ENHANCING SENTIMENT LEXICA WITH NEGATION AND MODALITY FOR SENTIMENT ANALYSIS OF TWEETS

CANBERK BERKİN ÖZDEMİR

A thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Computer Science

Concordia University

Montréal, Québec, Canada

August 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Canberk Berkin Özdemir**

Entitled: **Enhancing Sentiment Lexica with Negation and Modality for Sentiment Analysis of Tweets**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

______________________________ Chair

Dr. Nikolaos Tsantalis

______________________________ Examiner

Dr. Leila Kosseim

______________________________ Examiner

Dr. René Witte

______________________________ Supervisor

Dr. Sabine Bergler

Approved ______________________________

Chair of Department or Graduate Program Director

____________ 2015 ______________________________

Dr. Amir Asif, Dean

Faculty of Engineering and Computer Science

Abstract

Enhancing Sentiment Lexica with Negation and Modality for Sentiment Analysis of Tweets

Canberk Berkin Özdemir

Sentiment analysis became one of the core tasks in the field of Natural Language Processing especially with the rise of social media. Public opinion is important for many domains such as commerce, politics, sociology, psychology, or finance. As an important player in social media, Twitter is the most frequently used microblogging platform for public opinion on any topic.

In recent years, sentiment analysis in Twitter turned into a recognized shared task challenge. In this thesis, we propose to enhance sentiment lexica with the linguistic notions negation and modality for this challenge. We test the interoperability between various sentiment lexica with each other and with negation and modality and add some Twitter-specific ad-hoc features. The performance of different combinations of these features is analyzed in comprehensive ablation experiments.

We participated in two challenges of the International Workshop on Semantic Evaluations (SemEval 2015). Our system performed robustly and reliably in the sentiment classification of tweets task, where it ranked 9th among 40 participants. However, it proved to be the state-of-the-art for measuring degree of sentiment of tweets with figurative language, where it ranked 1st among 35 systems.

Acknowledgments

I would not be able to finish this study without the endless support of my supervisor Dr. Sabine Bergler. I am thankful to her for making me a part of CLaC Labs.

I am also thankful to my defense committee, Drs. Leila Kosseim and René Witte, for their salient feedback and comments.

I am lucky to have wonderful members of family and relatives who always supported me. I am always thankful to them.

The time I had with my friends in CLaC Labs is unforgettable. It was an invaluable experience to be with CLaCers.

My friends Nihat, Ada and Alp helped and supported me a lot that I will never forget. Thank you guys. Besides, I will always remember the ones who gave their energy, love, and time for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentiment analysis is a field of natural language processing that studies subjective expressions from natural language texts (Pang and Lee, 2004), (Turney, 2002), (Wiebe et al., 2005). General aims of sentiment analysis are to identify and/or extract subjectivity from various media, and to determine someone's attitude, judgement, or evaluation possibly towards an entity (Pang and Lee, 2008), (Pang et al., 2002), (Hu and Liu, 2004). This entity may be a person, brand, product, organization, service, or topic (Liu and Zhang, 2012).

(Pang and Lee, 2008) gives a general, detailed overview of aspects, approaches, challenges and applications of sentiment analysis. Aspects of sentiment analysis can be listed as subjectivity classification (Mihalcea et al., 2007), polarity classification (Kim and Hovy, 2004), detecting degree of polarity (Turney, 2002), emotion classification (Frantova and Bergler, 2009, Strapparava and Mihalcea, 2007), and source and/or target extraction of sentiment carriers (Broß, 2013, Zhai et al., 2011). Solutions to these tasks have been proposed using classification, information extraction, and regression supervision techniques. These aspects are applied on different genre of data: reviews, blog entries, newspapers, news headlines, emails, described comprehensively in (Pang and Lee, 2008) and tweets as microblog entries in (Kiritchenko et al., 2014).

(Liu, 2012) approaches sentiment while defining what an opinion is. In their earlier study, according to (Liu, 2010), an opinion is an insight that carries sentiment about an entity or an aspect of it, which is conveyed by a source in a time. This definition of opinion is a quintuple $opinion = (e_j, a_{jk}, s_{ijkl}, h_i, t_l)$, (Liu, 2010), where:

- e_j is a target entity of sentiment
- a_{jk} is an aspect of entity e_j
- h_i is the source of the opinion, or the opinion holder
- t_l is the time opinion conveyed
- s_{ijkl} is sentiment toward opinion on the aspect a_{jk} of the entity e_j from the opinion holder h_i, at the time t_l

Extracting sentiment of an opinion in the composition of (Liu, 2010) automatically requires a detailed linguistic analysis and is a challenging task (Liu, 2012), (Broß, 2013). Even if it is fulfilled, sentences, documents, reviews etc. may contain more than one opinion on the same or different subject with different sentiment, which the overall sentiment is wondered. Here the task of entity-level sentiment extraction evolves into others: sentence-level and document-level sentiment classification or sentiment degree measuring. In body of this study, sentiment analysis of a recent type of documents, microblog entries are studied.

1.1 Objectives

This thesis concentrates on *sentiment polarity classification* and *measuring degree of sentiment* in microblog entries, tweets.

Sentiment polarity classification is a binary or ternary classification of polarity classes of text instances into polarity class of *positive, negative,* or *neutral.* In the cases of a text having opinions with different polarities, the goal becomes to vote for the overall dominant sentiment for the text. In Example 1, an example tweet demonstrates such a case beginning with a slightly negative statement (a hedged negated positive one: *may not be the right*), then the final statement dominates it with a positive statement (an hedged negated negative one: *can prevent the spread of HIV*).

Example 1. *It may not be the right fit for everyone, but PrEP therapy can prevent the spread of HIV, read here: http://t.co/fOpwPX3RvZ #HIVPrevention* (tweet sentiment label: positive)

Measuring degree of sentiment can be modelled as a regression or a multiple-class classification problem of capturing sentiment degree of a text on a scale. Detecting how many stars a review has and measuring the sentiment strength of a tweet on a scale (e.g. from -5 to 5, or from 0 to 1) are examples of this fine-grained sentiment analysis sub-problem. The middle point of a scale can be interpreted as neutral or sentiment-free. From this point to the highest one, degree of positivity increases and degree of negativity increases towards the lowest point.

Since social media has become an environment that contains massive amounts of public opinion on current events, news, and products; it has attracted increasing attention. In International Workshop on Semantic Evaluation series[1] (SemEval), sentiment analysis of tweets challenges have become ongoing challenges with high ratio of attendance and participation (Nakov et al., 2013), (Rosenthal et al., 2014), (Rosenthal et al., 2015), (Ghosh et al., 2015).

Evaluating a system on these challenges is advantageous since different approaches on the same task are compared. It also gives the chance of reporting clearly what the system achieves and what it does not on the same data, with the predefined evaluation process. The system prepared in this thesis competed in two challenges of SemEval 2015[2]. The system ranked 9th among 40 participant team's submissions in sentiment polarity classification of tweets task (SemEval 2015's Task 10) and ranked 1st among 35 submissions of 15 teams in measuring degree of sentiment of figurative language in tweets task (SemEval 2015's Task 11). Our results show that our system performance is robust and reliable.

In 2013 and 2014 SemEval challenges of Sentiment Analysis of Tweets, state-of-the-art systems, (Mohammad et al., 2013) and (Zhu et al., 2014), leveraged bag-of-words approaches along with lexical and tweet-specific structural features. In contrast to bag-of-words approaches, we test negation[3] and modality[4] context aware lexical features, encoded with frequency-based and sentiment association score[5] based attributes; constructing smaller feature space, in which it is easier to analyze features: Interoperability of different sentiment lexica with the linguistic notions of negation and modality is investigated as well as the influence of adding negation and modality context in a comprehensive

[1] https://en.wikipedia.org/wiki/SemEval

[2] SemEval challenges provide a period for participants to build and train a model on a training set, and then release a test set during evaluation period. We prepared our feature space in order to use supervision techniques to train models for the tasks by following SemEval's timeline.

[3] In linguistics, negation is a form of a grammatical category. It is used to express the falsity of a statement.

[4] Modality expresses possibility, necessity, or ability.

[5] A sentiment association score of a term is a continuous value that shows the degree of its sentiment within a scale.

ablation study.

Although the informal nature of tweets makes it relatively difficult to analyze their syntax, we need syntactical information in order to detect context of the linguistic notions of negation and modality. Therefore, we apply specific annotation-level preprocessing heavily relying on having optimal tokens. Then, in a pipeline structure we use syntactical information to extract context, to be used to create context-aware sentiment lexical features.

1.2 Contributions

Combining sentiment lexical features with negation and modality contexts for the domain is a novelty of our study. The impact of negation context has been investigated for sentiment domain in the last decade, which will be discussed in the next chapters. However, the effect of modality in this particular domain has not been studied. We report on the usage and impact of modality context with or without negation context. Organizers of a shared task challenge in SemEval 2016 announced that they will give a place to modality as well as negation. Thus, this supports our conclusion that modality context has an impact in the sentiment domain.

As another contribution, we compiled a lexical resource automatically, called Gezi. The top-performing system of SemEval 2013 Sentiment Analysis of Tweets challenge, (Mohammad et al., 2013), created the NRC Hashtag Lexicon, on which they reported that the resource contributes well in their ablation studies. We extend their approach by adding data cleaning, annotation-level preprocessing for optimizing tokenization, negation context detection, and creating dependency triples[6]. We compare these resources both in our ablation studies and in coverage of manually-curated lexica studies. We conclude that Gezi outperforms the NRC Hashtag Lexicon in both studies.

The techniques in our pipeline for linguistic parsing of tweets cover preprocessing for optimizing tokenization and optimizing part-of-speech tags specifically for tweets. These steps are applied to determine the linguistic context more accurately. In our Chapter 7, we illustrate that these techniques improve results significantly.

[6] pairs of terms having a grammatical relation, extracted from dependency trees; to be mentioned in detail in Chapter 5.

Finally, attaching additional *prior polarity*[7] classes to the traditional ones for automatically compiled lexica by using their sentiment association scores is another novelty which improves results significantly as well.

These contributions are also demonstrated in (Özdemir and Bergler, 2015a) and (Özdemir and Bergler, 2015b) (to be published) while demonstrating the usefulness of the lexical features interoperating with linguistic scope and tweet-specific structural ad-hoc features on SemEval challenges.

1.3 Outline

The layout of the thesis is as follows: a background of sentiment, negation and modality, and literature review for sentiment analysis of tweets is introduced in Chapter 2. Chapter 3 explains the methodologies of annotating tweets for sentiment analysis purposes and gives details of SemEval sentiment datasets used in this study and other datasets in the literature. The implementation of our system is described in Chapter 4, 5, and 6 where we convey our study's resources, tools, and system's pipeline with its design along with all their usages. Baseline and system development experiments are provided in Chapter 7 where SemEval tasks experiments have been introduced with their definitions, evaluation metrics and official results. Chapter 9 concludes the study, after we analyze the results and the error cases in the participated challenges in Chapter 8.

[7] Prior polarity is a term's assigned initial polarity without any context (i.e. *good* has a prior polarity of *positive*). Here the additional prior polarities are strong version polarity classes of *positive* and *negative* (i.e. *strong positive* for *magnificent*).

Chapter 2

Background

To understand the subject of sentiment analysis of tweets, certain attributes of tweets and basic sense of sentiment are needed to be illuminated. After explaining what tweets are, what sentiment is, and the linguistic notions that we incorporate with them, we give pointers to related work in sentiment analysis on tweets, particularly the usage of negation context in the sentiment domain.

2.1 Nature of Tweets

Twitter[1] is a popular microblogging platform. Millions of users create content via a special message type, the *tweet*, which is constrained to 140 characters. The social media platform makes tweets publicly available via its easy-to-use API. the Twitter API[2] can be polled for any tweet or limited to tweets with certain terms. Thus, it is relatively easy to create tweet collection for any purpose.

Tweets generally contain informal content. Words may be spelled incorrectly (possibly intentionally), abbreviated, elongated, or all-capitalized or all-lowercased. Signs signalling emotions, called emoticons[3], slangs, URLs and hashtags[4] frequently occur in tweets. Therefore, extra effort is required to recognize and analyze these terms and create accurate tokens, and therefore part-of-speech (hereafter POS) tags and parse trees for tweets in order to process them accurately. In our pipeline,

[1] https://www.twitter.com

[2] https://dev.twitter.com/rest/public

[3] projection of facial expressions in written communication, i.e. :) for a smile

[4] concatenated terms begin with a #, generally used to identify message(s) on particular events or topics, for instance #TheNorthRemembers

we address these points to some extent which will be explained in the Chapter 6.

We use the Stanford Parser to generate parse trees (Socher et al., 2013a) and dependency relations (de Marneffe and Manning, 2008) that convey syntactic information, which builds a tree for a sentence statistically.

Example 2's syntax and dependency parse trees are introduced in Figure 1 and Figure 2.

Example 2. *Working for 5 hours should be fun.*

The syntax parse tree illustrated in Figure 1 represents the words and the phrases hierarchy of the sentence of Example 2.

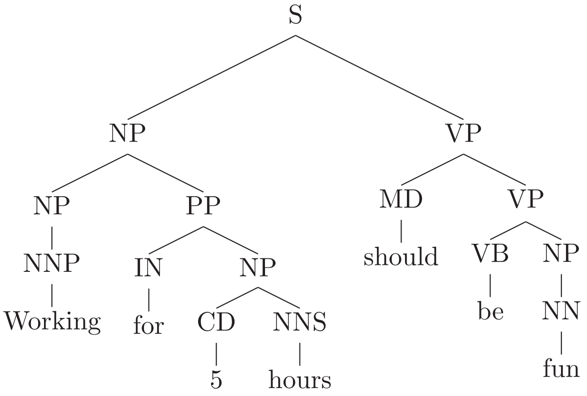

Figure 1: Syntax tree of Example 2.

Figure 2 shows the special grammatical relations called dependency relations between words in Example 2. The relations are basic triples of *dependency type(head,dependent)* (i.e. *nmod(Working,hours)* or *cop(fun,be)*).

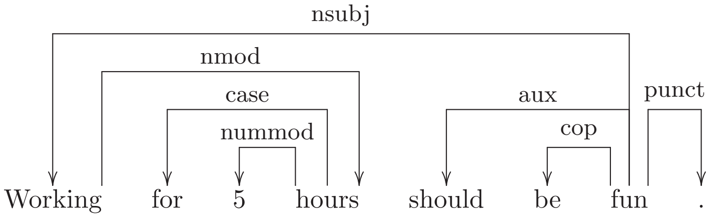

Figure 2: Dependency tree of Example 2.

2.2 Sentiment

Sentiment is a feeling, can be expressed as a view of or attitude towards an event or situation Pang et al. (2002). Many ways exist to express sentiment: a look, gesture, reaction, or text can convey sentiment. We are interested in analyzing sentiment when expressed in natural language texts. When sentiment is conveyed via text, it could be expressed by using formal or informal, literal or figurative language with or without idiomatic expressions. Interpreting sentiment is domain-sensitive, which depends on context from one domain to another (Andreevskaia and Bergler, 2008).

2.3 Linguistic Notions

In language, extra-propositional meaning aspects can be used to hedge or speculate on propositions. Negation and modality are elements of extra-propositional meaning aspects which modify meaning of statements. (Morante and Sporleder, 2012) gives a detailed background for the phenomena.

When scoping over sentiment-laden terms, negation and modality change the effect of the terms, therefore these linguistic notions need to be examined for sentiment analysis as a semantic interpretation.

Recent work in CLaC Labs on embedding predication by (Kilicoglu, 2012), negation by (Rosenberg, 2013), and modality by (Rosenberg et al., 2012) emphasized that embedding constructions of syntax have influence over the meaning of constituents. We include negation and modality for analyzing sentiment.

(Kilicoglu, 2012)'s dissertation brings out embedding predications have influence over semantics of constituents, including negation and modality. We use modality triggers only on modal part-of-speech tag from (Kilicoglu, 2012) and negation triggers from (Rosenberg, 2013) to find the scope of those linguistic phenomena with the pipeline built by (Rosenberg et al., 2012).

On the training set of Task 10B dataset size of 6822, modality triggers are present in 1,785 tweets and negation triggers are in 1,356. This is 1195 for modality and 1,990 for negation in measuring sentiment degree of tweets with figurative language training dataset, which has a size of 7,383. Consequently, they occur frequently and these notions have the potential to influence the results to a measurable degree.

2.3.1 Negation

Negation is a phenomenon in logic which we use to change the truth value of a proposition. In logic, addressing the scope of negation is trivial (Horn, 2001). In language, this is more complex due to propositions are not in simple first order logic (Zwicky and Pullum, 1983).

In sentiment analysis, negation is one of the key points, since it can change or reverse meaning of sentiment bearing words or phrases. Studies using negation prove finding and utilizing scopes of negation triggers or even only the presence of negation triggers, increases performance (Günther and Furrer, 2013, Kökciyan et al., 2013, Mohammad et al., 2013). However, those studies propose a simple rule-based detection of negation scope by annotating the tokens either between a negation trigger and punctuation sign or immediate term before and after a negation trigger. Our system uses a syntax-aware negation trigger and scope detection system developed by (Rosenberg, 2013) as an extensive work on negation. Since scope of negation does not necessarily end with a punctuation nor the word next to the trigger, a system uses syntax is expected to be more precise.

The effect of negation on the text that negation scopes over varies when the effect is interpreted. (Kennedy and Inkpen, 2005) encode negation as a simple reverser of polarity values (multiplying them by -1). Yet, negation does not always reverse the effects of the sentiment carriers, especially for the terms having strong prior polarities. For instance, this judgement illustrates the point: *not terrible* does not necessarily mean *excellent*. Since negated sentiment carriers do not default to one fixed resulting sentiment value but have to be assessed in their linguistic context, we do not resolve the negation numerically, but encode its occurrence in a separate feature (i.e. *negated-positive, negated-negative*), a technique similar to (Kennedy and Inkpen, 2006). When computing the association scores for sentiment carriers in negation context, it results in multiplying sentiment association scores by *-0.5* in order not to reverse its impact.

In Example 3, a sentence from a tweet with negation is given where negation trigger is emphasized and its scope is underlined. The syntax tree of Example 3 produced by the Stanford Parser can be seen in Figure 3. Negator simply traverses tree after identifying the negation trigger, using rules, here the scope of the negation trigger is found with the rule: if a trigger exists in a verb phrase which has no following-sibling, the scope is the triggers' following-sibling(s).

Example 3. *El Classico on a Sunday Night is*n't <u>*perfect for the Monday Morning*</u> *!!*

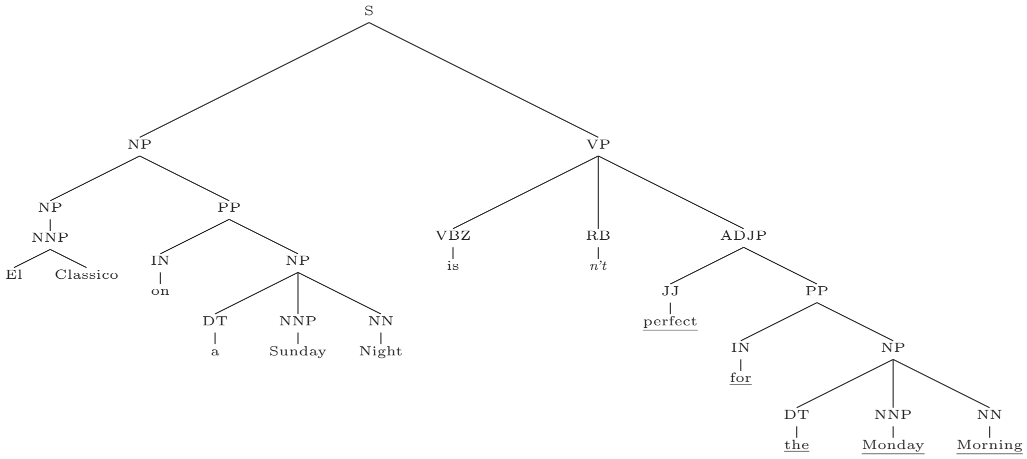

Figure 3: Syntax tree of Example 3.

When modality and negation brings extra-propositional meanings together, their scope has a more nuanced effect: *it may be good, it is not good, it may not be good.* These phenomena are addressed in a series of challenge tasks (BioNLP Shared Tasks 2008-2010 (Kim et al., 2009), CoNLL 2010 (Farkas et al., 2010), *Sem (Morante and Blanco, 2012), QA4MRE (Morante and Daelemans, 2012)) in recent years .

2.3.2 Modality

Modal verbs embed possibility, necessity, or ability to the sentences in which they occur(Palmer, 2001). It lessens the factuality of a statement: beliefs, thoughts, or hypotheticals are phrased with modality frequently (Palmer, 2001). In formal text like newspapers, reports or documentations, modality is a rare phenomenon. But, in informal text like tweets, it is frequent and carries important meaning aspects. Using them with the verbs in different tenses and aspects serves to present different conditional worlds[5] which may have never existed. The phenomenon may be used for describing the emotions like wishes, hopes, or regrets (Palmer, 2001).

The BioNLP Shared Task series (Kim et al., 2009) gave special interest in speculative language, and QA4MRE (Morante and Daelemans, 2012) acknowledged the interaction of the notions of negation and modality. (Rosenberg et al., 2012)'s approach on the QA4MRE pilot task dominated the competition where modality is treated the same as negation is treated. We follow (Rosenberg et al.,

[5] An example is: *This film might have been an excellent one.* The example implies that the film was not excellent due to a possible reason.

2012)'s treatment in our study.

(Kilicoglu, 2012) gives a detailed taxonomy of modality categories with modality triggers. We only make use of modals from those triggers which have modal part-of-speech tags.

We assume that the epistemic, deontic, dynamic, and evidential contexts of modal triggers are weakened and we do not differentiate the ontological modality types. Therefore, we do not distinguish between them. For the reason that modality does not directly change or reverse polarity types of sentiment carriers to a particular degree, we explicitly encode their occurrences in separate features as we did for negation (i.e. *modality-positive, modality-negative*). From the point of view that the context of modality is weakened, we dampen the sentiment association score of a sentiment carrier when it occurs in scope of modality. Therefore, association scores for sentiment carriers in modality context are multiplied by *0.5* dampening the effect of a sentiment carrier between its own association score and its neutralized version, while computing the scores.

