they used the POS taggers in pre-processing steps. Our findings show that the unigram-based Model 6 performed better compared to the model with high-order grams and skip-grams. Pang et al. (2002) also found that unigrams outperformed bigrams.

In the beginning of this thesis, we described previous research findings in the field of sentiment classification at document level, sentence level and sub-sentence level. Our study focused on binary sentiment classification at sentence level. The Model 6 tested with the Amazon and YelpIMDB datasets after the Model 6 was trained by the Review dataset performed well. Among the previously reported sentence-level sentiment classification studies (Hu & Liu,

2004; Jain et al., 2016; Liu et al., 2014; Meena & Prabhakar, 2007), the model of Hu and Liu (2004) had an accuracy score of 0.84. However, we strongly believe that their study was not purely based on the sentence-level sentiment classification since they used contextual information when necessary. From this perspective, the study of Hu and Liu (2004) can be also considered as an example of a document-level sentiment classification task. Regarding the previously reported studies in the field of document-level sentiment classification (Moraes et al., 2013; Pang et al., 2002; Tsutsumi et al., 2007), not surprisingly, the document-level studies showed high accuracy scores, which seems to stem from the fact that the document-level studies had a great amount of informative features compared to sentence-level studies. Regarding the previous research studies on sub-sentence level sentiment classification(Chen et al., 2015; Zirn et al., 2011), the accuracy scores of the models (i.e. 0.63 and 0.69) built by Chen et al. (2015) and Zirn et al. (2011) showed lower accuracy scores compared to the previously reported sentence-level and document-level studies. These differential results might have derived from the fact that sub-sentence-level studies had less informative features than the sentence-level and document-level studies.

In conclusion, our work shows that linguistic aspects and external lexical resources have differential effects on the sentiment accuracy depending on how we run our models using training and test sets as well as the sizes and domains of the datasets to be used. Although we could not see the contribution of the modal verbs to the sentiment accuracy in our study as proposed by Jain et al. (2016) and Liu et al. (2014), our study can be considered as a starting point for employing much more complex strategies towards the modal verbs and similar linguistic aspects.

Chapter 8

8. Conclusion

This thesis investigated contributions of the linguistic aspects (e.g. modal verbs, negation words and adversative conjunctions) and external lexical resources (e.g. the SentiWordNet) on the performance of our models built by a non-linear 'rbf' SVM for the three datasets. Our conclusions include that (i) modal verbs do not positively affect sentiment-analysis accuracy, (ii) negation words contribute to the sentiment accuracy if a sufficient amount of data is cross-validated and (iii) adversative conjunctions do contribute to the accuracy scores when implemented on large-scale datasets, and that (iv) the SentiWordNet sentiments are helpful in sentence-level sentiment analysis when training and test sets are in different domains.

Running models on large-scale training sets has a positive impact on the accuracy scores of test sets. Findings from this study may provide insights for decision makers who want to know which linguistic features and external lexical resource should be used for their available datasets.

References

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.* Paper presented at the LREC.

Baker, K., Bloodgood, M., Dorr, B. J., Filardo, N. W., Levin, L., & Piatko, C. (2014). A modality lexicon and its use in automatic tagging. *arXiv preprint arXiv:1410.4868*.

Blitzer, J., Dredze, M., & Pereira, F. (2007). *Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification.* Paper presented at the ACL.

Chen, Z., Huang, Y., Tian, J., Liu, X., Fu, K., & Huang, T. (2015). Joint model for subsentence ‐ level sentiment analysis with Markov logic. *Journal of the Association for Information Science and Technology, 66*(9), 1913-1922.

Choi, Y., & Cardie, C. (2010). *Hierarchical sequential learning for extracting opinions and their attributes.* Paper presented at the Proceedings of the ACL 2010 conference short papers.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Das, S., & Chen, M. (2001). *Yahoo! for Amazon: Extracting market sentiment from stock message boards.* Paper presented at the Proceedings of the Asia Pacific finance association annual conference (APFA).

