ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

PROGRAMME OF POSTGRADUATE STUDIES

IN COMPUTER SCIENCE

Department of Informatics

M.Sc. THESIS

SOCIAL MEDIA SENTIMENT ANALYSIS

Rafael - Michael Karampatsis

Supervisor: Ion Androutsopoulos

Assistant Supervisor: Prodromos Malakasiotis

ATHENS, JUNE 2014

Abstract

During the last years, the popularity of microblogging and social media services such as Twitter has increased significantly. Lots of users often use these services to express feelings or opinions about a variety of subjects. The analysis of this kind of content can extract useful information for fields such as personalized marketing or social profiling. In addition, it can help consumers decide whether to buy or not a certain product. However such a task is not trivial, because the language used in Social media is often informal presenting new challenges to text analysis. In this thesis we describe a system that was developed to detect sentiment in microblogging content such as Tweets or SMS messages in English. Our system has competed in two international challenges and has achieved very good results. We also apply our methodology to create a system for Greek. Finally, we propose ideas for future work.

Acknowledgements

I would like to thank my supervisor, Ion Androutsopoulos, for his consistent and restless guidance during the creation of this thesis. I would also like to thank all the members of the Natural Language Processing Group of AUEB's Department of Informatics and especially Makis Malakasiotis and John Pavlopoulos for their advice, discussions, support, excellent collaboration and patience. Special thanks to Qualia for providing the Greek data of Chapter 3 and their overall support. Finally, I would also like to thank all the people who stood by me during this process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of the thesis

The increasing popularity of social network services and the vast number of opinions published on the web call for sentiment analysis systems that will be able to process large amounts of data and will also be able to handle the special challenges of social media posts. Because of the interest in utilizing this freely available information by research and industry, sentiment analysis of social media has become a popular research topic during the last years.

Sentiment analysis is the field of study that analyzes people's sentiment and opinions from written (and less often also spoken) language (Liu, 2012). In this thesis we focus on sentiment analysis of social media posts. To address this problem we designed and implemented a system able to detect sentiment in social media messages written in English and decide about the polarity of these messages as well. The system focuses on tweets, but it has also been tested with messages of other genres, such as blog posts and SMS messages. The system has participated in two international challenges achieving good results. We have also implemented and experimented with a Greek version of our system.

1.2 Structure of the thesis

Chapter 2 gives a basic overview of sentiment analysis and the methods usually employed. Moreover, it presents the system we developed for sentiment analysis of social media posts written in English, with which we participated in two international challenges. The chapter also presents experimental results of the system and proposes possible future improvements.

Chapter 3 discusses the application of our methodology to Greek data in order to create a corresponding system for Greek social media messages. The chapter first presents the mofications or expansions that were made to existing Greek natural language processing tools and the Greek sentiment analysis system for social media we developed using them. Secondly, it evaluates the performance of the Greek system. Finally, it proposes ideas for future improvements of the Greek system.

Chapter 4 summarizes the conclusions and the contributions of this thesis.

Chapter 2

Message-level Sentiment Estimation for Social Networks [1]

2.1 Background and related work

Sentiment analysis (SA) is the field of study that analyzes people's sentiment and opinions from written (and less often also spoken) language. It can be performed at the document level, the message/sentence level or even the aspect/feature level. A popular strategy to deal with the task is to follow a two-stage approach. During the first stage, subjectivity detection, a text is classified as subjective if it expresses sentiment, or as objective if it does not. During the second stage, polarity detection, subjective texts are further classified as positive, negative, neutral or sometimes conflict. In some cases the intensity (e.g., strong, mild, weak) of the sentiment is also considered. The classification of texts using this taxonomy has been very popular for the last ten years (Liu, 2012; Pang and Lee, 2005; Tsytsarau and Palpanas, 2012). However, it is not rare for neutral texts to be considered as objective (Figure 2.1).[2] In this thesis we consider this simplification as it has been adopted by international challenges (Wilson et

[1] A summary of this chapter has already been published (Malakasiotis et al., 2013; Karampatsis et al., 2014).

[2] In this thesis we consider objective and neutral to be the same class and we will use the two terms interchangeably.

al., 2013; Rosenthal et al., 2014) our system participated in. Finally, some researchers have gone beyond polarity classification and have introduced classification to emotional states such as "angry," "sad," and "happy" (Bellegarda, 2010; Mohammad and Turney, 2013; Alm, 2005). In this chapter we focus on message level sentiment analysis for social networks. [3]

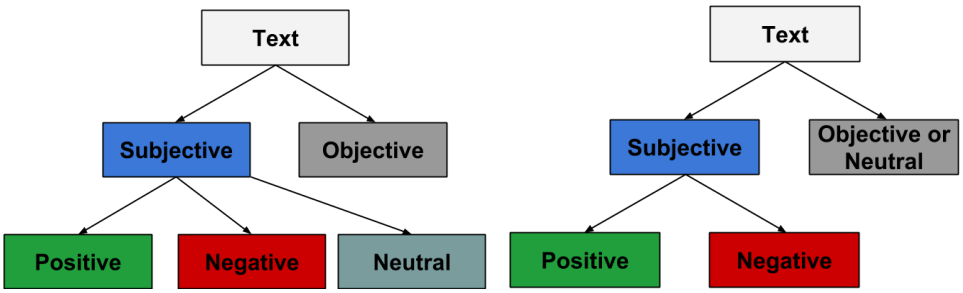

Figure 2.1: Taxonomy of sentiment analysis before and after simplification.

The main objective of message level sentiment analysis systems is to decide whether a message M conveys positive, negative or no sentiment (objective or neutral). In the case of conflict sentiment, the dominant sentiment is considered to be the sentiment of the message. For instance M_1 below expresses a positive sentiment, M_2 a negative one, while M_3 has no sentiment at all, thus being a neutral message.