We also investigate how utilizing modality scoping over sentiment carrier words, functions in sentiment analysis of tweets. (Rosenberg et al., 2012) developed and tested their system for detecting scope of modality with syntax-aware heuristics.

A sentence with modality is given where the trigger is in italics and its scope is underlined in Example 4. The dependencies of Example 4 are demonstrated in Figure 4 and syntax tree in Figure 5 is produced by the Stanford Parser.

Example 4. *Working for 5 hours* should <u>*be fun*.</u>

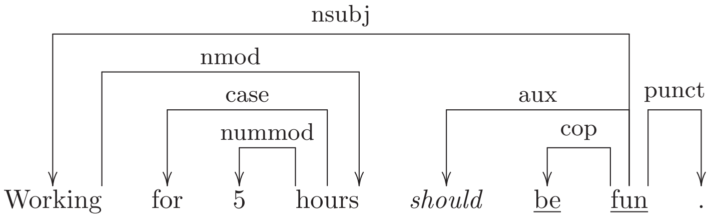

Figure 4: Dependency tree of Example 4.

2.4 Literature Review

This section addresses salient work on the sentiment analysis of tweets and usage of negation scope in sentiment analysis. The approaches and main contributions of these studies are exhibited. The

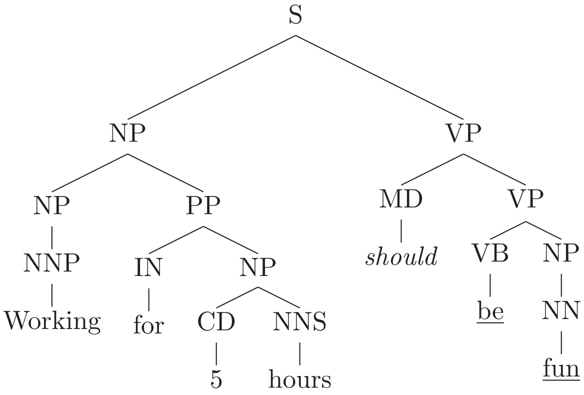

Figure 5: Syntax tree of Example 4.

background for the systems prepared for SemEval challenges of sentiment analysis of tweets are demonstrated in following chapters after we describe the challenges.

2.4.1 Sentiment Analysis of Tweets

In the recent decade, social media made massive amounts of data. Since tweets may contain opinions, thoughts, or beliefs on anything; applying sentiment analysis for tweets became a recognized challenge in the field. Below, selected significant studies are emphasized.

(Go et al., 2009) apply binary classification sentiment analysis on tweets using a distant supervision technique. In this study, the Twitter API was polled to retrieve tweets with positive and negative emoticons. Emoticons' polarity classes were attached to retrieved tweets as message-level polarity class label. This idea was used for automatically labelling a large amount of data to avoid the cost of manually labelling. After preprocessing and balancing the tweet set, they had 800,000 tweets for each polarity class. They manually collected and labelled the test set from different categories of topics. Test set contains 182 positive and 177 negative tweets. By using unigrams, bigrams and POS tags, (Go et al., 2009) trained Naive Bayes, Maximum Entropy and Support Vector Machine classifiers. They reported their best result to be 82.2% accuracy with an SVM classifier using only unigrams.

(O'Connor et al., 2010) make use of negative and positive terms from the MPQA Subjectivity Lexicon (Wilson et al., 2005) to classify tweets in a rule-based approach by counting positive and negative terms. The aim of the study is to compare this approach to respectable public opinion polls for consumer confidence and political opinion in time series with respect to the rule-based sentiment analysis. This work concludes that even this simple approach may replicate the expensive and time-consuming public polls.

(Barbosa and Feng, 2010) collected tweets from online sentiment detection tools[6] with predicted sentiment labels. They created a two-level sentiment framework similar to (Pang and Lee, 2004, Wilson et al., 2005) whereby polarity classification is applied after binary objectivity/subjectivity classification. The tweets that are labelled as neutral by the sources are accepted as objective classes for training for the first phase. For subjective tweets as negatives and positives; if disagreement on the polarity labels of tweets is found between different online tools, those tweets are removed. For objective tweets that are labelled as neutral by the tools, if they contain sentiment-laden terms from the MPQA Subjectivity Lexicon, they are also removed. After preprocessing tweets for subjectivity detection, the training datasets contain approximately 200,000 tweets, nearly half are objective and the other are subjective. For polarity classification, 71,046 positive and 79,628 negative tweets form the training set. The features used in the study are POS tags, polarity frequencies (switching was applied in the case of negation between negative and positives) twitter-specific features based on URLs, hashtags, retweets, punctuations, emoticons, and upper-case words. Using an SVM classifier on 1,000 manually-labelled tweets for development and 1,000 tweets for test set, they had an 18.1% error rate on subjectivity classification and an 18.7% error rate on polarity classification. They report that POS tags and polarity frequencies are the features useful to the classifier for polarity classification. (Barbosa and Feng, 2010) concludes that their approach is robust and effective due to the fact that they did not use anything for word representations but abstractly mapped their polarities into their feature space. This is a similar characteristic that our study shares as it does not use any bag-of-words like approaches to represent words themselves.

(Nielsen, 2011) creates a sentiment lexicon manually by giving polarity classes and assigning valence scores to 2477 English formal and informal sentiment-laden words and a few phrases, called AFINN-111. The assignments of the features to the terms of the lexicon was done by the author

[6] Those tools are automated tweet sentiment polarity classifiers and no longer available anymore due to Twitter's updated content sharing policy.

manually by checking the terms' contexts in tweets. While compiling the lexicon, the words which may have ambiguity (i.e. *mean*) or may have different sentiments in different contexts (i.e. *surprise*) are excluded from the resource.

(Agarwal et al., 2011) build models for both binary and ternary sentiment polarity classification. They experiment with a unigram model as a baseline, a tree representation model developed for tweets and a feature space model with traditional features. The traditional feature types created in this study (Agarwal et al., 2011) are count-based features such as count of number of positive adjectives, negative emoticons or hashtags; real number based features such as sum of the prior polarity scores of words with POS tag of adverbs; boolean features such as the presence of exclamation marks and the presence of capitalized text. They make use of a 8,753 manually-annotated tweets, and then they balance tweets equally for each polarity class by keeping 5127 tweets, 1709 from each class. They apply a generic preprocessing to tweets: they assign prior polarities to emoticons and they replace these 5-level polarity classes' names with emoticons. For informal acronyms, they replace them with the original phrases. Negation triggers are normalized to *not* tokens. Usernames and URLs are normalized as well, and elongated terms' repeated characters have been reduced to three characters. They use an SVM classifier and experiment with 5-fold cross validation over their balanced training set. They report a 75.39% accuracy on binary polarity classification using unigrams with the features they created. For ternary polarity classification, their best performer system is their tree representation with the features which achieved a 60.83% accuracy.

2.4.2 Usage of Negation for Sentiment Analysis Purposes

(Pang et al., 2002) bring the idea of using the simple heuristic that defines negation scope between a negation trigger and the first punctuation mark after the trigger. In their bag-of-words model, the words in the scope of negation are added to the space separately with a negation marker.

(Kennedy and Inkpen, 2006) extract scope of negation triggers by traversing syntax trees. While in their early approach, (Kennedy and Inkpen, 2005), they initiated a simple heuristic of negating a sentiment-laden word if only it is directly preceded by a negation trigger. There they did not discover a significant influence of using negation. They applied the theory from (Polanyi and

Zaenen, 2006) by modifying the polarity class and sentiment score features. In their extended study (Kennedy and Inkpen, 2006), they report that leveraging negation improves their results.

(Choi and Cardie, 2008) expand negation triggers by also appending implicit negation triggers (i.e. *fail*) in an application of compositional inference rules for sentiment polarity classification of expressions. They extract scope of negation via simple syntactic patterns. In their approach (Choi and Cardie, 2008), they also model multiple negations in their compositional semantic models, where they demonstrate their compositional semantic models outperforms traditional bag-of-words approach.

(Socher et al., 2013b) create a Sentiment Treebank containing 215,154 phrases of 11,855 parsed sentences. The aim of this work is to understand compositionality by using a deep learning method over labelled phrases in the sentiment treebank. The phrases in sentiment treebank is annotated by three annotators where they scale phrases' sentiment to 25 fine-grained values. (Socher et al., 2013b) adapts recursive neural networks to represent input vectors of the phrases more explicitly to a model called recursive neural tensor networks. After giving their deep learning system details and parameters, they evaluate their system on a fine-grained five class level[7] and a binary sentiment classification on both phrases and sentences. They split the dataset of sentences to 8,544 instances for training, 1,101 instances for development and 2,210 instances for testing. For binary classification they remove the neutral labelled instances, with the set's sizes dropping to 6,920, 872, and 1,821 instances accordingly. They report their accuracy results on the fine-grained model on phrases as 80.7% and on sentences as 45.7% performing significantly better than other approaches. For binary classification their deep learning system's accuracy results are 87.6% on phrases and 85.4% on sentences. All these results are reported as state-of-the-art results at the time. (Socher et al., 2013b) treats negation separately for negative and positive sentences. They claim that negation over positive phrases changes its sentiment to negative; however this is not the case for negating negatives. Negating negatives makes a negative phrase less negative according to them.

[7] These classes are defined in the study as *negative, somewhat negative, neutral, somewhat positive*, and *positive.*

Chapter 3

Sentiment Annotation of Tweets

Well-known tweet corpora in the field and those that were used in this study are listed with their attributes, samples of tweets and their annotations in the following sections.

3.1 Datasets of the Study and Their Annotations

Creating a consistent dataset containing tweets and tweet-level annotations for sentiment analysis requires several phases. Tweets are retrieved and annotated for their sentiment by multiple annotators. The annotations need to have a level of agreement between annotators to be reliable and this level is measured by calculating inter-annotator agreement (Nakov et al., 2013). Further analysis may even be required to check whether annotators from crowdsourcing systems swindle while annotating the data (Ghosh et al., 2015).

Such datasets have been generated for the sentiment track of SemEval challenges in order to evaluate a variety of systems developed for the problem of sentiment analysis of tweets. We test our system by participating in two SemEval 2015 sentiment track challenges.

The sentiment track of SemEval contained two tasks towards message-level sentiment analysis of tweets. A message polarity classification task, Task 10B[1] (Sentiment Analysis in Twitter) (Rosenthal et al., 2015), where the task is polarity classification of tweets as positive, negative and neutral. The other task is Task 11[2] called Sentiment Analysis in Figurative Language in Twitter, where the task

[1] http://alt.qcri.org/semeval2015/task10/

[2] http://alt.qcri.org/semeval2015/task11/

is to measure the degree of polarity of tweets on a scale between -5 and 5 either by treating sentiment values on the scale as discrete or continuous.

Both datasets of the tasks are collected via the Twitter API and annotated by using crowdsourcing services, namely Amazon Mechanical Turk[3] for Task 10 and CrowdFlower[4] for Task 11. (Sabou et al., 2014) describes and compares different crowdsourcing approaches in order to find best practices for annotation of corpora via crowdsourcing.

3.1.1 Polarity Classification of Tweets Datasets

In SemEval Task 10B, message level polarity classification, the data instances are tweets and their annotated polarities.

Task organizers streamed millions of tweets via the Twitter API. Then, those tweets are filtered in two steps: The first one by detecting named-entities via a named entity recognition system which is trained on tweets (Ritter et al., 2011), then the tweets occurring with selected named entities of different topics are kept to make sure the collection has tweets on different topics. Secondly, the tweets were filtered using a lexicon, SentiWordNet[5] (Esuli and Sebastiani, 2006), such that if a tweet does not contain a term from the lexicon with an association score greater than 0.3 as positive or negative, it is removed. The reason behind this filtering is to balance the the data more in term of polarity classes, given that the tweets containing named entities reported to be heavily objective or neutral (Nakov et al., 2013).

Filtered tweets were annotated as positive, negative, or neutral by Amazon Mechanical Turk workers. After an inter-annotator agreement study tweets with polarity labels became ready.

Task 10B consisted of one single training and development set, with separate test sets for consecutive years of 2013, 2014, and 2015. The 2013 test set (Nakov et al., 2013) contains a set of tweets and a set of SMSes. For 2014 (Rosenthal et al., 2014), the test set contains a set of tweets, a set of Live Journal items and a special tweet set of sarcasm. There are a general tweet set and a tweet set of sarcasm in the 2015 test set (Rosenthal et al., 2015).

The distribution of instances per polarity class is shown in Table 1 for each data set.

[3] https://www.mturk.com/mturk/

[4] http://www.crowdflower.com/

[5] SentiWordNet is a resource that keeps association scores for terms in scale between 0 and 1 separately for polarity classes of negation and positive where 1 indicates full strength of polarity and 0 represents no strength of polarity.

Corpus	Positive	Negative	Neutral
2013 Twitter training	3,662	1,466	4,600
2013 Twitter development	575	340	739
2013 Twitter test	1,572	601	1,640
2013 SMS test	492	394	1,207
2014 Twitter test	982	202	669
2014 Twitter sarcasm test	33	40	13
2014 LiveJournal test	427	304	411
2015 Twitter test	1,038	365	987

Table 1: Polarity classification of tweets datasets instance distributions.

The training set developed in 2013 is also used for the next year's challenges. In addition, participants were allowed to use the development set in the training set.

The task organizers created a script for participants to retrieve tweets by IDs of tweets. The reason that tweet texts are not directly given is the privacy policy of Twitter[6]. Accordingly, participants might end up downloading different sizes of training and development sets since tweets are volatile[7].

As expected, we were not able to download all the tweets for the above-mentioned reason. In Table 2, we compare the sizes of original merged training and development test sets and the one we downloaded and merged. This comparison concludes that 15.21% instances of these sets are not present in the downloaded sets. Several participants share the same situation and the size of data of each participant varies. Therefore, almost every participant trains on different sizes; which is open to criticize.

Type	Positive	Negative	Neutral	Total
original	4,237	1,806	5,339	11,382
downloaded	3,560	1,496	4,595	9,651

Table 2: Comparison of training and development datasets' sizes.

The tweets retrieved by the script, a set of twitter status IDs, the polarity label and the tweet content is an instance in dataset separated by tabs.

After retrieving a tweet via the script, it is registered in a line of the file the set is kept as

[6] The policy states that Twitter can only share its public content with users, but avoids sharing between users themselves.

[7] Users may delete their messages or they can make their accounts private, then their content is not reachable.

illustrated in Example 5.

Example 5. *264070796710932480 $< TAB >$ 113186 $< TAB >$ positive $< TAB >$ Can't wait to go to the UGA vs. Ole Miss game on Saturday! #toomuchexcitement*

Although tweets' gold labels are agreed upon by multiple annotators, those gold annotations include mis-labelling (see Example 6a) and borderline judgements (see Example 6b) as expected.

Example 6. Questionable gold standard judgements:

a. *I haven't eaten chicken nuggets since I was like 6 or 7.. Who wants to get some McDonald's with me tomorrow?* `(gold:Negative)`

b. *Class early in the mornjng = it's bedtime! But do get to see my Sam tomorrow :)* `(gold:Neutral)`

3.1.2 Figurative Language in Twitter Dataset

The organizers of Task 11 (Ghosh et al., 2015), collected tweets of figurative language such as metaphor, sarcasm and irony via the Twitter API.

The tweets were collected via a set of search queries and a list of hashtags defined by the organizers. The tweets were retrieved between 1st of June, 2014 and 30th of June, 2014. After a post-processing, the final training set consists of 8,000 tweets containing 5,000 sarcastic, 1,000 ironical and 2,000 metaphorical tweets. Those tweets were annotated in an 11 point scale between -5 and 5 by using a crowd-sourcing system, CrowdFlower. Released data had two formats: discrete integer-valued and continuous real-valued sentiment scores. Although organizers give information regarding how many annotations per figurative language types have been labelled, they did not release the types' annotations within the dataset. Thus, we will have restricted analysis of the task in terms of figurative language types.

The inter-annotator agreement process of data simply has three parts. Firstly, organizers of the challenge, 7 researchers, labelled a portion of the data. Secondly, each tweet was annotated by 7 different CrowdFlower annotators. In the last step, for scammers, who do not properly apply the task, a scammer detection was generated to discard them and their annotations. Accordingly, The annotations created by the researchers on a portion of the data in the first step was used to compare the annotations with scammer's. Standard deviations are calculated between the annotations of the researchers and the annotators to distinguish scammers.

Given that privacy policy of Twitter does not let 3rd party users share the content of any tweet, IDs of tweets were published. Since tweets are volatile, not all the tweets in the collection were retrieved. In Table 3, we demonstrate the sizes of original training, trial and test sets and the ones we downloaded.

Type	Training	Trial	Test	Total
original	8,000	1,000	4,000	13,000
downloaded	6,905	478	3,999	11,382

Table 3: Training, trial and test data sizes of the figurative language task.

The data instances contain tweet status ID, tweet sentiment value and tweet content in a tab-separated format:

Example 7. *463066661671534592 < TAB > -2.86 < TAB > 3 Full days at uni this week. Soo looking forward to it. #Sarcasm*

The instances were also released with their integer sentiment values for the participants who see the problem as a classification instead of a regression one:

Example 8. *463066661671534592 < TAB > -3 < TAB > 3 Full days at uni this week. Soo looking forward to it. #Sarcasm*

In the task, participants were allowed to add a trial annotated data containing 1,000 tweets to the training dataset. Therefore, we added 478 retrieved tweets from the trial data tweet instance to our training set, so totally we had 7,383 tweet instances in our final training set. In Table 4, the distribution of the instances of test and final training set per integer sentiment value is shown.

Sentiment Value	-5	-4	-3	-2	-1	0	1	2	3	4	5
Test Size	4	99	836	1540	679	297	168	154	200	110	4
Training Size	4	434	2741	2546	811	297	171	206	107	52	5

Table 4: Distribution of the instances of test and final training set of the figurative language task.

The distributions of the test and final training sets are mainly negative between -1 and -4, and the distributions of the test and training sets are similar.

Chapter 4

Resources and Tools

In the last decade, many resources and tools have been compiled and developed for sentiment analysis. In this chapter, an overview on traditional and novel lexical resources, and the tools developed for tweet processing is given. Finally, basic statistics on the SemEval challenge datasets are supplied.

4.1 Lexical Resources

Several lexical resources have been compiled for sentiment analysis purposes. They serve well for the domain as a core aspect since they carry prior polarity information of lexical entries (Pang and Lee, 2008), (Kiritchenko et al., 2014). In this section, these resources are split into two main categories: manually versus automatically created lexica.

4.1.1 Manually-created Lexica

In our study, we use three well-known and well-studied manually-created lexica, namely the MPQA Subjectivity Lexicon, The Opinion Lexicon of Bing Liu, and AFINN-111.

- ***The MPQA Subjectivity Lexicon*** (Multi-Perspective Question Answering) by (Wilson et al., 2005), is manually created with prior polarities for 8,222 terms in three categories:

positive, negative, and neutral. The words also have pseudo-POS tag[1] and subjectivity information. The latter are given as features depending on whether the words are strongly or weakly subjective. Many terms labelled as negative and positive from The General Inquirer (Stone et al., 1966)[2] and (Hatzivassiloglou and McKeown, 1997) are also added into the MPQA lexicon.

- ***The Opinion Lexicon of Bing Liu*** by (Hu and Liu, 2004), contains 6,789 manually selected terms that are labelled as either positive or negative. It also includes misspellings, slangs, and social-media related terms in it, by being updated over years.
- ***AFINN-111*** by (Nielsen, 2011) is a lexicon of 2,477 terms manually rated for valence scores with an integer between -5 (negative polar) and 5 (positive polar). During the generation of this lexicon, the words which may have had an ambiguity were excluded from the resource. While selecting the valence scores for words, their contexts were checked in tweets.
- ***SentiWordNet*** by (Esuli and Sebastiani, 2006), is a resource that assigns sentiment association scores of positivity, negativity, and objectivity to every WordNet (Fellbaum, 1998) synset.

In this study, the SentiWordNet lexicon is not used so as not to bias our system given that every instance of message-level polarity classification data has been reported to contain at least one term from this resource.

The General Inquirer's entries have been used in other lexica. Even, polarity values of the entries of the lexicon is changed by the creators of other lexica, since different senses were developed for some of entries over the years.

4.1.2 Automatically-created Lexica

- ***NRC Hashtag Sentiment Lexicon*** by (Mohammad et al., 2013) is a lexicon that assigns sentiment association scores to its entries calculated by pointwise mutual information[3] (Church

[1] In natural language processing community, when POS tags are generally used and are assumed to be used in Penn Treebank Style which are detailed tags (Marcus et al., 1993). However in MPQA, entries are attributed with shallow POS tag information, such as *noun, verb, adverb*, or *adjective*.

[2] An old, general purpose lexicon which also has sentiment polarity attributes as negative and positive categories for entries.

[3] PMI is a association measure of coincidence calculated between a pair of attributes of two discrete random variables assuming that the variables are independent of each other.

and Hanks, 1990). The method that is used to produce the lexicon is streaming tweets via hashtags of some sentiment-laden terms[4] and calculated by pointwise mutual information (hereafter PMI). For the production of the resource, tweets were streamed by using the Twitter API every four hours from April to December 2012 with the seed hashtags. In total, 775,000 tweets were retrieved and labelled as positive or negative according to the seed hashtag each carried. Then, unigrams[5], bigrams[6], and skip bigrams[7] are extracted to be the entries of the lexicon: it contains 54,129 unigrams, 316,531 bigrams and 480,010 skip bigrams extracted from the tweet collection.

- ***Gezi***[8] ***Lexical Resource*** by (Özdemir and Bergler, 2015a), is our automatically created lexicon. 20 million tweets are streamed with hashtags of some sentiment-laden terms, then after processing tweets in a pipeline, PMI scores are calculated for unigrams, bigrams, and dependency pairs. The lexicon is first compiled with tweets collected between December 2013 and May 2014. In Chapter 5, the details of the resource will be described.