De Haan, F. (1999). Evidentiality and epistemic modality: Setting boundaries. *Southwest journal of linguistics, 18*(1), 83-101.

DiNucci, D. (1999). Fragmented future. *Print, 53*(4), 32.

Duverle, D. A., & Prendinger, H. (2009). *A novel discourse parser based on support vector machine classification.* Paper presented at the Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2.

Fellbaum, C. (1998). *WordNet*: Wiley Online Library.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015). *Semeval-2015 task 11: Sentiment analysis of figurative language in twitter.* Paper presented at the Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015).

Hu, M., & Liu, B. (2004). *Mining and summarizing customer reviews.* Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

Israel, M. (2004). The pragmatics of polarity. *The handbook of pragmatics*, 701-723.

Jain, S., Colaco, L. M., & Rodrigues, O. (2016). *Sentiment Analysis with Modality Processing.* Paper presented at the Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015.

Jespersen, O. (1924). The Philosophy of Grammar.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*: Kluwer Academic Publishers.

Kennedy, A., & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence, 22*(2), 110-125.

Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015). *From group to individual labels using deep features.* Paper presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Liu, Y., Yu, X., Liu, B., & Chen, Z. (2014). *Sentence-level sentiment analysis in the presence of modalities.* Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics.

Loper, E., & Bird, S. (2002). *NLTK: The natural language toolkit.* Paper presented at the Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1.

Lyons, J. (1977). Semantics.

Martineau, J., & Finin, T. (2009). Delta TFIDF: An Improved Feature Space for Sentiment Analysis. *Icwsm, 9*, 106.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093-1113.

Meena, A., & Prabhakar, T. (2007). *Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis.* Paper presented at the European Conference on Information Retrieval.

Mithun, M. (2001). *The languages of native North America*: Cambridge University Press.

Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications, 40*(2), 621-633.

Morante, R., & Sporleder, C. (2012). Modality and negation: An introduction to the special issue. *Computational linguistics, 38*(2), 223-260.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). SemEval-2016 task 4: Sentiment analysis in Twitter. *Proceedings of SemEval*, 1-18.

Nissim, M., & Patti, V. (2016). Semantic aspects in sentiment analysis. *Sentiment Analysis in Social Networks, Federico Pozzi, Elisabetta Fersini, Bing Liu, and Enza Messina (Eds.). Elsevier*.

Palmer, F. R. (1986). 2001. *Mood and modality*.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up?: sentiment classification using machine learning techniques.* Paper presented at the Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10.

Paul, P. (1978). Longman dictionary of contemporary English. *England: Longman Group Limited*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(Oct), 2825-2830.

Popescu, A.-M., & Etzioni, O. (2005). *Extracting product features and opinions from reviews.* Paper presented at the Proceedings of the conference on human language technology and empirical methods in natural language processing.

Roberts, J. R. (1990). Modality in Amele and other Papuan languages. *Journal of linguistics, 26*(02), 363-401.

Su, Q., Huang, C.-R., & Chen, H. K.-y. (2010). *Evidentiality for text trustworthiness detection.* Paper presented at the Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground.

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications, 34*(4), 2622-2629.

Tsutsumi, K., Shimada, K., & Endo, T. (2007). *Movie Review Classification Based on a Multiple Classifier.* Paper presented at the PACLIC.

van der Wouden, T. (1997). Negative Contexts: Collocation, Polarity and Multiple Negation.

Von Wright, G. H. (1951). An essay in modal logic.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). *Recognizing contextual polarity in phrase-level sentiment analysis.* Paper presented at the Proceedings of the conference on human language technology and empirical methods in natural language processing.

Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics, 35*(3), 399-433.

Zirn, C., Niepert, M., Stuckenschmidt, H., & Strube, M. (2011). *Fine-Grained Sentiment Analysis with Structural Features.* Paper presented at the IJCNLP.