In Table 5, the characteristics of the lexica used in this study are listed. These characteristics are sentiment polarity classes held, type of creation, whether terms are attached with association scores or not, whether terms have POS tag information or not, and the number of entries per lexicon.

Lexicon	Categories	Creation	Assoc. sc.	POS tag	Size
AFINN-111	positive-negative	manual	yes	no	2,477
Bing Liu	positive-negative	manual	no	no	6,786
MPQA	positive-neutral-negative	manual	no	yes	8,222
NRC	N/A	auto	yes	no	54,128
Gezi	N/A	auto	yes	yes	376,863

Table 5: Comparing attributes of lexica.

As it will be mentioned in Table 10, the NRC Sentiment Lexicon has 316,383 bigrams and 308,791 skip bigrams; while, the Gezi Lexicon has 922,773 bigrams and 850,074 dependencies.

In this study, we use the NRC Sentiment Lexicon (Mohammad et al., 2013). The resource has been updated with a newer version in (Zhu et al., 2014), however this latest version has not been

[4] 32 positive and 36 negative seed hashtags are built from sentiment-laden terms taken from the Rogets Thesaurus (Roget, 2011) manually.

[5] Unigrams are unique single terms that occurred in the collection.

[6] Bigrams are two unique consequtive terms that occurred in the collection.

[7] Skip bigrams are two unique ordered non-consequtive terms occurred in the collection.

[8] ***I dedicated my work on compiling this lexical resource to Gezi Park protests.***

released yet.

4.2 Lists and Tools Used in the Study

4.2.1 Lists

Emoticons We classified emoticons into positive and negative using the Wikipedia list of emoticons[9]. We encoded frequency-based features for positive and negative emoticons.

Contrastive Discourse Markers Tweets may contain both negative and positive polarity showing a comparison, a conclusion or an interpretation on things with contrasting opinions. Contrastive discourse markers are often used in those situations. We used a feature of the frequencies of contrastive discourse adverbs and a contrastive discourse connector[10]. Yet, we observe that they need to be investigated more carefully and encoded according to their semantics.

4.2.2 Tools

We process tweets in a pipeline structure in the GATE framework (Cunningham et al., 2013). After processing tweets, encoding features for them and creating feature spaces by making use of the tools in our pipeline, feature spaces for each tweet dataset are exported. These exported spaces are used to run supervised machine learning algorithms in a tool, called Weka (Witten and Frank, 2011).

In this section, the main processing tools used in our pipeline are described.

Tokenization In recent decades, many tokenizers are developed for various genre of data. For tweets, a specific tokenizer (Bontcheva et al., 2013) and Twokenizer (Gimpel et al., 2011) are built. These tokenizers are developed specificly for the informal language of tweets. However, they can create the tokens differently[11]. As a design selection, we kept our twitter-specific tokens in single tokens. Also, in order to tokenize non-twitter-specific part, we selected to use a well-known and domain independent tokenizer. Therefore, we use ANNIE (Cunningham et al., 2002) and Twokenizer (Gimpel

[9] 33 positive and 25 negative emoticons are used from http://en.wikipedia.org/wiki/List_of_emoticons .

[10] Contrastive discourse adverbs that we used are *even though, although, though, despite the fact, in spite of, on the contrary, however, but, yet, on the other hand, instead, whereas, nevertheless, nonetheless* and the contrastive discourse connector is a conjunction which is (*but*).

[11] i.e. for one hashtags are tokenized into one single token and for another they can be tokenized into two as hash sign and the term.

et al., 2011) tokenizers together to create hybrid token sets. This is covered in Section 6.2 in detail. In addition, (Bontcheva et al., 2013) created a specific tokenizer for hashtags in order to segment concatenated tokens. This module could be used to segment and analyze hashtags reflecting phrases or sentences. However, this module is built in a rule-based fashion and it can easily fail tokenizing ambiguous cases of concatenated tokens[12]. Given the automatically compiled lexica used in our study covers a huge number of hashtags[13], we prefer not to create tokens of hashtags, rather choose their lookup information with sentiment association scores from automatically compiled lexica.

Sentence Splitting and Normalization The Stanford Sentence splitter is used. Sentence splitting is required to segment sentences of tweets to create parse trees in a more accurate way. Pruning the sentences after segmentation is useful to skip some tweet specific tokens as hashtags, usernames, URLs, interjections at the beginning or the end of a sentence. The reason behind this is tweet-specific tokens are generally not parts of the sentences.

Part-of-Speech Tagging Running a POS tagger trained on a tweet corpus that creates Penn Treebank formatted tags is a plus in the case that those are needed to be sent to linguistic parser with the tokens they belong to. The CMU POS tagger (Gimpel et al., 2011) is used for tagging the tokens not only for having linguistic information, also having tweet-specific tokens for hashtags, URLs, retweets and usernames. We used a model that produces Penn Treebank formatted tags which is trained on (Ritter et al., 2011)'s POS tag annotated data.

Named Entity Recognition Ritter's NER (Ritter et al., 2011) is used for finding named entities (NE). NEs are unified as one single token and if they carry sentiment triggers, those triggers are deleted since proper names can carry sentiment-laden terms which do not necessarily reflect their sentiment.

Parsing The Stanford parser (Socher et al., 2013a) is used to create both constituent and dependency parse trees.

[12] #thisappletsmerock is tokenized into tokens of *#,this,apple,ts,me,rock* where the expected tokens are *#,this,app,lets,me,rock*.

[13] 10,240 hastags in NRC Hashtag Sentiment Lexicon's unigrams (of 54,128) and 165,342 hashtags in Gezi Lexicon's unigrams (of 376,863).

Finding Scopes of Negation and Modality CLaC Labs' Negation and Modality Scope Module is used. Since negation changes meaning in language in completely different ways, having negation triggers and scopes provides the chance of creation features for the sentiment triggers which are under scope of the negation.

4.3 Basic Statistics on the SemEval Tweet Datasets

This section presents basic statistics on training and development data of SemEval 2013 Twitter shared task.

For each tweet instances' polarity class of sentiment, we record the presence of some linguistic phenomena: negation, modality, specific sentiment-laden idiomatic expressions, contrastive discourse marker conjunctor and adverbs, and named entities. We call frequency of a phenomenon in tweets as *tweet frequency*[14], and call frequency of a phenomenon in all collection as *collection frequency*[15].

Negation is detected for both implicit and explicit negation triggers of the Negator module, and the way for modality is detected constrained version of Kilicoglu's modality triggers. Frequent sample idiomatic expressions with their different forms[16]) are kept in a gazetteer to see its behaviour on data. Contrastive discourse markers are also kept in a gazetteer list including a conjunction[17] and 14 adverbs[18].

4.3.1 Statistics on SemEval Task 10B Training and Development Sets

In Table 6, in the merged second column, *total* represents the total of three polarity classes, and the rest represents separate classes' information. The second row shows which dataset that the polar class information falls into, and the third row shows how many instances per phenomenon each dataset has. Other columns show the percentage of tweets that the phenomenon is present in. Note

[14] This is similar to *document frequency* in Information Retrieval, where we count the total number of times a phenomenon occurs. Therefore, even if a phenomenon occurs more than once in a tweet, we only add one count to the collection frequency.

[15] This is similar to *corpus frequency* in Information Retrieval, which we count a phenomenon's all occurrences in the collection.

[16] *can't wait, cannot wait, can not wait, cant wait, can't expect more, cannot expect more, can not expect more, cant expect more, can't do better, can't make better, don't miss, do not miss, dont miss, don't forget, do not forget, dont forget*

[17] *but*

[18] *even though, although, though, despite the fact, in spite of, on the contrary, however, but, yet, on the other hand, instead, whereas, nevertheless, nonetheless*

that even though the phenomenon occurs more than once in a tweet, it is counted as one, since we want to record occurrence only.

class	total		negative		neutral		positive	
dataset	train	dev	train	dev	train	dev	train	dev
phenomenon/size	6,822	1,042	976	194	3,320	463	2,526	385
implicit negation	0.04	0.05	0.09	0.10	0.03	0.03	0.03	0.03
explicit negation	0.15	0.20	0.36	0.48	0.08	0.11	0.15	0.17
implicit & explicit negation	0.01	0.02	0.02	0.04	0.00	0.01	0.01	0.02
modality	0.26	0.28	0.32	0.36	0.24	0.25	0.27	0.28
idiomatic expression	0.02	0.02	0.01	0.00	0.00	0.00	0.05	0.06
contrastive discourse cc	0.06	0.07	0.10	0.09	0.04	0.06	0.06	0.07
contrastive discourse adv	0.02	0.01	0.04	0.03	0.02	0.01	0.02	0.01
named entities	0.92	0.91	0.90	0.91	0.93	0.93	0.91	0.89

Table 6: Percentages of linguistic phenomena in 2013's training and development sets.

Comparing those phenomena's collection frequencies to their tweet frequencies (presence) is interesting to assess their difference. In Table 7, the collection frequencies and percentages of the phenomena are shown on tweet polarity classification training and development test sets. The only significant difference is in named entities in tweets as its occurrence nearly twice as its presence. This may be due to either the fact that the tweets have been collected with a certain topic or its nature. Nonetheless, the other phenomena have slight increases.

dataset	training		development	
phenomenon/size	6,822		1,042	
	percent	frequency	percent	frequency
implicit negation	0.04	298	0.05	50
explicit negation	0.17	1132	0.22	233
implicit & explicit negation	0.03	187	0.02	23
modality	0.30	2028	0.34	355
idiomatic expression	0.02	124	0.02	24
contrastive discourse cc	0.06	384	0.07	72
contrastive discourse adv	0.02	138	0.01	15
named entities	1.76	12,005	1.72	1,795

Table 7: Collection frequencies and percentages of linguistic phenomena in 2013's training and development sets.

4.3.2 Statistics on SemEval Task 10B 2013, 2014 and 2015 Test Sets

The presence of those phenomena are also extracted from test sets over three years, presented with their percentages in Table 8. The only notable unexpected distribution is the selected idiomatic expressions in 2015 such that even the size of the 2015 test set is greater than 2014's, idioms in 2015's are far less than 2014's.

dataset	Twitter 2015 test		Twitter 2014 test		Twitter 2013 test	
phenomenon/size	2,390		1,853		3,813	
	percent	frequency	percent	frequency	percent	frequency
implicit negation	0.05	126	0.02	41	0.04	137
explicit negation	0.17	403	0.14	258	0.16	601
implicit & explicit negation	0.01	25	0.01	16	0.01	45
modality & explicit negation	0.16	376	0.10	186	0.13	504
modality & implicit negation	0.03	78	0.01	22	0.02	90
modality	0.31	748	0.24	447	0.29	1104
idiomatic expression	0.02	37	0.03	64	0.02	79
contrastive discourse cc	0.07	171	0.06	109	0.06	243
contrastive discourse adv	0.02	44.00	0.02	33	0.02	62.00
named entities	0.92	2,208	0.90	1,671	0.90	3,431

Table 8: Tweet frequency of linguistic phenomena in tweet test sets.

4.3.3 Statistics on Task 11 Training and Test Sets

We extracted the same structural statistical information from figurative language dataset as well. In Table 9, it can be observed that figurative language datasets have more negation triggers than modality ones in contrary to tweet polarity classification datasets. This observation shows that those datasets have different trigger distributions as different characteristics.

dataset	training		test	
phenomenon/size	7,383		3,999	
	percent	frequency	percent	frequency
implicit negation	0.04	263	0.05	187
explicit negation	0.23	1,727	0.30	1,194
implicit & explicit negation	0.01	53	0.01	51
modality	0.16	1,195	0.21	857
idiomatic expression	0.01	67	0.01	33
contrastive discourse cc	0.05	398	0.08	335
contrastive discourse adv	0.02	128	0.02	95
named entities	0.24	1,753	0.25	1,014

Table 9: Tweet frequency of linguistic phenomena in Task 11 training and test sets.

Chapter 5

Compiling a Lexical Resource Automatically

In this chapter, a contribution of this work, an automatically compiled lexicon with distant supervision, namely Gezi, and its compilation technique are described with detailed analysis. Also, we use Gezi on a pilot subtask of determining the strength of the terms, namely Task 10E, and we present its usage in the task.

5.1 Gezi Lexicon

As described in Section 4.1, the Gezi resource consists of unigrams, bigrams and dependency triples is an automatically created lexicon for sentiment analysis purposes. Modelled after (Mohammad et al., 2013)'s approach, we streamed tweets containing a set of hashtags of sentiment-laden terms. The set of 35 positive and 34 negative seed hashtags are obtained from the Oxford American Writer's Thesaurus (Moody and Lindberg, 2012) by expanding the adjectives *good* and *bad*[1]. Table 10 compares Gezi's methodology with the NRC Hashtag Lexicon.

As Table 10 demonstrates, skip bigrams are not used in our study, dependency triples are used instead since those triples have grammatical relation between two terms. Skip bigrams, however,

[1] Adjectives from the synsets of *good* and *bad* are manually selected and this process is iteratively applied for the selected adjectives.

	NRC	Gezi
Collection Period	April to December 2012	December 2013 to May 2014
Collection Size	775K	20M
Source Hashtags	select tweets with 32 pos, 36 neg hashtags (Roget's)	select tweets with 35 pos, 34 neg hashtags (OAWT)
Preprocessing		remove tweets with little text and duplicates (→ 7.2M)
Negation	positive, negative	mark negation context: positive, negated positive negative, negated negative
POS		parse, POS annotate entries

Size	unigrams(54,128)
	bigrams(316,383)
	skip bigrams(308,791)
		unigrams(376,863)
		bigrams(922,773)
		dependency triples(850,074)

Association Scores	scores between -5 and 5	scores between -8 and 8

Table 10: Methodological comparison of the Gezi with NRC Hashtag Sentiment Lexicon.

does not require a grammatical relation between the pair of terms. So, by skip bigrams' nature, they contain noisy entries as pairs occurred non-consecutively in a tweet. Therefore, in our feature space, we keep unigrams and bigrams of the Gezi and NRC Hashtag Sentiment Lexicon separately to compare with each other as feature subsets. Feature subsets of dependency triples are also encoded separately, and we exclude them from comparison.

At the end of the streaming process, nearly 20 million tweets were collected. After this step, three phases are applied to generate the resource in a pipeline; namely *data cleaning and preprocessing*, *data processing*, and *exporting resources*.

5.1.1 Data Cleaning and Preprocessing

A tweet from the collection is removed under one of the following conditions listed.

- If the tweet contains non-English words determined using language detection[2]
- If the tweet is a duplicate such as a retweet or a modified tweet
- If both a negative and a positive seed hashtag exist in a tweet

[2] http://code.google.com/p/language-detection/

- If non-content tweet tokens[3] represent more than 40% of the tokens of tweet

All URLs are removed from the content of tweets. Then, all the tweets are annotated with the polarity class of their seed hashtag.

5.1.2 Data Processing

We process the tweet collection with the modules that we designed in our sentiment pipeline. After tokenizing the tweet collection with our hybrid tokenizing module, We split sentences in tweets and POS tag the tokens. Then, we apply linguistic parsing module to parse tweets and identify negation triggers and their scopes. After using these modules from our sentiment pipeline, we iterate over tweets and register frequencies of unigrams, bigrams and dependency triples (type-head-modifier) in the context which they occur by taking negation scope into consideration. For instance, if a term occurs in a positive-annotated tweet where it is not under negation scope, its positive count is incremented, whereas if it is in a positive-annotated tweet and under negation scope, then its negated positive count is incremented. Thus, reflection of different contexts of the terms in the lexicon is achieved by discriminating negated and non-negated context. In addition, we keep terms with different part-of-speech tags separately in the resources, in case different POS tags have different impacts in terms of sentiment[4].

In order to illustrate the process of registering terms with their polarity frequencies, Example 9 is to be processed. Assume that the tweet in the example is the *n+1*th one in the collection that we are processing while compiling the Gezi resource for unigrams[5]. Also let us assume we have recorded frequencies of some terms from Example 9 in Table 11 after we processed *n* tweets.

Example 9. *I am* not *relaxed, in fact I am pretty nervous #sad*
sentiment:NEGATIVE

In Table 11, terms with their POS[6] are represented with or without negation scope information[7]. In the example, the scope of negation (underlined), is determined by a negation trigger (emphasized).

[3] URLs, hashtags, and usernames are defined as non-content tweet tokens since they point to concepts without being in the content of a clause.

[4] For instance, the adjective sense of the term *just* is positive, however the adverb sense remains neutral.

[5] The tweet's sentiment is determined by the hashtag #sad as it occurs in our negative seed hashtags.

[6] Here the term *pretty* has an *adverb* part-of-speech, nevertheless the term's *adjective* entry is also represented to emphasize keeping the terms with their POS tags.

[7] In the Gezi resource, negation scope information was used. However, the hypothetical situation of not using negation is demonstrated as well in Table 11.

negation usage	term	POS	positive		negative		negated neg.		negated pos.	
			before	after	before	after	before	after	before	after
used	*relaxed*	adjective	145	145	12	12	15	16	3	3
not used	*relaxed*	adjective	148	148	27	28	n/a	n/a	n/a	n/a
used	*pretty*	adjective	132	132	23	23	11	11	0	0
not used	*pretty*	adjective	132	132	34	34	n/a	n/a	n/a	n/a
used	*pretty*	adverb	89	89	32	33	8	8	9	9
not used	*pretty*	adverb	98	98	40	41	n/a	n/a	n/a	n/a
used	*nervous*	adjective	5	5	97	98	3	3	9	9
not used	*nervous*	adjective	14	14	100	101	n/a	n/a	n/a	n/a

Table 11: Representation of some terms in Gezi uniterms before and after processing Example 9.

After processing the *n+1*th tweet of Example 9, the updated registries are illustrated in the *after* columns in Table 11. Apart from how we generate the resource, this example directs to two nuance points. The first is that when negation context is determined, the terms could be registered into appropriate categories. In Example 9, a positive term, *relaxed*, which is in the context of negation occurs in a negative labelled tweet. If negation context information were not used, the term's negative frequency would have been incremented. This would not be right due to the fact that its negated content is used to convey a negative sentiment (negated neg.), not itself. Second, keeping term's different POS tagged entries may be useful as we saw in the sample of *pretty*. The adverb form could be used both in negative and positive phrases, but its adjective form is generally used in positive phrases.

Association scores are calculated using PMI (see Equation 1) per term, after registering counts of the terms to positive, negative, negated positive, and negated negative categories by scanning all the collection. Experiments are conducted for using fine-grained polarity labels for terms, as to be demonstrated in the Section 7.3.

In Equation 1, the general formula of PMI is illustrated as giving the relevance between individual distributions and coincidences of the concepts of a and b.

$$PMI(a,b) \equiv log_2 \frac{p(a,b)}{p(a) * p(b)} = log_2 \frac{p(a|b)}{p(a)} = log_2 \frac{p(b|a)}{p(b)} \tag{1}$$

As an application of Equation 1 to measure sentiment scores, Equation 2 calculates the mutual information of a term for a particular sentiment class (positive or negative since we only have these labels for tweets in the collection), where the concepts are namely a term, *term*, and a , *sentimentClass*, that those may occur with, and N is the number of tweets in the collection. Here, *p(a,b)* is a term's frequency occurring in a tweet with a particular sentiment class, *p(a)* is the term's frequency in whole collection, and *p(b)* is basically the count of the tweets with the particular sentiment class.

$$PMI(term, sentimentClass) = log_2 \frac{freq(term, sentimentClass)}{freq(term) * \frac{freq(sentimentClass)}{N}} \tag{2}$$

PMIs of terms related to negative and positive classes can be calculated since the collection of tweets has positive and negative labels, prepared for creating a sentiment resource. Equation 3 leverages those information to calculate a sentiment score for a term:

$$sentimentScore(term) = PMI(term, positive) - PMI(term, negative) \tag{3}$$

Expanding Equation 3 shows the equivalence in Equation 4, where R is the ratio of negative instances to positive ones in the collection.

$$\begin{aligned} sentimentScore(term) &= log_2 \frac{freq(term, positive) * \frac{freq(negative)}{N}}{freq(term, negative) * \frac{freq(positive)}{N}} \\ &= log_2 \frac{freq(term, positive) * freq(negative)}{freq(term, negative) * freq(positive)} \\ &= log_2 \frac{freq(term, positive) * R}{freq(term, negative)} \end{aligned} \tag{4}$$

5.1.3 Exporting Resources

In this phase, we export unigram and bigram terms into specific gazetteers and dependency triples into a database. For unigram and bigrams, we set variables to give terms prior polarity classes. These classes may be chosen as either 3-class as positive, negative and neutral, or 5-class by adding strong positive and strong negative.

```
procedure Task10E($terms, scores, GeziUnigramSet, GeziBigramSet, aFinnSet$)
    $normalizeBetween0to1(GeziUnigramSet)$
    $normalizeBetween0to1(GeziBigramSet)$
    $normalizeBetween0to1(aFinnSet)$
    for all $term \leftarrow terms$ do
        if $GeziBigramSet.contains(term)$ then
            $term.score \leftarrow GeziBigramSet.get(term).score$
        else if $GeziUnigramSet.contains(term) \quad OR \quad aFinnSet.contains(term)$ then
            if $GeziUnigramSet.contains(term) \quad AND \quad aFinnSet.contains(term)$ then
                $term.score \leftarrow (GeziUnigramSet.get(term).score + aFinnSet.get(term).score)/2$
            end if
            if $GeziUnigramSet.contains(term)$ then
                $term.score \leftarrow GeziUnigramSet.get(term).score$
            end if
            if $aFinnSet.contains(term)$ then
                $term.score \leftarrow aFinnSet.get(term).score$
            end if
            if $term.inScopeOfNegation()$ then
                $term.score \leftarrow term.score * -0.5$
            end if
        else
            $term.score \leftarrow 0.5$
        end if
        $scores.add(term.score)$
    end for
    return $scores$
end procedure
```

5.2 Validating the Gezi Lexicon on a Task

5.2.1 Determining Strength of Association of Terms

In SemEval 2015 Task 10, a pilot subtask was given to provide association scores to terms extracted from tweets (*Subtask 10E*). A test set that consists of words or phrases was given to be associated with scores between 0 and 1 where 1 stands for maximum of positive strength and 0 does for maximum negative.

5.2.2 Validating Gezi on Subtask 10E

We followed a simple, rule-based approach: We normalize Gezi and AFINN-111 sentiment association scores between [0,1] to treat them equally. Then, we apply the algorithm below to give association scores to the terms supplied in the task.

5.2.3 Evaluation metrics

The evaluation metrics are Kendall and Spearman rank correlation coefficients (Nelson, 2001) for subtask 10E between gold values of words or phrases and predicted values. Gold values are human judgements from the compilation of the NRC lexicon (Kiritchenko et al., 2014).

5.2.4 Results

Our simple rule-based and lexicon-driven system submitted for Task 10E ranked 4th among 10 submitted systems in both correlation coefficient evaluations. Our Kendall rank correlation coefficient result is 0.584 where all results range between 0.625 and 0.254, and our Spearman rank correlation coefficient result is 0.777 where results range between 0.817 and 0.373.

5.3 Determining Prior Polarity from Association Scores

PMI score derived lexica have continuous association scores for polarity. However, they do not possess prior polarity classes attached to their entries. Those classes could be useful for feature encoding since they are discrete in contrast to association scores. So, here this idea triggers another: prior polarity classes can be assigned to terms by selecting thresholds. For instance, terms will be deemed positive if their score is above 2 and as negative if their score is below -2, while the rest is considered neutral. However, these thresholds must be chosen in a way that association scores would fit.

We segment polar classes by defining thresholds. We assess the segmentation in two different ways: in two polarity classes as positive and negative and in four polarity classes as strong positive, positive, negative, and strong negative by excluding neutrals in both ways. We attribute the lexica by using them with 2 polar classes and 4 finer level polar classes in an empirical method to overcome which resource give better insight to analyze sentiment. The neutral category (association score close to 0) denotes terms occurring nearly as often in tweets labelled negative as in positive ones and a clear classification is not possible. Another reason is that terms like stopwords, determiners, or sentiment-free terms are likely to occur both in positive and negative labelled tweets. This makes these terms are gathered around the score of 0, where we call this region as *neutral zone*.

Association scores for Gezi and the NRC Hashtag Sentiment Lexicon do not range over the same scale. Their scales are not design points for automatically created lexica, rather results of the mutual information formula. Scales of NRC Hashtag Lexicon and Gezi Lexicon are different due to different collection size of tweets, and the term's frequencies are relatively higher in Gezi than the NRC Hashtag Sentiment Lexicon.

Here, a term is considered positive if it occurs at least twice as often in positive tweets as in negative tweets. Thus positive terms have association scores greater than 1, calculated by PMI. For a term to be categorized as strongly positive, its score has to be greater than the geometric mean of the positive space $gMean(1, 8) = \sqrt{8} = 2.83$. Similarly, a term is considered negative if its association score lies below -1 and as strongly negative if its association score lies below -2.83. The region between positive and negative ranging from -1 to 1 is accepted neutral zone.

The NRC Hashtag Sentiment Lexicon has association scores between -5 and 5. For our experiments with it, we segment the same way as we did for Gezi.

polar class	NRC unigrams	Gezi unigrams	NRC bigrams	Gezi bigrams
strong-positive	3,390	24,739	20,636	73,087
positive	10,276	108,685	53,059	228,255
negative	8,447	62,333	49,566	199,443
strong-negative	3,605	24,639	27,104	105,169
no neutral	25,721	220,339	150,368	605,957
all	54,126	376,863	316,531	922,773

Table 12: Fine grained prior polarity class distribution of automatically compiled lexica.

In the neutral zone, stopwords, determiners, or sentiment-free terms are frequent and these terms are mostly sentiment irrelevant. As a design decision, we remove the terms in the neutral zone to increase the ratio of sentiment relevant terms. After removing these terms from the NRC Hashtag Sentiment Lexicon, the size of unigrams decreases 52.48%, from 54,129 to 25,721, bigrams from 316,531 to 150,368. In the Gezi Lexicon, this removal resulted in the size of unigrams decreasing 41.49%, from 376,863 to 220.399, bigrams 34.33%, from 922,773 to 605,958. These differences and the distributions of the terms per polarity class are shown in Table 12.

Giving prior polarity classes to terms and excluding terms in the neutral zone for automatically created lexica changes their characteristics in terms of size and categories, represented in Table 5. In Table 13 the updated characteristics are listed.

Lexicon	Categories	Creation	Assoc. sc.	POS tag	Size
AFINN-111	positive, negative	manual	yes	no	2,477
Bing Liu	positive, negative	manual	no	no	6,786
MPQA	positive, neutral, negative	manual	no	yes	6,886
NRC	strong positive, positive, negative, strong negative	auto	yes	no	25,721
Gezi	strong positive, positive, negative, strong negative	auto	yes	yes	220.399

Table 13: Comparing attributes of lexica without neutral terms.

Chapter 6

System Design and Implementation

In this chapter, the flow of our system architecture is described with the usage of the phenomena discussed in earlier chapters as the tools, lexical resources, and linguistic notions.

Experiments conducted in this thesis aim to predict tweets' sentiment labels. These rely on a series of processes in a pipeline structure in order to create feature spaces and to apply supervision techniques to test sets where training sets are supplied. Models are developed for these sets by projecting their features extracted via leveraging lexical resources and several tools to determine linguistic information.

6.1 System Architecture

The system architecture of this study is simply processing tweet data within a pipeline and creating models for supervision techniques for sentiment prediction. We process tweets in the GATE framework by (Cunningham et al., 2013), extract features, create feature combinations and run supervised machine learning algorithms using Weka by (Witten and Frank, 2011) on these combinations.

The general flow of system architecture is described in Figure 6. In the next sections, each step of the flow will be explained.

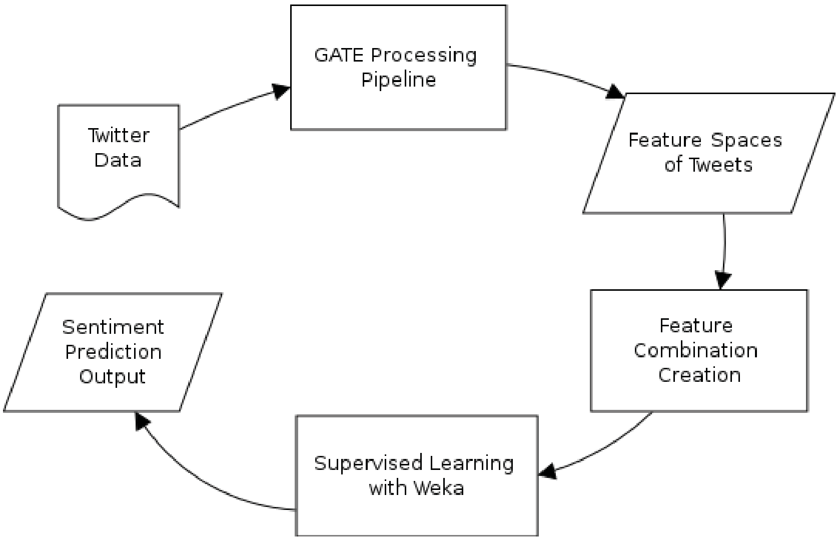
GATE Processing Pipeline
Twitter Data
Feature Spaces of Tweets
Sentiment Prediction Output
Feature Combination Creation
Supervised Learning with Weka

Figure 6: General system architecture.

6.2 Sentiment Pipeline

The main study for processing data, which is made in a pipeline structure was generated in GATE by (Cunningham et al., 2013), which is an NLP framework contains multiple tools and easy-to-develop and easy-to-plug the tools in.

In order to show how lexical features incorporated with linguistic phenomena, dependency driven features and other specific types of features are encoded within the pipeline, modules in the pipeline are explained in this chapter. The order of the modules are as in Figure 7, which is the extended part of the *GATE Processing Pipeline* from Figure 6.

6.2.1 Hybrid Tokenizer Module

This pipeline tokenizes with the ANNIE (Cunningham et al., 2002) and Twokenizer (Owoputi et al., 2013) (CMU tokenizer hereafter) to two different token sets. We run the Stanford sentence splitter (Manning et al., 2014) to segment sentences in tweets and the CMU POS tagger (Gimpel et al., 2011) on the CMU token set. Because the CMU tokenizer gives accurate results on tweet-specific tokens, only the tokens related with specific POS tags[1] are added into the hybrid set. Then, the non-added CMU token set is compared with the ANNIE set to check if each compared tokens have

[1] *hashtag* (HT), *retweet* (RT), *username* (USR), and *URL* (URL). Also it works accurately for *cardinal numbers* (CD).

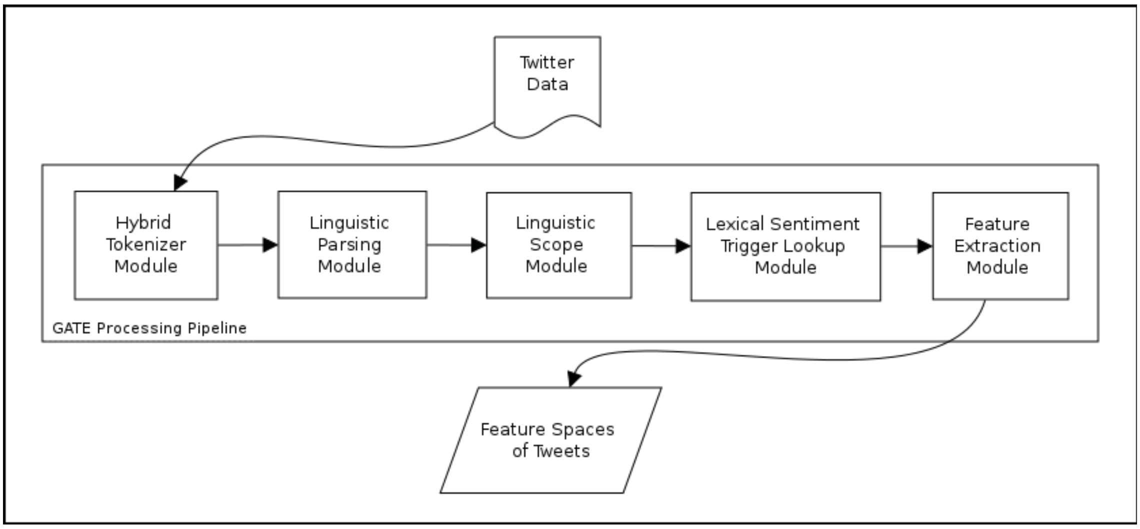

Figure 7: General pipeline overview.

the same start and end offsets. The tokens in the ANNIE set which do not match with the rest of the CMU tokens, are selected to be kept in the hybrid set as well. This comparison of offsets is to check whether there is an intersection between the selected tokens. Thus, the specific tokens from CMU, mostly of social media related POS tags are selected and the other tokens are taken from ANNIE.

The main reason for using a hybrid tokenizer module is that the CMU tokenizer is tokenizing social media based tokens accurately[2], however the ANNIE tokenizer is much more effective in tokenizing general text, even in the cases of misspelled text. So, the more accurate tokens of pipelines are selected. For instance, the word *isn't* is tokenized as one single token by the CMU tokenizer and two tokens by ANNIE tokenizer as *is* and *n't* in which we cannot make use of negation trigger *n't* with CMU, however we can with ANNIE. On the other hand, a URL is tokenized to multiple tokens by ANNIE and one single token by CMU. One single token seems more meaningful, so in this case CMU token works.

After sentence splitting, we prune sentences by excluding tweet specific tokens as hashtags, usernames, URLs, interjections occurring at the beginning or the end of sentences of tweets. The reason behind this is tweet-specific tokens are generally not in the context of the sentences. In

[2] The design of the tokenization rules are built specially for the informal language of tweets.

addition, this normalization step relaxes statistical parsers, which are not trained with those tweet-specific tokens.

Another step for optimizing tokens is finding named entities and fusing them into a single token. In the next steps, lexical triggers found in the body of named entities are to be removed as well, because named entities may not reflect the triggers' sentiment.

The importance of having accurate tokens is that more accurate parse trees are created which leads to a better context detection of linguistic notions. Consequently, more accurate features are expected to be created since they heavily rely on linguistic phenomena.

In our experiments, we have observed that using the hybrid tokenizer method outperforms using only ANNIE or CMU tokenizers. We demonstrate these experiments in Section 7.3.

Figure 8 shows this module as an overview of the flow.

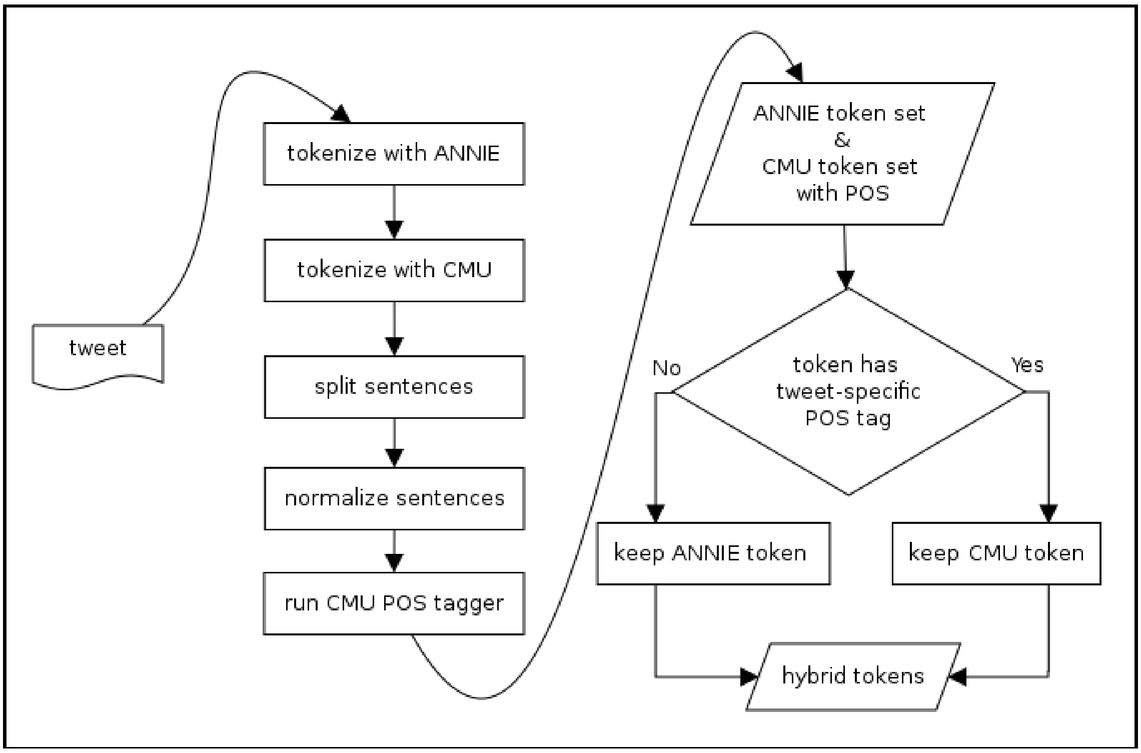

Figure 8: Hybrid tokenization module.

6.2.2 Linguistic Parsing Module

After creating hybrid tokens and sentences of tweets with the hybrid tokenizer module, the CMU part-of-speech tagger is used for tagging the tokens of sentences. Then, a simple sentence normalization step was applied. In this step, tokens with specific part-of-speech tags occurring at the beginning and end of the sentences are removed. Those specific tags are social media related such as retweets, URLs, hashtags and usernames. The aim here is to give those sentences to the parser in order to obtain better parse trees.

The GATE wrapper around the Stanford Parser (Klein and Manning, 2003) (de Marneffe and Manning, 2008) hides the capability of the parser accepting tokens with partial POS tags[3]. For that, the wrapper is updated so that it can accept partial POS tags. This process was applied because sentences in tweets contain some POS tags that are incompatible with Penn Treebank style, namely *hashtag* (HT), *retweet* (RT), *username* (USR), and *URL* (URL). When tokens having these POS tags are sent to the parser without their POS tags, parser statistically predicts POS tags for those tokens in Penn Treebank style format, create the linguistic parse, and then gives its syntax parse trees and dependency parse trees.

6.2.3 Finding Negation and Modality Module

In order to incorporate lexical sentiment triggers with negation and modality, those phenomena's scopes are found by using their trigger sets (Rosenberg et al., 2012) in a module called Negator, developed by (Rosenberg and Bergler, 2012).

(Rosenberg and Bergler, 2012) explain how to use this pipeline as it is necessary to supply tokens, sentences, syntax trees and dependencies to Negator. GATE's morphological analyzer and verb phrase chunker are used before identifying the triggers and their scopes.

There are three different negation scope types that the pipeline produces: explicit, implicit and affixal negation scopes. However, in our system, those different types are not distinguished. The modality scope has one single type produced in the pipeline.

[3] The Stanford Parser can parse a sentence, which some of its tokens may have their POS tags and others may not. However, in the wrapper, the parser only accepts tokens either all with POS tagged or none.

6.2.4 Lexical Trigger Lookup Module

In this module, entries of lexical resources are looked up and annotated if matches are found on data. If features exist per lexical term, such as part-of-speech tag information or PMI score, those have been carried over. As a post-processing step, part-of-speech tag compatibility is checked for sentiment lexical entries. If an entry has POS tag and is matched with text, tokens are checked if their POS tags are compatible with the entry's.

Dependency Score Lookup phase in this module mainly relies on a resource built while compiling Gezi sentiment unigram and bigram triggers. The resource is a database containing the dependency triples extracted from Gezi tweet collection by using Stanford Parser's dependency module (de Marneffe and Manning, 2008). Dependency triples are dependency relations with head and modifier, described in (de Marneffe and Manning, 2008) in detail. An PMI score has been calculated for each triple by keeping how many times they occurred in negative and positive tweets in the corpus, in the same way it is calculated for Gezi unigrams and bigrams.

Notice that the tweets, sourced from twitter, are assumed to be either positive or negative according to the initial hashtag that they carry. For using the resource efficiently, all the association scores with the dependency relations from millions of tweets are kept in a database. Therefore, any dependency relation of tweets in training or test sets could be queried as to whether there are any association scores for dependencies. Then, these scores are attached to the head token annotation.

Thus, given that scores of the dependency relations are queried, two different feature groups are created per tweet as :

a. Scores of each specific dependency type

b. Polarity counts (for negative and positive polarities only) of each specific dependency type

6.2.5 Feature Extraction Module

This last module in the pipeline makes use of annotations produced to extract features and map them into a feature space. The aim is simply to export those feature spaces to run supervised machine learning experiments. We generate those experiments in the Weka environment (Witten and Frank, 2011), thus we create Weka-specific *arff* formatted files per tweet dataset.

This last part of the pipeline will be investigated in the next section in detail.

6.3 Feature Creation and Feature Spaces

Our approach to sentiment analysis of tweets incorporates sentiment carrier terms with their negation and modality context. We use supervised techniques for our approach on the tweet datasets. Firstly, in our feature space, we represent different polarity classes of terms for each lexicon by distinguishing and separating them in which linguistic notion's scope that those terms occur (*inverse linguistic scope*). In simple words, we encode features per tweet looking up the lexical terms' *polarity classes*, *lexical resources*, and *inverse linguistic scope.* The reasons we use polarity classes of terms instead of terms themselves, are to represent features per lexicon as polarity classes no matter how different terms the lexica keep, and to keep the feature space smaller. We define this group of features as *primary feature set* and we group features for each lexical resource in a feature subset. Secondly, we add features to the space that are frequencies of specific annotations (i.e. emoticons, implicit-explicit negation triggers, modality triggers, named-entities, contrastive discourse connectors and markers), frequencies and sentiment association scores for tokens with specific POS tags, POS tags and sentiment association scores first and last two tokens of tweets, and the highest and lowest sentiment association scores within tweets. We call this group of features as *secondary feature set.*

Table 14 lists the feature sets and labels them with an id for later use. For illustration, the number of features in each subset is indicated.

6.3.1 Primary Features

To represent our features compactly, we use compound *primary features* that encode *polarity class in linguistic context* as described above paired with the *lexical resource* that supplied this score. Abstracting away from actual sentiment terms to their polarity class helps to manage the feature space dimensionality. It also smooths over the different lexical gaps of each lexicon by mapping sentiment terms to their polarity class. Primary features from a lexical resource are bundled under the name of that lexicon.

ids		# features
	Primary Feature Subsets	
f_1	AFINN-111	9
f_2	MPQA	12
f_3	BingLiu	8
f_4	NRC unigrams	17
f_5	NRC bigrams	17
f_6	Gezi unigrams	17
f_7	Gezi bigrams	17
f_8	dependency scores	13
f_9	dependency frequencies	8
	Secondary Feature Subsets	
f_{10}	POS tag based scores and frequencies	9
f_{11}	frequencies of specific annotations	12
f_{12}	position and top-lowest scores	6

Table 14: Feature subset bundles.

The linguistic context features were encoded as occurrences. The general schema of this integration for our system can be formulated as `polarityClass, lexicalResource, linguisticScope`, where `polarityClass` is one of *positive, negative, neutral, strong positive, strong negative*, `lexicalResource` represents a lexical resource and `linguisticScope` is one of *none, negation, modality, negation+modality.* For each tweet token, its prior polarity and any scope annotation is checked (a score feature is created if a lexicon provides score information for its terms).

Table 15 shows the primary features created from the AFINN-111 lexical resource for Example 10. *perfect* is the only sentiment carrier term from *aFinn* as a *positive* entry with a score of *3.* In the phrase of main verb of Example 10[4], there is a negation trigger which has the scope over the constituent of an adjectival phrase containing the only sentiment carrier. Since the sentiment carrier occurs within a negation scope, we treat it as a negated positive term. Therefore, the score of the term which is 3, is multiplied by -0.5 as we discussed in Section 2.3, and the frequency feature *positive-aFinn-negated* is incremented from 0 to 1 in Table 15.

Example 10. *El Classico on a Sunday Night is*n't perfect *for the Monday Morning !!*

[4] Notice that the negation trigger is emphasized, the tokens in its scope are underlined, and the sentiment carrier is both emphasized and underlined.

feature	value
positive-aFinn frequency	0
positive-aFinn-negated frequency	1
positive-aFinn-mod frequency	0
positive-aFinn-mod-negated frequency	0
negative-aFinn frequency	0
negative-aFinn-negated frequency	0
negative-aFinn-mod frequency	0
negative-aFinn-mod-negated frequency	0
aFinn total score	-1.5

Table 15: AFINN-111 subset features for Example 10.

6.3.2 Secondary Features

Secondary features are a collection of ad-hoc features, such as specific annotations (i.e. emoticons, implicit-explicit negation triggers, modality triggers, named-entities, contrastive discourse connectors and markers), frequencies and sentiment association scores for tokens with specific POS tags, POS tags and sentiment association scores of the first and last two tokens of tweets, and the highest and lowest sentiment association scores within tweets.

In their detailed ablation study, (Kiritchenko et al., 2014) uses these type of ad-hoc features and show them they are useful and beneficial for sentiment analysis of tweets. They make use of terms with highest and lowest sentiment scores within a tweet, last token's score, POS tag related features. We extend these features and represent them in our feature space as:

- POS counts and scores by type
- first and last two tokens' POS type and sentiment score
- highest and lowest sentiment scores within a tweet
- emoticons
- negation/modality triggers
- contrastive discourse markers (e.g. *although*)
- selected idioms (e.g. *can't wait*)
- named entities

In order to outline how the features are encoded, selected features of the tweet in Example 11 from primary and secondary feature sets with their values and their source attributes have been represented in Table 17. The table takes its lexical information from Table 16.

Example 11. *Working for 5 hours tonight with* ***no*** help ***should*** *be* fun *and* joyful #NOT

In Example 11, negation and modality triggers, *no* and *should*, are in bold; negation scope is in double underline; modality scope is in one underline and some selected sentiment-laden triggers, listed in Table 16, occurred in the example which are emphasized.

In Table 16, the sentiment triggers' final sentiment association scores are calculated according to the context they are in. Recall that for negation scope initial association score is multiplied with -0.5 and this is 0.5 for modality scope as explained in Section 2.3.

trigger	resource	category	in context of	association score	final score
help	AFINN-111	positive	negation	2	2 *-0.5 = -1
fun	AFINN-111	positive	modality	4	4 * 0.5 = 2
joyful	AFINN-111	positive	modality	3	3 * 0.5 = 1.5
			...		
#NOT	Gezi	negative	-	-1.88	-1.88
			...		

Table 16: Some sentiment triggers' representation for Example 11.

In Table 17, selected linguistic scope motivated lexical sentiment features, primary feature set, are represented in the first part as they are represented in the feature space. These features are frequency and score-based ones as explained in detail for Example 10.

In the second part of the table, the secondary feature set is introduced with a few selected features. Here basic structural features (as last token's final sentiment association score and POS tag) and frequency-based features (as contrastive discourse marker and modality trigger frequencies) are listed.

6.4 Feature Combinations

In their two-phase learning technique, (Shareghi and Bergler, 2013a) create every single feature combination's training model. Figure 9 describes their method. This is the first phase of the two-phase exhaustive feature combination technique of (Shareghi and Bergler, 2013a,b). We use this

term	score	feature subset	feature	value
		AFINN-111	positive-aFinn freq.	0
help		AFINN-111	positive-aFinn-negated freq.	1
fun,joyful		AFINN-111	positive-aFinn-mod freq.	2
		AFINN-111	positive-aFinn-modNegated freq.	0
		AFINN-111	negative-aFinn freq.	0
		AFINN-111	negative-aFinn-negated freq.	0
		AFINN-111	negative-aFinn-mod freq.	0
		AFINN-111	negative-aFinn-modNegated freq.	0
$help+(fun+joyful)$	-1+3.5	AFINN-111	aFinn total score	2.5
		. . .		
$\#NOT$		Gezi unigrams	negative-GeziUnigrams freq.	1
		. . .		
$\#NOT$	-1.88	Gezi unigrams	GeziUnigrams total score	-1.88
		. . .		
$\#NOT$	-1.88	pos. and top-low sc.	last token's score	-1.88
$\#NOT$		POS tag sc. and freq.	last token's POS	HT
		. . .		
		freq.s of specific ann	contra. disc. freq.	0
should		freq.s of specific ann	modal trigger freq.	1

Table 17: Some sample features created for Example 11.

methodology in order to create every combination of the feature subsets described in Table 14.

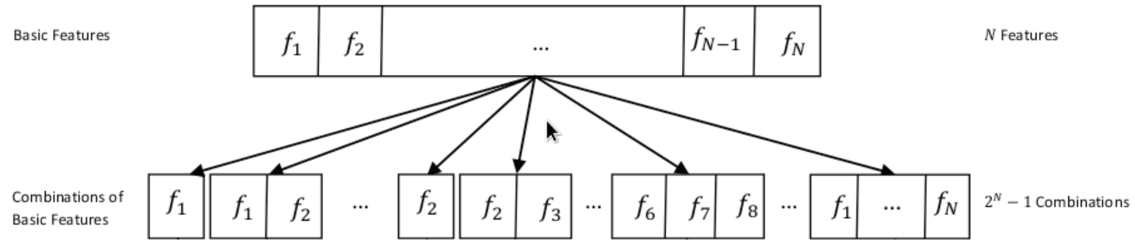

Figure 9: First phase of (Shareghi and Bergler, 2013a)'s two-phase learning technique.

6.5 Supervised Learning

For Task 10B of SemEval 2015 (addressed in Section 7.4), we process each feature subset combination with libSVM of (Chang and Lin, 2011) with RBF kernel[5] and parameters of cost=5, gamma=0.001

[5] (Chang and Lin, 2011) states that RBF kernel is the most reasonable kernel to use since it is also able to handle non-linearity relation between the gold labels and the attributes of the instances. Therefore, we make use of this kernel.

and weights=[neutral=1; positive=2; negative=2.9], and M5P by (Wang and Witten, 1997), a decision tree regressor, to predict continuous values[6] for Task 11 in Weka tool (Witten and Frank, 2011).

We make use of the Weka API to create classes for applying supervised learning. Since Task 10B, tweet-level sentiment polarity classification, is a classification task, we use the off-the-shelf classifiers of Naive Bayes and Support Vector Machines available in Weka. On the other hand, Task 11, measuring the degree of sentiment of tweets in figurative language, is a task that both regression and classification can be applied given that both discrete and continuous gold labels exist. However, the scale of -5 to 5 contains 11 discrete ordinal labels that makes the classification harder due to its high number of classes. Therefore, we select regression to train our model with a decision tree regressor.

[6] http://www.opentox.org/dev/documentation/components/m5p

Chapter 7

Experiments

In this chapter, our experiments are described: baseline experiments, system development experiments, and official experiment on the SemEval challenges.

7.1 Baseline Study

For message polarity classification, baseline experiments are applied to evaluate a basic approach for sentiment analysis of tweets in comparison with gold standard data from SemEval 2013 Twitter Sentiment Analysis Task (Task 10B of 2015). The basic approach is to classify the tweets based on a term count method where terms come from a basis sentiment lexicon, namely MPQA. The sentiment clue terms are labelled and the tweet is annotated as the class of which mostly occurs as a rule-based prediction and in the case of a tie it is annotated as neutral; as applied in (Kennedy and Inkpen, 2006). We extended this rule-based baseline approach also to analyze it with supervision techniques. Count-based features are encoded per sentiment class in order off-the-shelf classifiers to predict sentiment as machine learning baselines.

Results of Rule-based Baseline Evaluation This baseline is simply voting for a sentiment label based on polarity term counting. Here, development and training sets of the data are combined, since there is no training. Table 18 shows evaluation results of the rule-based baseline.

Macro in Table 18 is the average of F-measures (*av. F*) of all three classes. In Task 10B, however, the evaluation was made only using macro F-measure averages of only *positive* and *negative* classes taken into consideration as final results, which can be seen at the last row of the table in italics. This is expressed in Section 7.4 in detail.

Annotation	Precision	Recall	av. F
negative	0.36	0.31	0.33
neutral	0.60	0.67	0.63
positive	0.57	0.52	0.54
Macro	0.51	0.50	0.50
Macro (pos+neg)	0.47	0.42	*0.44*

Table 18: Rule-based baseline results.

Classifiers' Results After having the datasets annotated with the MPQA Subjectivity Lexicon, three features of counts of term's polarity classes (positives, negatives, and neutral) of each tweet was written to an Weka-specific file, in order to classify and evaluate the development data. Naive Bayes and SVM classifiers with default parameters are used for these experiments.

The official evaluation metric of Task 10B is average F-measure of positive and negative classes. Since neutrals are the most frequent labels in the datasets, the evaluation metric is optimized for less frequent labels.

Table 19 shows results from Naive-Bayes baseline. Next, Table 20 and Table 21 show the results of non-weighted and weighted[1] SVM classifiers respectively. The reason for using weights on gold polarity classes is as follows: When it comes to giving different penalties on failing different classes, SVM classifiers can be configured to apply the given weights to the classes. In this case, distribution of classes are not balanced in the datasets and since we evaluate average F-measures of positive and negative classes, applying such weights improves the results for the evaluated less frequent classes, especially the negative class.

As a result of baseline studies, the only supervised baseline that could beat the rule-based baseline is weighted SVM. The reason behind this result is that Naive Bayes and non-weighted SVM prefer predicting the most frequent class, *neutral* and recalls of neutral increases where the others' decrease. This shows the importance of using weights: When the data is unbalanced and the most frequent class takes a little part in the evaluation metric, classifiers should be configured. This fact also

[1] weights=[neutral=1; positive=2; negative=2.9]

Class	Precision	Recall	av. F
negative	0.57	0.19	0.28
neutral	0.51	0.84	0.63
positive	0.56	0.31	0.40
Macro	0.55	0.45	0.44
Macro (pos+neg)	0.56	0.25	*0.34*

Table 19: Results of Naive Bayes Classifier on the Development Set.

Class	Precision	Recall	av. F
negative	0.57	0.16	0.25
neutral	0.58	0.74	0.65
positive	0.55	0.57	0.56
Macro	0.57	0.49	0.49
Macro (pos+neg)	0.56	0.37	*0.41*

Table 20: Results of SVM Classifier (without weights) on the Development Set.

concludes why supervision could not outperform a rule-based baseline.

In SemEval 2013 challenge, baseline was majority class voting. Positive class is selected as majority class, although neutrals were major, given that evaluation metric does not include them. Table 22 compares all these baselines, with their ranking if those baselines were submitted to the challenge.

7.2 Experiments on Automatically-created Lexica

In order to assess how the Gezi Lexicon and NRC Hashtag Sentiment Lexicon perform relatively, we run experiments of how they intersect and agree with manually-curated lexica and we also run experiments with them on SemEval challenge data.

Class	Precision	Recall	av. F
negative	0.38	0.46	0.42
neutral	0.61	0.61	0.61
positive	0.57	0.51	0.54
Macro	0.52	0.53	0.52
Macro (pos+neg)	0.48	0.49	*0.48*

Table 21: Results of SVM Classifier (with weights) on the Development Set.

Baseline Type	av. F	Ranking
weighted SVM	0.48	30/38
rule-based	0.44	34/38
non-weighted SVM	0.41	35/38
Naive Bayes	0.34	37/38
majority (official)	0.29	37/38

Table 22: Comparison of baselines to official baselines.

7.2.1 Term Intersection and Agreement between Lexica

To assess the relationship of size to unique content, we compared the five lexica and determined the size of the *Intersections* of the terms covered, indicating separately in how many cases the assigned sentiment prior polarity is the same (*Agreement*). Table 23 shows the intersection and the agreement level of polarity labels for pairs of lexica separately according to the creation types of the pairs.

	Lex A	Lex B	$A \cap B$	Agreement	$\frac{Agreement}{A \cap B}$	A/B	B/A
manual-manual	MPQA	Liu	5,414	5,369	0.992	1,472	1,372
	AFINN-111	Liu	1,314	1,298	0.988	1,162	5,472
	MPQA	AFINN-111	1,246	1,202	0.965	5,640	1,230
manual-auto	AFINN-111	Gezi	1,911	1,624	0.850	565	218,488
	AFINN-111	NRC	989	822	0.831	1,487	24,732
	Liu	Gezi	4,028	3,386	0.841	2,758	216,371
	Liu	NRC	1,840	1,488	0.809	4,946	23,881
	MPQA	NRC	1,819	1,340	0.737	5,067	23,902
	MPQA	Gezi	4,105	2,993	0.729	2,781	216,294
auto-auto	NRC	Gezi	16,868	13,957	0.827	8,853	203,531

Table 23: Intersection and agreement between sentiment lexica.

The highest agreement rate is between manually created lexica. The lexicon pair with the highest agreement rate is MPQA-Liu. This is expected since MPQA formed the seed for Liu. The lowest rate of agreements are between MPQA and automatically created lexica since they have a lack of the neutral category in our usage, Liu has a higher agreement ratio with automatically created lexica since it also does not have neutral labelled entries.

Gezi agrees with and covers manually-created lexica more than the NRC Hashtag Lexicon does, shows that when using negation context, having more entries and applying preprocessing can improve the quality of automatically-created lexica.

Another observation here is that larger lexica do not necessarily intersect and agree better proportionally. Terms contained by small and manually-created lexica are core and frequent ones in general; whereas automatically created lexica possess also relatively low-frequency terms that may be domain-dependent and/or volatile.

The disagreements of the terms' polarity classes between different lexica may be due to sentiment-laden terms that may have different sentiment polarity values depending on contexts. This is observed in (Andreevskaia and Bergler, 2006), described as sentiment-laden terms form a fuzzy set.

7.2.2 Comparing Gezi and NRC Lexica Feature Subsets

Table 24 illustrates results of feature subset combinations of NRC and Gezi resources. Each combination for the subsets of unigrams and bigrams per lexicon has been shown in the table on 2 years' tweet test sets. Also, two runs with Gezi dependency based features with NRC and/or Gezi show that they contribute to some degree. It is noteworthy that when we use both lexica's the feature subsets, the combination works better than the combination of each lexicon with dependency features.

feature subsets	Test 2014 av. F	Test 2013 av. F
NRC&Gezi unigrams, bigrams	64.26	59.60
Gezi unigrams, bigrams	63.79	58.03
Gezi unigrams, bigrams, deps	63.38	59.13
Gezi unigrams, NRC bigrams	63.21	58.63
Gezi unigrams	60.81	57.86
NRC unigrams, bigrams, Gezi deps	59.44	56.94
NRC unigrams, Gezi bigrams	58.05	55.36
Gezi bigrams	56.40	50.45
NRC unigrams, bigrams	55.96	55.09
NRC bigrams	53.48	52.31
NRC unigrams	52.39	50.90

Table 24: Different feature subset combinations of NRC and Gezi.

It is also noteworthy that dependencies improve NRC's results more than it improves Gezi. Since Gezi unigrams and bigrams are extracted from the same collection that the dependencies are extracted, they may represent similar aspects as features and it may go well with another resource which is compiled in a different time frame and a different collection.

7.3 System Development Experiments

After studying the baseline, experiments have been conducted to examine the features with varying settings with lexical and linguistic phenomena. These variations include running the pipeline with different tools and resources (tokenizers, POS taggers etc.), usage of the Stanford parser, encoding features by scaling and non-scaling, encoding lexical features for automatically compiled lexica with traditional 3 polarity classes and 5 finer grained polarity classes and finally the effect of negation and modality on results by encoding features with or without them.

Test Sets and Feature Combinations Experiments are generated on test sets of 2013's and 2014's Sentiment Analysis of Tweet challenge by training the challenge's only training set, given the fact that the test sets were released by the time we started these experiments. The features used in these experiments are the ones described in Chapter 6: primary feature set (negation and modality context-aware lexical features) and secondary feature set (ad-hoc features of several annotations and features makes use of structure of tweets). Their feature subsets are present in Table 25, illustrated in Chapter 6 and revisited here. Due to the fact that we experiment on every single combination of 12 feature subsets, $2^{12} - 1 = 2047$ combinations, a constrained number of results will be pointed out by using their ids.

ids		# features
	Primary Feature Subsets	
f_1	AFINN-111	9
f_2	MPQA	12
f_3	BingLiu	8
f_4	NRC unigrams	17
f_5	NRC bigrams	17
f_6	Gezi unigrams	17
f_7	Gezi bigrams	17
f_8	dependency scores	13
f_9	dependency frequencies	8
	Secondary Feature Subsets	
f_{10}	POS tag based scores and frequencies	9
f_{11}	frequencies of specific annotations	12
f_{12}	position and top-lowest scores	6

Table 25: Feature subset bundles revisited.

Although those limited results will be listed for each experiment, the effect of the components will be tested for testing statistical significance with inclusion of every combination to the tests in order not to conclude by chance for the experiments.

Statistical Significance To observe whether any component brings improvement to the system, significance tests are conducted to the (possibly pairs of) result lists before and after applying the component. *Hypothesis Testing* (Manning and Schütze, 1999) is used for discovering change or difference occurred by chance or not. In this test, a hypothesis, called the *null hypothesis* (H_0), is formulated in a way that difference occurs by chance. This test may make the null hypothesis come true since looking only a small numbers of samples may give bias. In the light of this hypothesis, the probability of the difference, *p-value*, and measure of the size of the difference relative to the variance of the samples, *t-value*, are calculated; if probability of the difference, p-value, is lower than the predefined significance level, α, the null hypothesis H_0 can be rejected. In addition, t-value supplies information whether this difference occurs as an improvement or the opposite. Generally, significance level is generally defined as 0.05 or 0.01 in literature.

In this study, our samples are feature subset combinations' result pairs, coming from results before applying the component and after. Therefore, we have 2047 samples, *N*, and *degree of freedom, df = N-2 = 2045*. For each experiment, our hypothesis will be that a component improves the results $(H_1 : \mu > 0)$, thus our null hypothesis should state reverse, which there is no improvement: $(H_0 : \mu \leq 0)$. In consideration of this hypothesis and the structure of our paired samples, one-sided and paired *Student's t-test* (Press et al., 1988) is applied for the development experiments to see if the changes are statistically significant. We use a significance level of 0.01 in our significance tests.

Paired t-test requires sample data to be in intervals and normally-distributed (Moore and McCabe, 1989). Our sample data fulfils the former as it is in intervals. For testing normality, Shapiro-Wilk test is used as it is reported to be the best choice (Thode, 2002), (Ghasemi and Zahediasl, 2012). An off-the-shelf implementation exist in the R programming language for Shapiro-Wilk test[2]. We fulfilled the latter requirement of normality via this tool for each experiment. If the data samples for any experiment weren't be normally distributed, we would have to use Wilcoxon signed-rank test[3] (Woolson, 2008).

[2] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/shapiro.test.html

[3] This is a non-parametric statistical hypothesis test.

For each experiment described in this section, the experiment definition, constrained results and statistical significance test interpretations and decisions will be explained.

7.3.1 Experiments on Feature Scaling

Scaling features for machine learning experiments is reported to be advantageous by avoiding dominance of large ranges of some attributes unequally over smaller ones. In their extensive user guide for SVM applications, (Hsu et al., 2003) explains the importance in a similar way and points out sample experiments to illustrate how scaling matters. In sentiment analysis domain, a general overview by (Pang and Lee, 2008) also concludes the same way that scaling features bag-of-words approaches gives improvements. Given that our system does not have a bag-of-words approach, scaling is tested on the system's features by scaling frequency-based features into a range of [-1,1]. This experiment was conducted with the ANNIE tokenizer and the weighted SVM classifier and its results are available in Table 26 over the 2013 and 2014's tweet test sets.

	Scaled Features		Non-scaled Features	
ids	Test2014 av. F	Test2013 av. F	Test2014 av. F	Test2013 av. F
$f_{1,2,3,6,7,10,11,12}$	67.88	65.12	67.98	65.23
$f_{3,6,7,8,9,10,11,12}$	66.45	65.15	66.30	64.73
$f_{1,3,4,5,7,8,9,10,11,12}$	68.63	67.60	69.05	67.94
$f_{1,3,6,7,8,9,10,11,12}$	68.68	67.16	68.92	67.36
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	68.74	67.18	69.12	67.63
$f_{1,2,3,6,7,8,9,10,11,12}$	68.67	67.57	69.18	67.14
$f_{1,3,5,6,7,8,9,10,11,12}$	68.11	67.01	68.34	67.44
$f_{1,5}$	64.31	61.53	64.11	61.57
$f_{1,6,7,8,9,10,11,12}$	67.95	66.83	68.78	67.07
$f_{1,4}$	63.20	60.49	63.25	60.18

Table 26: Constrained results of scaled and non-scaled features.

In statistical significance test on the results of the experiment, the null hypothesis that scaling improves the results could not be rejected for both 2013 and 2014 test sets. For 2013 test set, p-value is calculated as 0.14 which is not lower than significance level of 0.01, meaning that the null hypothesis cannot be rejected and significance does not exist and t-value is 1.47. For 2014 test set, p-value is 0.20 and t-value is -1.29, where again no significance exists and confidence level, t-value, is low relatively. Therefore, we select to continue to our experiments without scaling the feature space.

7.3.2 Experiments with Different Tokenizers

In this experiment, the hybrid tokenization module, explained in Chapter 6, is tested against CMU and ANNIE tokenizer. Table 27 shows the results of different tokenizer runs on the selected combinations. Given that weighted SVM classifier outperformed other classifiers, only it is used for this experiment.

	CMU		ANNIE		Hybrid module	
ids	Test2014	Test2013	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	64.31	68.74	65.91	68.78	66.64	68.76
$f_{1,2,3,6,7,10,11,12}$	63.40	66.32	64.33	68.08	66.19	69.38
$f_{3,6,7,8,9,10,11,12}$	60.72	64.36	62.02	65.16	64.46	66.66
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	63.54	68.21	64.30	68.89	67.16	68.97
$f_{1,3,4,5,7,8,9,10,11,12}$	64.84	67.96	65.04	67.98	67.75	69.06
$f_{1,3,6,7,8,9,10,11,12}$	65.32	68.43	65.13	68.58	67.23	69.53
$f_{1,2,3,6,7,8,9,10,11,12}$	63.17	67.21	65.23	69.20	67.24	69.64
$f_{1,3,5,6,7,8,9,10,11,12}$	64.76	67.86	66.20	68.57	66.99	68.42
$f_{1,6}$	61.85	63.37	62.03	64.31	62.32	65.08
$f_{1,6,7,8,9,10,11,12}$	66.32	68.54	66.27	68.83	67.12	68.85
$f_{1,4}$	60.12	63.22	61.50	63.52	61.45	63.61

Table 27: Constrained results of different tokenizers.

The same Student's t-tests are applied on the pairs of the hybrid tokenizer module and others. In these experiments, a high significant difference of hybrid tokenizer module is observed: Between the samples of ANNIE and hybrid tokenizer, p-value is calculated as $0.44 * 10^{-19}$ for 2013 test set which is extremely low, meaning it is significant and t-value is 9.37. From this point on such extremely low p-values will be denoted as $p < 0.001$ in order to follow APA guidelines. For 2014 test set, p-value is $p < 0.001$ again and t-value is 7.08, where again high significance exists. For the samples between CMU and hybrid tokenizers the probability of null hypothesis is even lower.

Since we observed that the hybrid tokenizer module is highly significant, the experiments are to be continued with only tokenizing with this module.

7.3.3 Experiments with Different Classifiers

For the baselines, weighted SVM classifier outperformed non-weighted SVM and Naive Bayes classifiers. We also tested these three classifiers on every feature subset combination. Table 28 shows the results of these classifier runs on the selected combinations and the table reflects that weighted SVM outperforms other classifiers.

	Naive Bayes		non-weighted SVM		weighted SVM	
ids	Test2014	Test2013	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	62.02	64.32	63.52	64.60	66.64	68.76
$f_{1,2,3,6,7,10,11,12}$	62.61	65.82	63.23	64.54	66.19	69.38
$f_{3,6,7,8,9,10,11,12}$	59.20	59.12	58.73	60.89	64.46	66.66
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	60.92	63.48	62.83	64.68	67.16	68.97
$f_{1,3,4,5,7,8,9,10,11,12}$	62.98	64.89	63.06	64.77	67.75	69.06
$f_{1,3,6,7,8,9,10,11,12}$	61.56	64.12	62.84	63.73	67.23	69.53
$f_{1,2,3,6,7,8,9,10,11,12}$	60.57	63.96	63.42	64.60	67.24	69.64
$f_{1,3,5,6,7,8,9,10,11,12}$	62.34	63.81	62.70	64.26	66.99	68.42
$f_{1,6}$	61.88	63.32	60.74	61.34	62.32	65.08
$f_{1,6,7,8,9,10,11,12}$	61.05	62.08	63.08	64.31	67.12	68.85
$f_{1,4}$	58.05	58.54	58.98	60.02	61.45	63.61

Table 28: Constrained results of different classifiers.

In order to see if these results are significant, one-sided Student's t-tests are conducted on combination pairs of weighted SVM/non-weighted SVM and weighted SVM/Naive Bayes. Null hypothesis of weighted SVM's difference is by chance is rejected for both tests by t-tests with extremely low p-values ($p < 0.001$ for each test) and high t-values. T-values for sample pairs of weighted SVM/non-weighted SVM are 36.95 for 2013 test set and 33.24 for 2014's. For sample pairs of weighted SVM/Naive Bayes these values are 53.83 and 56.95 respectively.

Given that weighted SVM is highly significant, the experiments are to be continued with only weighted SVM from this point on.

7.3.4 Experiments on Different Parsing Modules

Creating accurate syntax parse trees and dependency graphs gives rise to have better scope of negation and modality, and this also does creating more accurate features. The GATE framework has a built-in Stanford parser plugins to create syntax parse and dependency tree annotations with its default settings which could be changed to improve results: Using POS tags created by the parser and a probabilistic context-free grammar. We send POS tags that are existing in our annotations thanks to our pipeline and to use a caseless probabilistic context-free grammar which is trained on informal texts (Socher et al., 2013a).

To give our pipeline's CMU POS tags, the obstacle of non Penn Treebank style POS tags (*HT, URL, USR, RT*) should be overcome since the parser accepts Penn Treebank style FPOS tags. Given that Stanford parser accepts partial POS tags, we let the parser predict POS tags of tokens with non Penn Treebank style POS tags by not sending their POS tags.

In this experiment, we tested these settings of partial POS tags and caseless PCFG grammar against default settings[4]. Table 29 shows the results of these parser modules' runs on the selected combinations.

	Parser with default settings		Parser with new settings	
ids	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	67.02	69.17	68.24	69.67
$f_{1,2,3,6,7,10,11,12}$	66.19	69.38	67.29	68.57
$f_{3,6,7,8,9,10,11,12}$	63.81	66.78	64.55	65.85
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	67.16	68.97	67.22	68.74
$f_{1,3,4,5,7,8,9,10,11,12}$	67.75	69.06	67.65	69.35
$f_{1,3,6,7,8,9,10,11,12}$	67.23	69.53	66.67	68.73
$f_{1,2,3,6,7,8,9,10,11,12}$	67.24	69.64	67.15	69.76
$f_{1,3,5,6,7,8,9,10,11,12}$	66.84	69.00	66.84	68.37
$f_{1,6}$	62.32	65.08	65.67	67.58
$f_{1,6,7,8,9,10,11,12}$	67.12	68.85	67.15	69.53
$f_{1,4}$	61.45	63.61	62.04	63.36

Table 29: Constrained results of parsing module with different settings.

For this experiment, the null hypothesis of improvements of the settings we used for the parser could only be rejected in 2014 test set with an extremely low p-value ($p < 0.001$) and a t-value of 7.16. For 2013 test set, p-value is 0.03 and t-value 2.15; where p-value is slightly greater than our significance level of 0.01. If the traditional significance level were defined as 0.05; the improvements would be significant.

In the experiments after this one, the settings of partial POS tags and caseless PCFG[5] grammar are used for the parser module, because significance was detected in one test set and was missed

[4] These settings are a general PCFG grammar and parsing the sentences by using parser's POS tagger itself

[5] probabilistic context-free grammar

slightly in the other.

7.3.5 Experiments with Different Prior Polarity Classes

As stated in Chapter 4, prior polarities can be assigned to the terms of automatically compiled lexica since their entries have only association scores initially. In our study, this is done by defining thresholds as described. The question that needs to be answered in this experiment is whether assigning traditional prior polarity classes (positive, negative, and neutral) or more fine-grained prior polarity classes (adding strong-positive and strong-negative to traditional classes) is beneficial. The question arose scaling adjectives, as <*strong negative, negative, neutral, positive, strong positive*>, to see if polar terms have a difference with its strong form (i.e. *good* as positive, *excellent* as strong positive). Also, it is worth to check the behaviour of strong polarity classes under scope of negation and modality.

After the mention of scaling sentiment-laden terms according to their polarity and their usage with linguistic notions, some important studies come to mind: (Horn, 1984) brings *Scalar Implicature*[6] into terminology while expanding Grice's *Maxim of Quantity*[7] *(Informativeness)* (Grice, 1975). In Horn's theory of implicature, writers or speakers follow Maxim of Quantity and give sufficient and necessary information. When it comes to making a statement of a scale in terms of quantity, the statement implicates the negated higher scaled statement as in Example 12 in the scale of <*none,some,most,all*>

Example 12. Most *of the students came to the class.*
Implicature: *Not all* of the students came to the class.

The reason pointing out Horn's theory here is not to infer implication statements from adjective scales. Notice that for different scales implicatures can be different in negated statements. If one tries to see whether anything is implied by negated statements, it can be seen that nothing certain can be implied (Carston, 1998): Not a certain implication in Example 13 can be inferred:

Example 13. Not most *of the students came to the class.*

[6] An implicit meaning is inferred from a statement which has a reason not supplying a more informative term on the same scale with the term supplied

[7] *A writer or speaker should be informative as much as needed.*

Here in Example 13, the reason that nothing certain could be implied comes from the nature of negation. It states that one term in the scale is not the case, and others could be possible to be true, but not certain. Looking from the perspective of negation, adjective scaling seems interesting: When an adjective is used in negation scope coming from a scale of <*terrible, bad, okay, good, excellent*>, it may be inferred not only cancelling the negated adjective but others are to be possible to be cancelled as well:

Example 14. *The film was* not good.

It, most probably, does not only cancel the quality of the film as being good here, but it also cancels being excellent as well. Even being okay may have a chance as an implication. However, in the case of negating a strong polar adjective, the situation seems to become different:

Example 15. *The film was* not excellent.

The meaning in Example 15 by negating a strong positive adjective, may imply any other adjective, including *good*[8], in the scale depending on content it has, but apparently it has a different semantics than a non-strong positive adjective. The reasoning here is that given the information should be informative, sufficient, and necessary, following Horn's implicature theory; the necessity level of the quantity should be given. Here in Example 15, the stronger term (*excellent*) as opposed to the less strong one (*good*) is necessary. In addition, the idea of directly reversing the polarity class seems not to work here as something not strong positive may be positive or neutral. Therefore, it seems to be necessary to create finer-grained prior polarity classes by including strong forms of polarities, especially for usage of linguistic notion of negation, or even for modality, when it is possible.

Consequently, the ideas above directed us to conduct an experiment by running our pipelines separately with automatically-compiled lexica for traditional 3 prior polarity classes and finer-grained 5-classes. Table 30 demonstrates the results of these separate runs on the selected combinations.

The null hypothesis that fine-grained classes does not improve results is rejected for both tweet test sets of 2013 and 2014 with extremely low p-values ($p < 0.001$) and t-values of 4.76 for 2013 test set and 8.76 for 2014's. Therefore, we see that leveraging fine-grained prior polarity classes promises

[8] The quality of the film could be thought a high-level one, but not as much as an excellent high-level.

ids	Traditional classes Test2014 av. F	 Test2013 av. F	Fine-grained classes Test2014 av. F	 Test2013 av. F
$f_{1,2,3,6,8,9,10,11,12}$	66.96	69.25	69.95	70.31
$f_{1,2,3,6,7,10,11,12}$	67.29	68.57	69.44	70.78
$f_{3,6,7,8,9,10,11,12}$	64.53	65.40	67.80	68.37
$f_{1,3,4,5,7,8,9,10,11,12}$	66.78	69.02	68.57	70.25
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	67.22	68.74	69.98	70.81
$f_{1,3,6,7,8,9,10,11,12}$	66.79	69.04	69.12	70.31
$f_{1,2,3,6,7,8,9,10,11,12}$	67.15	69.76	69.63	70.44
$f_{1,3,5,6,7,8,9,10,11,12}$	67.09	68.25	69.57	70.61
$f_{1,6}$	61.55	62.81	63.60	64.89
$f_{1,6,7,8,9,10,11,12}$	67.15	69.53	68.63	70.06
$f_{1,4}$	61.83	62.65	63.01	64.15

Table 30: Constrained results of automatically-compiled lexica 3 vs. 5 classes discrimination.

to improve results significantly. For the experiments after this one, fine-grained prior polarity classes for automatically compiled lexica with linguistic context are used.

7.3.6 Experiments on Linguistic Contexts

In the experiment of linguistic contexts of negation and modality as mentioned in Section 2.3, we tested using context of negation and modality by incorporating them with lexical features. Thus, we run our pipeline to create feature spaces separately by giving no context information, only negation context, only modality context, and negation and modality contexts together. Then, after having results of four different runs, statistical significance tests are conducted with three different negation and modality treatment experiment results against the experiment without scope information.

Table 31 gives results of experiment pairs of no linguistic treatment versus negation and modality treatment together on tweet test sets of 2013 and 2014. Null hypothesis of negation and modality treatment together does not improve results is rejected for both datasets with extreme low p-values ($p < 0.001$) and t-values of 22.49 for 2014's test set and 15.39 for 2013. As a result, linguistic treatment leveraged with lexical features matters for sentiment analysis of tweets in these test sets.

Results of experiment pairs of no linguistic treatment versus only negation treatment on tweet test sets of 2013 and 2014 are illustrated in Table 32. The null hypothesis that negation treatment does not improve results is rejected again for both datasets with extreme low p-values ($p < 0.001$) and t-values of 20.75 for 2014's test set and 13.55 for 2013. Therefore, negation treatment does

ids	No Scope		Negation & Modality	
	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	67.23	68.85	69.95	70.31
$f_{1,2,3,6,7,10,11,12}$	65.48	66.17	69.44	70.78
$f_{3,6,7,8,9,10,11,12}$	65.44	66.52	67.80	68.37
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	67.05	68.93	69.98	70.81
$f_{1,3,6,7,8,9,10,11,12}$	66.01	68.42	69.12	70.31
$f_{1,3,4,5,7,8,9,10,11,12}$	66.35	68.45	68.57	70.25
$f_{1,2,3,6,7,8,9,10,11,12}$	67.58	69.23	69.63	70.44
$f_{1,3,5,6,7,8,9,10,11,12}$	67.28	69.31	69.57	70.61
$f_{1,6}$	60.62	61.99	63.60	64.89
$f_{1,6,7,8,9,10,11,12}$	67.98	68.56	68.63	70.06
$f_{1,4}$	59.16	61.33	63.01	64.15

Table 31: Constrained results of negation and modality scope compared to no scope.

ids	No Scope		Only Negation	
	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	67.23	68.85	68.99	70.24
$f_{1,2,3,6,7,10,11,12}$	65.48	66.17	68.79	70.28
$f_{3,6,7,8,9,10,11,12}$	65.44	66.52	67.87	67.29
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	67.05	68.93	69.13	70.24
$f_{1,3,6,7,8,9,10,11,12}$	66.01	68.42	69.25	69.40
$f_{1,3,4,5,7,8,9,10,11,12}$	66.35	68.45	69.11	69.57
$f_{1,2,3,6,7,8,9,10,11,12}$	67.58	69.23	69.39	70.36
$f_{1,3,5,6,7,8,9,10,11,12}$	67.28	69.31	69.39	70.50
$f_{1,6}$	60.62	61.99	63.88	64.42
$f_{1,6,7,8,9,10,11,12}$	67.98	68.56	69.17	69.30
$f_{1,4}$	59.16	61.33	63.12	64.07

Table 32: Constrained results of only negation scope compared to no scope.

significantly improve results.

Results of experiment pairs of no treatment versus only modality treatment on tweet test sets can be seen in Table 33. Null hypothesis of modality treatment does not improve results could not be rejected here for both datasets with p-values of ($p = 0.071$) for 2014 test set, ($p = 0.088$) for 2013 test set and t-values of 0.38 for 2014's test set and 0.16 for 2013. This means modality treatment slightly improves but this can be by chance. This may be due to the fact that we use modality by over-generalizing its sub-branches. However, these results do not mean that modality is useless. Using modality with negation produces higher t-values than using only negation.

	No Scope		Only Modality	
ids	Test2014	Test2013	Test2014	Test2013
	av. F	av. F	av. F	av. F
$f_{1,2,3,6,8,9,10,11,12}$	67.23	68.85	68.21	68.89
$f_{1,2,3,6,7,10,11,12}$	65.48	66.17	64.47	66.74
$f_{3,6,7,8,9,10,11,12}$	65.44	66.52	65.99	66.33
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	67.05	68.93	67.62	68.78
$f_{1,3,6,7,8,9,10,11,12}$	66.01	68.42	67.63	68.55
$f_{1,3,4,5,7,8,9,10,11,12}$	66.35	68.45	67.00	68.47
$f_{1,2,3,6,7,8,9,10,11,12}$	67.58	69.23	67.83	69.01
$f_{1,3,5,6,7,8,9,10,11,12}$	67.28	69.31	68.04	69.03
$f_{1,6}$	60.62	61.99	61.80	62.17
$f_{1,6,7,8,9,10,11,12}$	67.98	68.56	68.08	68.38
$f_{1,4}$	59.16	61.33	60.62	61.50

Table 33: Constrained results of only modality scope compared to no scope.

7.4 SemEval Tasks

In this section, descriptions of the tasks and systems that participated in 2013's and 2014's SemEval Sentiment Analysis of Tweets challenge are pointed out firstly. Then, our participation is analysed.

7.4.1 Task Descriptions

Task 10B, Polarity Classification of Tweets Polarity classification of tweets at SemEval is a ternary classification of polarity classes. The problem is simply labelling a tweet as *positive*, *neutral*, or *negative*; pointed out in detail in (Rosenthal et al., 2015).

Task 11, Sentiment Degree Association for Figurative Language Tweets Sentiment Analysis of Figurative Language in Twitter was a pilot task at SemEval 2015 (Ghosh et al., 2015). The task is to determine the degree of sentiment in a fine-grained level set of tweets containing rich figurative language, *Sarcasm, Irony* and *Metaphor* and labelled with scores between -5 and 5 by human annotators.

7.4.2 Systems on Previous SemEval Sentiment Analysis of Tweets Challenges

(Nakov et al., 2013) and (Rosenthal et al., 2014) give details and analyze results of previous Sentiment Analysis of Twitter challenges of 2013 and 2014. Some selected systems in those challenges are investigated below.

(Mohammad et al., 2013) Team NRC-Canada, takes a bag-of-words approach. They make use of word and character n-grams, several lexica, negation and twitter-specific surface-level syntax features with an SVM classifier for message polarity classification task of 2013 challenge. They contribute two new automatically compiled lexica. The usage of the scope of negation in their system is the simple heuristic of taking every term between a negation trigger and following punctuation. Their system ranked first among 38 teams with an average of positive and negative F-measure of 69.02% outperforming the following team nearly 4%. For 2014's task, the same system (Zhu et al., 2014) is submitted with the improvements of negation over automatically compiled lexica. They improved their results on 2013's tweet dataset to F-measure of 70.75% by ranking 2nd on this dataset, and their system performs 69.85% ranking 4th; for surprise sets of the challenges, which are SMS, LiveJournal entries and sarcastic tweets, (Zhu et al., 2014) ranks first for each; for overall they ranked first among 50 teams, for both macro and micro averaged F-measures of all datasets. (Kiritchenko et al., 2014) gives a detailed explanations of the systems after giving a general overview of the literature on sentiment analysis of short informal texts.

(Günther and Furrer, 2013) trains stochastic gradient descent as a linear classifier with the features created from SentiWordNet (Esuli and Sebastiani, 2006) and words that are stemmed with the Porter stemmer (Porter, 1997) and clustered with Brown cluster[9] (Owoputi et al., 2013). Word clusters are used as bag of cluster of words in feature space. Negation scope is found as in (Mohammad et al., 2013) by marking words with a negation mark if they occur between a negation trigger and a punctuation. Their result for tweet-level polarity classification is reported an F-measure of 65.27% ranking as the second team. A more wide coverage system details and iterative experiments

[9] Clusters are sets of similar words, are used to map those terms into one in order to reduce the size of feature space.

for the system is pointed out in first author's thesis (Günther, 2013). (Günther et al., 2014) improved their system for 2014's task by using a Twitter-specific tokenizer, CMU tokenizer (Gimpel et al., 2011), adding new features and sentiment lexica, namely opinion lexicon by Bing Liu (Hu and Liu, 2004), the MPQA subjectivity lexicon (Wilson et al., 2005), and NRC Hashtag Lexicon (Mohammad et al., 2013). They encoded bag-of-words features based on unigrams and bigrams, and added features of POS tag based, presence of a question mark, and a hashtag or URL, presence of positive or negative term's domination from each lexica. They report an almost 4% improvement over 2013's tweet dataset, ranking 5th and they achieved a 69.95% F-measure, ranking 3rd in 2014's tweet dataset and overall.

(Miura et al., 2014) participated in the 2014's challenge by training a logistic regressor with a linear model by using weights for each polarity gold label given that the tweet datasets are imbalanced. They apply preprocessing to data as unifying all URL's into one single token, unicode normalization, lowercasing and spelling correction. Then, preprocessed data is POS tagged with both the Stanford POS tagger (Toutanova et al., 2003) and the CMU POS tagger (Gimpel et al., 2011); and simple standard heuristic of labelling negation scope between negation trigger and following punctuation is applied. Word-sense disambiguation was applied on words with their POS tag information from the Stanford POS tagger. As a design decision, lexica used in the study are categorized as formal and informal lexica and processed separately: Formal lexica are MPQA Subjectivity Lexicon (Wilson et al., 2005), General Inquirer (Stone et al., 1966), SentiWordNet (Esuli and Sebastiani, 2006); and the informal ones are opinion lexicon of Bing Liu (Hu and Liu, 2004), NRC Hashtag Lexicon (Mohammad et al., 2013), AFINN-111 (Nielsen, 2011) and Sentiment140 Lexicon (Mohammad et al., 2013). For formal lexica, features are extracted and encoded with negation and word-sense disambiguation information. For positive and negative polarity classes in each lexicon, counts of matched sentiment term, total score per tweet, maximum score in a tweet, and last token score features are encoded per tweet. Word and character n-grams and Brown clusters from (Owoputi et al., 2013) are other features used in the system. Their system was the top-performer in 2013 and 2014's tweet-level polarity classification datasets by achieving 72.12% and 70.96% respectively. However, their low results in the surprise sets lowered their overall micro-averaged result to fifth rank. This fact shows that their system is over-trained for tweets and giving weights to their classifier did not work as well on surprise sets as they did on tweet sets.

(Tang et al., 2014) creates a feature set called sentiment-specific word embeddings by making use of a deep learning system trained with neural network on 10M positive and negative labelled via distant supervision of emoticons; and they replicate the feature set of NRC-Canada's 2013 participation (Mohammad et al., 2013). These feature sets are concatenated at the end and an SVM classifier is used for supervision. The system is submitted to 2014's challenge. (Tang et al., 2014) reports results separately on their concatenated feature space, sentiment-specific word embeddings and NRC-Canada's replication. Sentiment-specific word embeddings feature set achieves 68.69% F-measure on 2013 tweet test set and 66.86% on 2014's. When adding NRC-Canada's replicated feature set, which is the submitted version; results improve to 70.40% ranking third and 70.14% ranking second respectively. They report results on NRC-Canada's replicated feature set as 66.18% on 2013 tweet test set and 67.07% on 2014's. (Zhu et al., 2014) reports on their older system's performance as 69.02% and 68.88% respectively. These differences of comparative results may be due to different classifier parameters and kernels, and the different sizes of the training data two different team retrieved at different years.

7.4.3 Task 10B: Polarity Classification of Tweets

Approach We submitted our system, CLaC-SentiPipe (Özdemir and Bergler, 2015a), after the development experiments, which we selected as parameters of the hybrid tokenizer module, multi-classed automatically created lexica, the Stanford Parser supplied with partial POS tags, and using negation and modality scopes; trained with weighted SVM classifier on a non-scaled feature space, using the first phase of the exhaustive feature combination evaluation technique of (Shareghi and Bergler, 2013a,b).CLaC-SentiPipe with 62.00% ranked 9th among 40 teams, a very strong placement considering less than 3% separated our results from the top ranking one which achieved 64.84 f-measure. The submitted combination, $\mathrm{f}_{1,2,3,6,8,9,10,11,12}$, contained feature subsets of AFINN-111, MPQA, Liu, Gezi unigrams, dependencies and secondary feature set, 94 features in total.

Evaluation metric Average F-measure of positive and negative classes was used as the official evaluation metric calculated as in Equation 5 and Equation 6, where P stands for precision and R

stands for recall:

$$F_{av} = 2 * \frac{F_{pos} + F_{neg}}{2} \tag{5}$$

$$F_{pos} = 2 * \frac{P_{pos} * R_{pos}}{P_{pos} + R_{pos}} \quad \& \quad F_{neg} = 2 * \frac{P_{neg} * R_{neg}}{P_{neg} + R_{neg}} \tag{6}$$

Results Our submission achieved an average F-measure of positive and negative classes of 62.00%. A comparison of the competing systems on the past two years' data shows that our system ranked 7th on 2013 Twitter data, 10th on 2014 Twitter data, 6th on 2014 Live Journal data, 18th on SMS messages from 2013, and 10th on Twitter 2014 Sarcasm data. Except SMS dataset, our results are in similar ranks, which is a feature few other teams have. This demonstrates robustness in performance. Also, the fact that our system ranked better on 2015 than on 2014 suggests no over-fitting happened. The detailed official results of selected teams are shown in Table 34.

Team (40)	#	**Tw15**	Sarc15	Tw14	Sarc14	Live-J	Tw13	SMS
Webis	1	**64.84**	53.59	70.86	49.33	71.64	68.49	63.92
unitn	2	64.59	55.01	73.60	55.44	72.48	72.79	68.37
Splusplus	5	63.73	60.99	**74.42**	42.86	**75.34**	**72.80**	67.16
IOA	7	62.62	**65.77**	71.86	51.48	74.52	71.32	68.14
CLaC-SentiPipe	9	62.00	58.55	70.16	51.43	73.59	70.42	63.05
Grad-Analytics	16	60.62	56.45	66.87	**59.11**	72.63	65.29	61.97
ECNU	18	59.72	52.67	66.37	45.87	74.40	65.25	**68.49**
average		57.13	52.12	64.88	47.02	68.13	63.22	60.19
median		59.42	54.48	66.21	47.00	69.98	65.68	62.05

Table 34: Selected teams' average F-measure results from SemEval 2015 Task 10B

Our system's detailed precision (P), recall (R) and average F-measure of negative and positive classes (av. F) results for each polarity class over all datasets along with overall average F-measure of positive and negative classes are illustrated in Table 35.

7.4.4 Task 11: Sentiment Degree Association for Figurative Language Tweets

Approach We tested the robustness of our linguistically motivated lexical features by submitting the same pipeline for text processing, feature creation and first phase of the exhaustive feature

	positive			negative			neutral			overall
dataset	P	R	av. F	P	R	av. F	P	R	av. F	av. F
Twitter2015	75.58	63.20	68.84	43.51	75.34	55.17	66.63	60.08	63.19	62.00
Twitter2015-sarcasm	55.56	55.56	55.56	61.54	61.54	61.54	43.75	43.75	43.75	58.55
LiveJournal2014	79.33	66.51	72.36	68.39	82.57	74.81	67.87	68.86	68.36	73.59
SMS2013	59.26	68.29	63.46	54.39	73.86	62.65	83.55	68.60	75.34	63.05
Twitter2013	73.45	75.13	74.28	59.50	75.54	66.57	75.66	66.52	70.80	70.42
Twitter2014	78.76	70.98	74.67	58.53	74.75	65.65	63.10	66.97	64.97	70.16
Twitter2014Sarcasm	50.91	84.85	63.64	90.91	25.00	39.22	40.00	61.54	48.48	51.43

Table 35: Official CLaC-SentiPipe results for Task 10B in 2015

combination evaluation technique of (Shareghi and Bergler, 2013a,b) via 10-fold cross validation on the training set with M5P (Wang and Witten, 1997), a decision tree regressor, with the same pipeline parameters that we selected for Task 10B. We conducted 10-fold cross validation to create predictions and then calculated correlation coefficients (Nelson, 2001).

The extracted features are the same as the features we extracted for Subtask 10B. The only difference is the gold labels since Task 11 requires continuous classes while these are discrete in Subtask 10B.

We used float-based gold labels for training data and treat the problem as a regression problem. The output of our system's predictions were then rounded to integer values, due to this was a requirement of the task.

The submitted combination; $f_{1,2,3,6,7,10,11,12}$ containing AFINN-111, MPQA, Liu, Gezi unigrams-bigrams, and secondary feature set, totally 90 features[10]; is selected upon the results on 10-fold cross validation experiment since it created the best performance.

Evaluation metrics The challenge reports two evaluation metrics: Cosine similarity and mean-squared error. Given n-sized two vectors of t, vector of gold label values and e, vector of predicted label values; formula for cosine similarity can be seen in Equation 7 and formula for mean-squared error can be seen in Equation 8.

$$\cos(\mathbf{t},\mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n}\mathbf{t}_i\mathbf{e}_i}{\sqrt{\sum_{i=1}^{n}(\mathbf{t}_i)^2}\sqrt{\sum_{i=1}^{n}(\mathbf{e}_i)^2}} \tag{7}$$

[10] We excluded NRC unigrams-bigrams as Gezi has similar characteristics with NRC and performs better. Dependency based features were not present in our experiments, however we show that they result in better scores.

$$\text{MSE}\ (\mathbf{t},\mathbf{e}) = \frac{1}{n}\sum_{i=1}^{n}(t_i - e_i)^2 \tag{8}$$

Results The single submission of our system, CLaC-SentiPipe, ranked first in both, cosine and mean squared error measures. There were wide margins between the first three systems.

Results for the different forms of non-literal language can be seen in Table 36. In mean square error, CLaC-SentiPipe ranked first in the *overall* score, the *metaphor*, and *other* categories. For the cosine measure, the third system of a competitor obtained best performance in the *other* category, but with a high mean squared error.

The second best system, interestingly, does not hold best performance in a single category, which demonstrates the good performance of a robust approach. The third ranked team obtained best performance for irony both in cosine similarity and least squared error, but not in their best performing submission.

	Team (15)	#	Overall	Sarcasm	Irony	Metaphor	Other
Cosine	CLaC-SentiPipe	1	**0.758**	0.892	0.904	**0.655**	0.584
	UPF	2	0.711	0.903	0.873	0.520	0.486
	LLT PolyU	3	0.687	0.896	**0.918**	0.535	0.290
	elirf	5	0.658	**0.904**	0.905	0.411	0.247
MSE	CLaC-SentiPipe	1	**2.117**	1.023	0.779	**3.155**	**3.411**
	UPF	2	2.458	**0.934**	1.041	4.186	3.772
	LLT PolyU	3	2.600	1.018	**0.673**	3.917	4.587
	elirf	8	3.096	1.349	1.034	4.565	5.235

Table 36: Official results of selected teams for SemEval Task 11.

Our system has shown robustness across Task 10B and Task 11, and the linguistic features encoded have been validated for their adaptability to figurative language.

Chapter 8

Analysis

8.1 Analysis of Results

Results for different feature bundles from our ablation studies are compared in Table 37. The results in italics represent official challenge results, while results in bold represent the best performing bundle for given datasets.

Our choice of system for Task 10B, which has its result in italics in Table 37, was informed by good performance on both, 2013 and 2014 datasets. Our best performing feature bundle is only marginally better and leaves a 2% gap to the competition winner.

Our choice for Task 11, which has its result also in italics in Table 37, was the best performing combination on correlation results on a 10-fold-cross validation experiment on the training data. Note that our competition submission does not include dependency features, however, if we include them instead of MPQA and Bing Liu feature subsets, performance increases by a cosine difference of .01.

We notice that there is no feature bundle that performs best on all datasets. Since the best performers for each dataset include almost all features, we conclude that the different features are compatible and at least to a small degree encode complementary information. However, the feature bundle that contains all features is never the top performer, indicating some interference between features.

Our single feature ablation performance table of each lexicon used as sole lexical resource is

	Task 10B F1 measures			Task 11
feature ids	2015	2014	2013	Cosine
$f_{1,3,5,6,7,8,9,10,11,12}$	**62.64**	69.57	70.61	0.763
$f_{1,2,3,6,7,8,9,10,11,12}$	62.38	69.90	**70.85**	0.767
$f_{1,2,3,4,5,6,7,8,9,10,11,12}$	62.18	69.98	70.81	0.765
$f_{1,2,3,6,8,9,10,11,12}$	*62.00*	**70.16**	70.42	0.761
$f_{1,3,6,7,8,9,10,11,12}$	61.88	68.97	70.03	0.765
$f_{1,6,7,8,9,10,11,12}$	61.31	68.63	70.06	**0.768**
$f_{1,3,4,5,7,8,9,10,11,12}$	61.25	67.96	70.36	0.757
$f_{3,6,7,8,9,10,11,12}$	60.77	67.80	68.37	0.763
$f_{1,2,3,6,7,10,11,12}$	60.17	65.80	66.91	*0.758*
$f_{1,6}$	58.28	65.48	65.40	0.576
$f_{1,4}$	57.33	63.01	64.15	0.617

Table 37: Performance of different feature bundles.

illustrated in Table 38 across datasets. Surprisingly, AFINN-111, the smallest, manually compiled lexicon, does not only outperform the others, but enhanced with our linguistic context annotations performs only 12% worse than the best combination on 2015 test set. The reason behind this may be the design criterion for AFINN-111 (Nielsen, 2011) to exclude terms that may have conflicting sentiment prior polarities altogether.

	Task 10B F1 measures			Task 11
feature ids	2015	2014	2013	Cosine
f_1:AFINN-111	54.97	60.26	62.19	0.558
f_6:Gezi unigrams	54.65	60.81	57.86	0.554
f_3:Bing Liu	53.88	53.90	57.20	0.555
f_2:MPQA	52.22	51.42	53.39	0.548
f_4:NRC unigrams	49.83	52.39	50.90	0.609

Table 38: Results on lexical feature subsets on Task 10B.

Comparing Gezi to NRC, we see that adding syntactic scope treatment while enlarging the size of the tweet collection for automatically creating a resource increases its efficiency. This point is also supported by the fact that Gezi intersects and agrees with manually created lexica to a higher degree than NRC, mentioned in Section 7.2. Also, Gezi unigrams outperforms NRC unigrams in the two-lexicon-only runs $f_{1,6}$ (AFINN-111, Gezi unigrams) and $f_{1,4}$ (AFINN-111, NRC unigrams) of Table 37.

8.2 Error Analysis

Determining one system's error cases, and analyzing them is salient to identify general confusion situations of the system. By detecting errors and proposing possible solutions can establish future directions.

We targeted frequent errors of our linguistically motivated lexical features oriented system on our SemEval submissions after defining what an error means for each task.

8.2.1 Errors on Polarity Classification of Tweets

Definition of Error Defining errors for polarity classification is trivial, a misclassified tweet is an error instance. In Table 39, where the confusion matrix of our 2015 SemEval Task10B's submission, non-bolded cells are error instances (e.g. *positive* gold labels predicted as *negative*).

Notice that since the official evaluation metric is average F-measure of positive and negative classes and negative instances are less frequent than positives, misclassifying negatives as another class or misclassifying another class as negative give more penalty, especially misclassification between positive and negative classes. Thus, we give our focus to misclassifications between those two classes in analysis.

Predicted Gold	neutral	positive	negative
neutral	**596**	244	58
positive	184	**659**	34
negative	207	135	**273**

Table 39: Confusion matrix of polarity classification task submission.

Error Cases

- ***Tweets with contrastive discourse markers***

  The most frequent case of errors are coming from tweets with contrastive discourse connectors. Out of 2340 instances of 2015's test set, 171 conjuctors and 44 adverbs of contrastive discourse connectors are present. In our system, their existence has only been checked. However, they encode semantics in a way that one side of two sides contrasting each other is emphasized. The form of *(modality(polarityX), but polarityY)* $-$ > *sentiment(polarityY)* occurs frequently as in Example 16a and Example 16b. The reason that our system did not distinguish the meaning behind, and does not encode it, thus it can fail easily.

Example 16. a. *I may not like the walking dead but Norman reedus is pretty attractive.*
(gold:positive, predicted:negative)
b. *Yeah, Christian Ponder may be a great guy, but he is a complete dumpster fire behind center.*
(gold:negative, predicted:positive)
c. *@OffsideLiam Carlo Ancelotti after LVG and then may be Giggs. Not a big fan of Guardiola, despite the success.*
(gold:negative, predicted:positive)

In Example 16c another contrastive discourse marker, *despite*, encodes contrast in a different form: *(polarityX, despite polarityY)* $-$ > *sentiment(polarityX)*. Therefore, for each contrastive discourse marker sentiment specific semantic compositionality rules might be adopted. This is left as a future work.

- ***Tweets with comparatives***

Comparatives can be tricky for sentiment. Since, there are two parts compared; one part is better and the other is worse relatively. Yet, to identify an overall sentiment becomes harder. In Example 17a, the user mentions a situation will get better implying that the current situation is worse; and the emoticon occurring at the end shows that the person is unhappy about it. Our system labels the tweet as negative also by relying on the negative emoticon, however, annotators perceive it as a positive message since the new situation will be better. This is arguable by depending on one's interpretation.

Example 17. a. *Nash's and Skylynn's video tomorrow ... I'm sure that the video will make me feel better /:*
(gold:positive, predicted:negative)
b. *Three hours of sleep would be a lot worse if it was Monday. It's a Sugar Free Red Bull breakfast morning. #TGIF*

(gold:negative, predicted:positive)

A special case in Example 17b is interesting such that a hypothetical situation which did not happen is mentioned as something bad but could be worse implying that it was not really bad. Here we see modality and conditional bring implication which matters for sentiment. However, annotators saw this situation of something bad did not get worse as negative which is arguable; since it might sound positive when approached optimistically. In the next section, comparatives are investigated briefly; however we leave it mostly for future work.

- ***Tweets with sentiment-laden terms of certain emotions***

  Sentiment is often expressed with emotions. If the attributes of emotional terms conveying sentiment could be encoded, a better analysis is highly possible.

  In Example 18a, a term of surprise is used. Although surprise occurs in positive context generally, here the surprise term, *wow*, encode an attitude of something unexpected toward something negative. Having positive sentiment score from lexica we use and being encoded as a structural feature by occurring at the beginning of the message make classifier biased and it misclassified the tweet. Another case is emotions can be exaggerated by using idiomatic expressions which may contain terms of opposite polarity (e.g. *tears of joy*). We see *to die because of excitement* in Example 18b, where a very negative term fails our classifier.

Example 18. a. *wow the 4th season of AHS is like not good*

(gold:negative, predicted:positive)

b. *Tomorrow is the out of Steal My Girl, next day, out of WWAT film, the next day,,, I'm gonna DIE because of excitement...*

(gold:positive, predicted:negative)

c. *My sincere condolences to Lee Soo Man's wife. May she rest in peace.*

(gold:positive, predicted:negative)

Another question is what to do when emotions are positive towards a person in a negative event? In Example 18c, positive wishes are expressed for someone who passed away. Our classifier labelled the tweet with the polarity of the event even though the tweet contains highly positive terms in it, yet failed. This is again a question of what perspective should be taken towards text containing positive wishes on undesired events that needs to be answered.

- ***Adjective scope over sentiment-laden terms***

  Adjectives play an important role in sentiment analysis. They may not only be an attribute of someone or something, but they may also have an impact on context. Although they are extreme examples, Example 19a and Example 19b illustrate how interestingly adjectives impact semantics. Example 19a implies that something desired did not come true, therefore it created disappointment[1]. In Example 19b, a negative adjective which seems to be used as an implicit negation trigger, scoped over a positive adjective which is under scope of negation; created an overall meaning with strong positive sentiment.

  Example 19. a. *I know it was too good to be true OTL*
  (gold:negative, predicted:positive)
  b. *I find it very difficult not to be happy with him*
  (gold:positive, predicted:negative)

  As a future work, adjective scope can be included into linguistic notions used for sentiment.

- ***Tweets with opposite sentiment triggers***

  In tweets having opposite sentiment triggers, generally one particular sentiment outweighs the other. This may happen due to one particular focus as in Example 20a or quantitatively outweighs as in Example 20b. In Example 20c and Example 20d, positive starts of messages dominated against negative endings. For these errors, compositionality could be used to determine which phrases or sentences are more focused or dominant.

  Example 20. a. *#DailyNBA Kobe Bryant looked like his old self against the Jazz during a 119-86 loss on Thursday, making 10-of-23 from the field and ...*
  (gold:positive, predicted:negative)
  b. *Perseverance is failing 19 times and succeeding the 20th. Julie Andrews http://t.co/QaNj6I3Nbo*
  (gold:positive, predicted:negative)

[1] Notice that when something is good to be true, it may also imply that is desired which is positive. But, the meaning here is implied by using past tense and a social-media term (OTL) used for disappointment

c. *Can't wait for tomorrow. #ProperFootball Only problem is my usual pub is showing the Arsenal game instead.*
(gold:positive, predicted:negative)
d. *@NiallOfficial i hope you wish me happy bday tomorrow Niall or i'll be so sad*
(gold:positive, predicted:negative)

- ***Tweets with unfamiliar terms or terms that require common-sense knowledge***

Automatically compiled lexica may cover too many terms which may be recent and give an idea about its general sentiment. However, the collections of the resources we used in our system have tweets until May 2014. The term *Ebola* in Example 21a appeared after this date, in which our system is incapable of having sentiment information about it. Second of all, the phrase of *to go crazy* in Example 21b is generally used for negative sentiment, which exists in our bigrams with a negative score. Nonetheless, in sports domain, the term is used for a team or player played excellent. Therefore, domain-based common sense information seems beneficiary for sentiment purposes.

Example 21. a. *Obama is a ditherer. (Or you can call him a deep thinker. Both correct.) But for the love of Ebola, DO SOMETHING. http://t.co/QGjbLVc2tg*
(gold:negative, predicted:positive)
b. *Bulls came back Butler went crazy in the 4th*
(gold:positive, predicted:negative)

- ***Tweets with figurative language***

One of the features of the nature of the social media is that users use figurative language more frequently. In Example 22a, the annotators labelled the tweet without seeing the irony explicitly stated by the hashtag at the end, which was captured by our system. However, when figurative language is implicitly conveyed as in Example 22b, it becomes hard to be determined especially when you have training data is not prepared for it.

Example 22. a. *So today we think @WHO r going to complete their draft to destroy #ecigs & Ashton is officially reinstated. Happy Friday guys. #sick*
(gold:positive, predicted:negative)

b. *Ben Affleck and Henry Cavill are in my city. Sucks to know that I'm only the 3rd sexiest dude here for a day.*
(gold:positive, predicted:negative)

8.2.2 Errors on Sentiment Degree Association for Figurative Language

Definition of Error For official evaluation on Task 11, our regression model calculated continuous real values between -5 and 5 as predictions. These values were required to be rounded to integers for submission to create equality between participants using regression models or classification ones. Then, evaluation was performed by using the metrics of cosine similarity and mean-squared error between those predicted rounded integer-based values and float-based gold values. Defining errors is not trivial, since no schema is defined for errors nor exactly matching of predicted and gold values, even both are rounded to integers, does not give expected insight. Otherwise, the instances predicted closely but not exactly can be found as errors (e.g. -2 predicted as -3). Therefore, after rounding gold and predicted values to integers, we proposed some error schemas for the task:

1. if gold and predicted values are not equal ($p \neq g$)
2. if absolute value of difference of gold and predicted values is greater than 1 ($|p-g| > 1$)
3. if absolute value of difference of gold and predicted values is greater than 2 $|p-g| > 2$

When these conditions of error schemas are applied, the instance numbers accepted as correct and errors, and the accuracy for each condition can be seen in Table 40.

error condition	correct	error	accuracy
$p \neq g$	2518	1481	62.97
$\|p-g\| > 1$	3085	914	*77.14*
$\|p-g\| > 2$	3664	335	91.62

Table 40: Accuracy table for each error condition.

From these error conditions, we select $|p-g| > 1$ for our error schema, given that $p \neq g$ is a very strict condition defines highly close predictions as errors (e.g. -3 and -4), and $|p-g| > 2$ achieves a high accuracy narrowing our error instances.

When the gold and predicted values are rounded to integers for discretization of continuous values, the confusion matrix of 11 discrete classes of our submission task is illustrated in Table 41. The bolded cells demonstrates the correct instances according to the selected schema's condition of $|p - g| > 1$.

Predicted Gold	-5	-4	-3	-2	-1	0	1	2	3	4	5
-5	**1**	**1**	0	2	0	0	0	0	0	0	0
-4	**1**	**3**	**15**	57	22	1	1	0	0	0	0
-3	1	**18**	**113**	**456**	120	22	6	1	0	0	0
-2	0	10	**105**	**1073**	**277**	66	6	3	0	0	0
-1	1	1	35	**387**	**175**	**58**	20	2	1	0	0
0	0	0	8	90	**109**	**63**	**25**	3	0	0	0
1	0	0	5	28	56	**44**	**26**	**9**	1	0	0
2	0	0	1	28	52	33	**31**	**9**	**1**	0	0
3	0	0	0	12	32	36	55	**53**	**12**	**1**	0
4	0	0	0	9	23	12	17	31	**11**	**6**	**2**
5	0	0	0	1	0	1	0	1	1	**0**	**0**

Table 41: Confusion matrix of sentiment degree association on figurative language task submission.

Error Cases Test set of the Task 11 contains both figurative and non-figurative language. Therefore, to generate an error analysis, it is important to know whether a tweet has figurative language or not and if it does, which figurative language type it is. However, those labels are not supplied to participants by the organizers of the task, even though it was requested. For this reason, a constrained error study has been conducted.

- ***Tweets with terms triggering literal meaning***

  One disadvantage of our non-bag-of-words approach is not able to represent terms as features in the feature space. The terms like *literally* encodes literal meaning to avoid expressing figurative language as in Example 23a and Example 23b, but there are exceptions like Example 23c[2].

Understanding if a tweet contains figurative language sometimes can be troublesome in the cases it is encoded implicitly. In Example 23a, in the hypothetical situation that someone receives a terrible news, the tweet becomes ironic.

Example 23. a. *RT @FreakinNess: literally received the best news today. #GodIsGreat*

(gold:4, predicted:-1)

b. *apples are literally magical whenever im not feeling good i can eat just eat an apple and i feel better*

(gold:3, predicted:-2)

c. *It's literally the best feeling being ditched mid conversation #not*

(gold:-3, predicted:-2)

- ***Training on structural features***

In the training data, users gave specific hashtags at the end of the tweets stating irony and sarcasm such as *#not, #irony.* Otherwise, the figurative language in body of tweets could not be shown explicitly. Since conveying figurative language implicitly may avoid figurative language captured by readers. These hashtags are represented in the Gezi Lexicon with negative sentiment scores. Due to the fact that in our secondary features which contain also structural features of *last token's sentiment score* and *last token's POS tag*, these features modelled training data highly with those hashtags indicating always figurative language. Recall that training data contained figurative language. Therefore, the tweets having non-figurative language content and a hashtag at the end as in Example 24a and Example 24b, generally got predicted incorrectly. This may be solved by taking an feature to differentiate between hashtags stating figurative language and others.

One complaint of ours here is there are many non-figurative tweets repeatedly exist in test set such as Example 24a.

Example 24. a. *Hi angus It's @nightskyxniall but I got on tweet limit To finally speak to my sunshines means so much #Bumps5SOSFanCall*

[2] Notice that this example is not an error and is given to outline the exception.

(gold:2, predicted:-2)

b. *Great to speak to Alan Kennedy again last night, top bloke @5timesco Both impressed with @lfc18alberto so far #leftbackunion*

(gold:4, predicted:-1)

8.3 Special Cases

8.3.1 Investigating Adverbs in SemEval 2013 Tweet Dataset

An adverb can modify a verb, an adjective, another adverb, a phrase, or a clause. An adverb indicates manner, time, place, cause, or degree and answers questions such as *how, when, where, how much.*[3]

Since adverbs are modifiers in contexts, it is worth to investigate them in Sentiment Analysis. For such an experiment, part of speech tags are annotated in GATE environment. In order to do that; a pre-annotator pipeline is built with the CMU tokenizer, Stanford Sentence Splitter and the CMU POS tagger, which runs on training and development sets of SemEval 2013 Tweet Dataset. Since the CMU tokenizer and POS tagger are trained on twitter data, they can deal with elongated, upper-cased or mistyped expressions in the data.

The aim of the experiment is to find the correlation between adverbs and different polarity classes of tweets.

After having tokens and part-of-speech tags in Penn Treebank style POS tags, in our pipeline 4 different tags are used: RB (general adverb), RBR (comparative adverb), RBS (superlative adverb), and WRB (Wh-adverb). Table 42 shows, how many adverbs occurred in different sentiment classes within two different datasets. The information of how many adverbs occurred uniquely in those datasets is also conveyed.

Data Set	unique	total	positive	negative	neutral
training data	461	5165	2109	1121	1935
development data	179	822	316	246	260

Table 42: Adverb information per message-level data set.

[3] from http://www.writingcentre.uottawa.ca/hypergrammar/adverbs.html

Occurrences of unique adverbs with no preprocessing from the Task 10B's training data in different classes are recorded as they can be seen in Table 43.

adverb	positive	neutral	negative
Hopefully	16	1	0
hopefully	9	2	0
forward	44	5	2
EVER	8	1	0
ever	45	14	9
Never	7	4	0
never	16	5	13
tho	14	5	1
When	6	15	1
Why	5	14	18
why	9	15	22
much	33	9	10
there	48	59	16
pretty	19	4	6
Pretty	2	0	3
more	59	58	21
too	53	25	20
SO	4	0	2
sooooooo	1	0	0
soo	3	0	0
NOT	3	1	4
not	63	74	116
fucking	2	1	3
fuckin	2	0	2
Unofficially	0	7	1
anymore	1	1	6
damn	1	2	6
very	25	10	16
just	149	187	97
still	50	54	27

Table 43: Selected adverbs from of Task 10B's training data

Since adverbs' modifications do not always have wide scope over sentences, a better experiment can be looking for adverbs in contextual gold standard data. However, Table 43 still gives us ideas how adverbs behave differently: Some occurs in certain classes like *hopefully*, some are distributed proportionally to sentiment classes according to the data like *just*. Some are in only sentiment-laden classes like *SO*. In addition elongation and/or all-capitalization seem to have effect on adverbs behaviours.

Since most of the lexica that are built for sentiment analysis tasks do not have POS tags for adverbs, simple lookup may mislead systems. For example, *pretty* can be an adjective, which mostlikely modifies a noun positively; or as an intensifier adverb which may intensify a negative adjective by increasing its negativity level. Again, it may occur to a diminisher adverb which modifies an adjective or a verb. Thus, a lexicon which keeps the terms with their part of speech tags should be considered, like MPQA.

8.3.2 Comparatives and Superlatives

In error cases we demonstrated that our system may fail easily on tweets containing comparatives. Here, we only give details on how many instances of comparatives there are in the Task 10B datasets. The same is also applied for superlatives.

Comparatives and superlatives are identified in two ways: Relying on POS tags of comparatives and superlatives from the CMU POS tagger and relying on the Gezi resources' terms.

In both ways, unigram words are identified through RBR, RBS, JJR and JJS tags and bigram words are identified with the same tags modifying adverbs or adjectives. In addition, we extracted superlatives and comparatives from Gezi unigrams and bigrams, where bigrams only having the adverbs of *more* and *most*.

In Table 44, apart from the last the row of Gezi's terms' superlatives and comparatives, rows show how many different types of superlatives and comparatives occur in different data sets detected with the CMU POS tagger. It is surprising that trigger counts are very close to total superlatives and comparatives found by using POS tags. This can be seen by comparing the last two rows. Note that the numbers next to the data set names are how many tweets they contain.

Since our automatically-compiled lexicon detects the comparatives in a wide coverage, its association score for a comparative can be used differently. This could be done by checking the cases in which they behave according to the perspective or the condition in the messages. Then, semantically compositional rules may be encoded to treat them in isolation.

8.3.3 Structure of Tweets

Tweets may have different structures by the nature of the social media design: they can be a single entry, a reply to another tweet, a retweet where a tweet is emphasized by another user, a modified

Type	POS tag	Train	Dev	Test2013	Test2014
		6,822	1,042	3,813	1,853
unigram with CMU POS	RBR	143	13	91	30
unigram with CMU POS	RBS	1	1	1	0
unigram with CMU POS	JJR	155	25	93	32
unigram with CMU POS	JJS	320	63	166	86
bigram with CMU POS	RBR	25	0	19	5
bigram with CMU POS	RBS	1	1	1	0
bigram with CMU POS	JJR	8	3	6	3
bigram with CMU POS	JJS	36	7	11	14
unigram with CMU POS	total	619	102	351	148
bigram with CMU POS	total	70	11	37	22
unigram+bigram with CMU POS	total	689	113	388	170
Gezi unigram+bigrams	-	655	109	361	159

Table 44: The occurences of unigrams, bigrams and lexical triggers (Gezi) in different datasets.

tweet where a tweet is shared with additional comments from another user, or a URL with its content from an external source.

In our study, we treated tweets assuming that they are single independent entries with their content. This assumption is not necessarily true, however we made it for the reason that the data supplied by the task organizers as individual instances.

We leave it to future work to analyze such structures of tweets for sentiment analysis of tweets.

Chapter 9

Conclusion

We proposed and tested how beneficial and useful it is to enhance lexical features with negation and modality for sentiment analysis of tweets. We tested compact frequency-based and context-aware lexical features by creating those features over polarity classes. This gave us the ease of adopting negation and modality together for the feature space, even though it quadruples the lexical features. Using modality context with negation for sentiment analysis of tweets is a novel approach.

Gezi is a negation-aware automatically created huge lexicon that is produced as a contribution of this study. It is a replication of the NRC Hashtag Lexicon with several additional attributes such as negation context and detailed preprocessing. In our ablation studies on sentiment polarity classification of tweets task and agreement-to-traditional-lexica studies, we observed that those additional attributes made Gezi perform better. It yet cannot outperform AFINN-111, the smallest sentiment lexicon which excludes terms having senses with different polarity. In addition to these, we demonstrated that automatically-compiled lexica can be adjusted to contain discrete prior polarity classes. Finally, giving fine-grained prior polarity labels are shown to be better than the basic categories of positive, negative, and neutral, which is tested on sentiment polarity classification of tweets task.

Creating linguistic information for informal text seems harder than formal text given that tokenizing and sentence splitting is required for parsing and these steps have new obstacles due to the nature of informal text which contains new components of acronyms, emoticons and even intentionally mistyped words. We have shown that with a careful preprocessing (i.e. a hybrid tokenization, using appropriate POS tags and parser models) makes it feasible and useful by creating more accurate tokens, sentences and POS tags practically.

Our decisions of the system design and selection of tools have been made by running comparative experiments validated with statistical significance tests. The main conclusions of these experiments on the task of sentiment polarity classification of tweets, and those conclusions are applied on the task of determining strength of sentiment in figurative language of tweets:

- Scaling features does not improve the results on top of our system
- Using a hybrid tokenizer for tweet processing significantly outperforms using a single tokenizer for informal text
- Preprocessing steps are necessary for more accurate parse trees of tweets
- Fine-grained prior polarity labels for automatically-compiled lexica combined with linguistic context improves the results significantly for sentiment polarity classification of tweets
- Using negation and modality context with sentiment lexica improves performance significantly for sentiment polarity classification of tweets

Our performance in different SemEval 2015 challenges demonstrated that our approach is robust. In the tweet-level polarity classification task, our system resulted within 3% margin of the best performer system's result.

Without adapting the system on measuring sentiment degree in figurative language pilot task, our system achieved the best in competition results. Therefore, a regressor can effectively derive sentiment values for tweets with figurative language from these general features when explicitly encoded without bias towards non-literal language. In addition, in our ablation studies, we observed that using dependency-based features improves the results.

For different lexica built for sentiment analysis purposes: although increasing the size of automatically created lexicon may have an impact on improving the results, the bigger is not proportionally better. For manually created lexica, we observe that the smallest lexicon performs the best with the attributes of excluding terms that may have different polarity senses depending on the context, and of being compiled manually from tweets with both polarity classes and association scores.

Our system could be improved by taking care of specific linguistic phenomena: Semantic compositionality for intensifier and diminisher adverbs, contrastive discourse markers, comparatives, and/or scopes of adjectives could be integrated to the feature space. These points together with the other error cases discussed in Chapter 8 are left for future directions.

Another future work is benefiting from Grice's Maxims (Grice, 1975) and Horn's scalar implicature (Horn, 1984). Extracting what is implied from a subjective text can be leveraged to boost sentiment analysis systems.

Bibliography

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of Twitter data. In *Proceedings of the Workshop on Languages in Social Media, 9-14 July 2011*, LSM '11, pages 30–38, Portland, Oregon, USA.

Andreevskaia, A. and Bergler, S. (2006). Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*.

Andreevskaia, A. and Bergler, S. (2008). When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 290–298.

Barbosa, L. and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 18-22 August 2008*, COLING '10, pages 36–44, Beijing, China.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. (2013). TwitIE: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, 9-11 September 2013, Hissar, Bulgaria*.

Broß, J. (2013). *Aspect-Oriented Sentiment Analysis of Customer Reviews Using Distant Supervision*

Techniques. PhD thesis, Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany.

Carston, R. (1998). Informativeness, relevance and scalar implicature. In Carston, R. and Uchida, S., editors, *Pragmatics And Beyond New Series*, pages 179–238. John Benjamins Publishing Co.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

Choi, Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, 25-27 October 2008*, EMNLP '08, pages 793–801, Honolulu, Hawaii, USA.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1).

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: an architecture for development of robust HLT applications. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, July 2002 (ACL'02)*, pages 168–175, Philadelphia, Pennsylvania, USA.

Cunningham, H., Tablan, V., Roberts, A., and Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS Computational Biology*, 9(2).

de Marneffe, M.-C. and Manning, C. D. (2008). Stanford typed dependencies manual. Technical report, Department of Computer Science, Stanford University.

de Marneffe, M.-C. and Manning, C. D. (2008). The Stanford typed dependencies representation. In *COLING 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, 18-22 August 2008*, CrossParser '08, pages 1–8, Manchester, United Kingdom.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06, 24-26 May 2006, Genoa, Italy*, pages 417–422.

Farkas, R., Vincze, V., Móra, G., Csirik, J., and Szarvas, G. (2010). The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task, July 15-16, 2010*, CoNLL '10: Shared Task, pages 1–12, Uppsala, Sweden.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press, Cambridge, MA, USA.

Frantova, E. and Bergler, S. (2009). Automatic emotion annotation of dream diaries. In *Proceedings of the Analyzing Social Media to Represent Collective Knowledge Workshop at K-CAP 2009, The Fifth International Conference on Knowledge Capture, 1-4 September 2009, Redondo Beach, California, USA*, California, USA.

Ghasemi, A. and Zahediasl, S. (2012). Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism*, 10(2):486.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., and Reyes, A. (2015). SemEval-2015 Task 11: Sentiment analysis of figurative language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 1-5 June 2015*, pages 470–478, Denver, Colorado, USA.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, 19-24 June 2011*, HLT '11, pages 42–47, Portland, Oregon, USA.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, http://www.stanford.edu/ alecmgo/papers/TwitterDistantSupervision09.pdf*, pages 1–12.

Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York, USA.

Günther, T. (2013). Sentiment analysis of microblogs. *Master's Thesis, University of Gothenburg.*

Günther, T. and Furrer, L. (2013). GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), June 2013*, pages 328–332, Atlanta, Georgia, USA.

Günther, T., Vancoppenolle, J., and Johansson, R. (2014). RTRGO: Enhancing the GU-MLT-LT system for sentiment analysis of short messages. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 497–502, Dublin, Ireland.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181, Madrid, Spain.

Horn, L. (1984). Toward a new taxonomy for pragmatic inference: Q-based and R-based implicature. *Meaning, form, and use in context: Linguistic applications*, pages 11–42.

Horn, L. (2001). *A Natural History of Negation.* University of Chicago Press.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 22-25, 2004*, KDD '04, pages 168–177, Seattle, WA, USA.

Kennedy, A. and Inkpen, D. (2005). Sentiment classification of movie and product reviews using contextual valence shifters. In *Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations, FINEXIN 2005, May 26-27 2005*, Ottawa, Canada.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.

Kilicoglu, H. H. (2012). *Embedding Predications.* PhD thesis, Department of Computer Science and Software Engineering, Concordia University.

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, June 4-5 2009*, BioNLP '09, pages 1–9, Boulder, Colorado, USA.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics, August 23-27 2004*, COLING '04, Geneva, Switzerland.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research (JAIR)*, 50:723–762.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, 7-12 July 2003*, ACL '03, pages 423–430, Sapporo, Japan.

Kökciyan, N., Çelebi, A., Özgür, A., and Üsküdarlı, S. (2013). Bounce: Sentiment classification in Twitter using rich feature sets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), June 2013*, pages 554–561, Atlanta, Georgia, USA.

Liu, B. (2010). Sentiment analysis and subjectivity. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL. ISBN 978-1420085921.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Aggarwal, C. C. and Zhai, C., editors, *Mining text data*, pages 415–463. Springer US.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting* *of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Mihalcea, R., Banea, C., and Wiebe, J. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, June 2007*, pages 976–983, Prague, Czech Republic.

Miura, Y., Sakaki, S., Hattori, K., and Ohkuma, T. (2014). TeamX: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), August 2014*, pages 628–632, Dublin, Ireland.

Mohammad, S., Kiritchenko, S., and Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), June 14-15, 2013*, pages 321–327, Atlanta, Georgia, USA.

Moody, R. and Lindberg, C. A. (2012). *Oxford American Writer's Thesaurus*. Oxford University Press.

Moore, D. S. and McCabe, G. P. (1989). *Introduction to the Practice of Statistics*. W.H. Freeman.

Morante, R. and Blanco, E. (2012). *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, 7-8 June 2012*, SemEval '12, pages 265–274, Montréal, Canada.

Morante, R. and Daelemans, W. (2012). Annotating modality and negation for a machine reading evaluation. In Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF (Online Working Notes/Labs/Workshop), 17-20 September 2012, Rome Italy.*

Morante, R. and Sporleder, C. (2012). Modality and negation: An introduction to the special issue. *Comput. Linguist.*, 38(2):223–260.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). SemEval-2013 Task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), June 14-15, 2013*, pages 312–320, Atlanta, Georgia, USA.

Nelson, R. B. (2001). Kendall Tau metric. *Encyclopaedia of Mathematics*, 3.

Nielsen, F. Å. (2011). A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011*, pages 93–98.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010.*

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 380–390.

Özdemir, C. and Bergler, S. (2015a). CLaC-SentiPipe: SemEval2015 Subtasks 10 B, E, and Task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 1-5 June 2015*, pages 479–485, Denver, Colorado, USA.

Özdemir, C. and Bergler, S. (2015b). A comparative study of different sentiment lexica for sentiment analysis of tweets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, 5-11 September 2015, Hissar, Bulgaria.*

Palmer, F. R. (2001). *Mood and Modality.* Cambridge University Press.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, July 21-26, 2004*, ACL '04, Barcelona, Spain.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, July 6-7, 2002*, EMNLP '02, pages 79–86.

Polanyi, L. and Zaenen, A. (2006). Contextual valence shifters. In Shanahan, J., Qu, Y., and Wiebe, J., editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer Netherlands.

Porter, M. F. (1997). An algorithm for suffix stripping. In Sparck Jones, K. and Willett, P., editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, USA.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, July 27-29, 2011*, EMNLP '11, pages 1524–1534, Edinburgh, United Kingdom.

Roget, P. M. (2011). *Roget's International Thesaurus*. Harper Collins, Canada.

Rosenberg, S. (2013). Negation triggers and their scope. Master's thesis, Department of Computer Science and Software Engineering, Concordia University.

Rosenberg, S. and Bergler, S. (2012). UConcordia: CLaC negation focus detection at *Sem 2012. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), 7-8 June 2012*, pages 294–300, Montréal, Canada.

Rosenberg, S., Kilicoglu, H., and Bergler, S. (2012). CLaC Labs: Processing modality and negation. working notes for QA4MRE pilot task at CLEF 2012. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S. M., Ritter, A., and Stoyanov, V. (2015). SemEval-2015 Task 10: Sentiment analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 1-5 June 2015*, pages 451–463, Denver, CO, USA.

Rosenthal, S., Ritter, A., Nakov, P., and Stoyanov, V. (2014). SemEval-2014 Task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), August 23-24, 2014*, pages 73–80, Dublin, Ireland.

Sabou, M., Bontcheva, K., Derczynski, L., and Scharl, A. (2014). Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 859–866.

Shareghi, E. and Bergler, S. (2013a). CLaC-CORE: Exhaustive feature combination for measuring textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, June 2013*, pages 202–206, Atlanta, Georgia, USA.

Shareghi, E. and Bergler, S. (2013b). Feature combination for sentence similarity. In Zaane, O. and Zilles, S., editors, *Advances in Artificial Intelligence: 26th Canadian Conference on Artificial Intelligence, Canadian AI 2013, Regina, Canada, May 28-31, 2013*, volume 7884 of *Lecture Notes in Computer Science*, pages 150–161. Springer Berlin Heidelberg.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013a). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 455–465.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., and Potts, C. (2013b). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the* *conference on empirical methods in natural language processing (EMNLP), October 18-21, 2013, Seattle, WA, USA*, volume 1631, page 1642.

Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). *The General Inquirer: a Computer Approach to Content Analysis*. M.I.T. studies in comparative politics. M.I.T. Press, Cambridge, MA, USA.

Strapparava, C. and Mihalcea, R. (2007). SemEval-2007 Task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations, June 23-24, 2007*, SemEval '07, pages 70–74, Prague, Czech Republic.

Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014). Coooolll: A deep learning system for Twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), August 2014*, pages 208–212, Dublin, Ireland.

Thode, H. C. (2002). *Testing for Normality*, volume 164. CRC press.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, May 27-June 1, 2003*, NAACL '03, pages 173–180, Edmonton, Canada.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, July 6-12 2002*, ACL '02, pages 417–424, Philadelphia, Pennsylvania, USA.

Wang, Y. and Witten, I. H. (1997). Induction of model trees for predicting continuous classes. In *Poster in Proceedings of the 9th European Conference on Machine Learning, April 23-25 1997*, Prague, Czech Republic. Faculty of Informatics and Statistics.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, October 6-8, 2005*, HLT '05, pages 347–354, Vancouver, British Columbia, Canada.

Witten, I. H. and Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition.

Woolson, R. F. (2008). Wilcoxon signed-rank test. *Wiley Encyclopedia of Clinical Trials*.

Zhai, Z., Liu, B., Xu, H., and Jia, P. (2011). Clustering product features for opinion mining. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, 9-12 February 2011*, WSDM '11, pages 347–354, Hong Kong, China.

Zhu, X., Kiritchenko, S., and Mohammad, S. (2014). NRC-Canada: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), August 23-24, 2014*, pages 443–447, Dublin, Ireland.

Zwicky, A. M. and Pullum, G. K. (1983). Cliticization vs. Inflection: English N'T. *Language*, 59(3):502–513.