M_1: GREAT GAME GIRLS!! On to districts Monday at Fox!! Thanks to the fans for coming out :)

M_2: Firework just came on my tv and I just broke down and sat and cried, I need help okay

M_3: Going to a bulls game with Aaliyah & hope next Thursday

SA of microblogging and social networks has focused on Twitter posts, which are known as tweets. Early work featured only polarity classification on the tweet level (Go et al., 2009). More recent work targets subjectivity detection as well (Barbosa and Feng, 2010). During the last two years two international challenges have had tasks focusing on

[3]The team that developed the system of this chapter comprises the author, Ion Androutsopoulos, John Pavlopoulos, and Makis Malakasiotis. In 2013, Nantia Makrynioti was also part of the team.

sentiment analysis of social networks. Task 2 of SEMEVAL-2013 (Wilson et al., 2013) consisted of two tasks focusing on different tweet levels, expression (subtask-A) and message (subtask-B) level. Various approaches were proposed by contestants. Mainly machine learning was utilized, mostly using either a Naïve Bayes classifier or a Support Vector Machine (SVM) with a linear kernel. The most common features were bag of words (BOW), lexicon based scores and part of speech (POS) based features. Task 9 of SEMEVAL-2014 (Rosenthal et al., 2014) was a rerun of 2013's task. The system described in this thesis participated in both these challenges achieving good results in 2013 and even better results in 2014.[4]

2.2 Datasets

Before we proceed with the system description, we briefly introduce the data used in this chapter. We have used the data of SEMEVAL-2013 Task 2 and SEMEVAL-2014 Task 9.

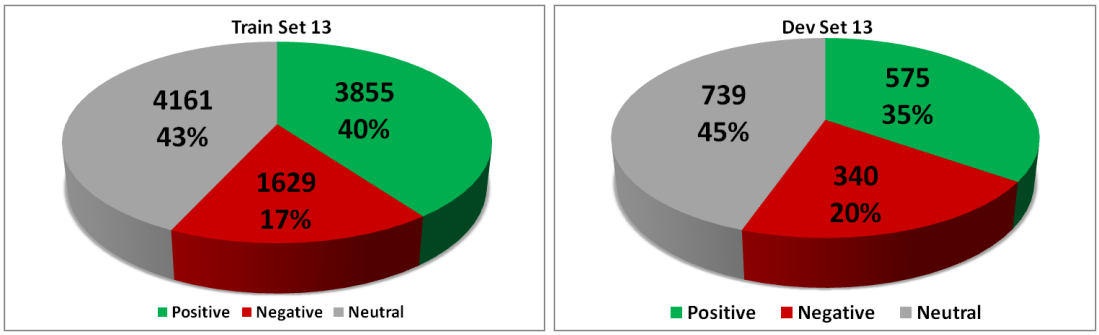

Figure 2.2: Train and development data class distribution in SEMEVAL-2013.

The training set of SEMEVAL-2013 consisted of a set of tweet IDs (each ID is unique and corresponds to only one tweet) instead of the original messages, along with their correct (human-annotated) sentiment labels (positive, negative, neutral). This method was chosen to address copyright concerns. The actual tweets were downloaded by each participant using the IDs through a Python script provided by the organisers. However,

[4]The system used in 2014's competition is an improved version of 2013's system.

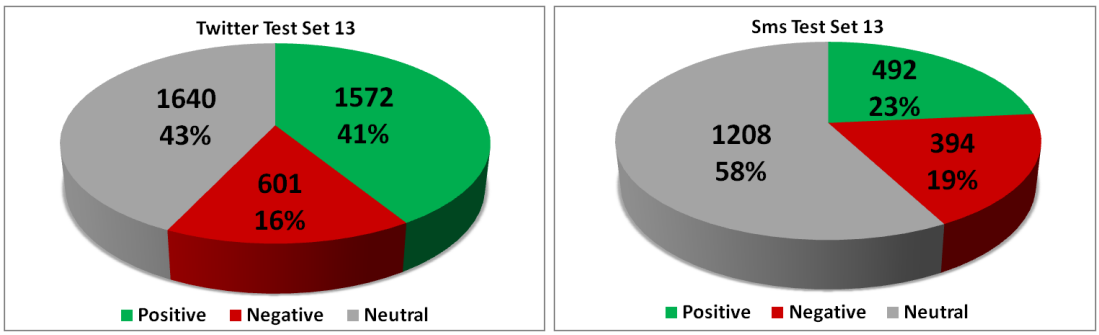

Figure 2.3: Test data class distribution in SEMEVAL-2013.

the main drawback of this approach is that different participants had slightly different versions of the training set, since tweets may often become unavailable due to a number of reasons such as the deletion of a tweet or the modification of a user's profile from public to private. The test sets were provided directly by the organisers via FTP, because if the participants downloaded different test sets then the evaluation would be difficult and the results not directly comparable. Two test sets were provided one containing tweets and one containing SMS messages. The SMS set was provided in order to measure the performance of the systems on messages of a different type than the one the systems had been trained on. It is worth noting that no SMS training and development data were provided.

A first analysis of the SEMEVAL-2013 data indicates that they suffer from class imbalance (Figure 2.2). In more detail, the training data we downloaded contained 8730 tweets from which 3280 were positive, 1289 were negative, and 739 were neutral. Moreover, the development data contained 1654 tweets; 575 positive, 340 negative, and 739 neutral. Similar problems were also observed in the test sets (Figure 2.3). Specifically, the tweets test data consisted of 3813 tweets (1572 positive, 601 negative, 1640 neutral), while the SMS test data consisted of 2094 messages (492 positive, 394 negative, 1208 neutral).

Concerning SemEval-2104, the training and development datasets were merged into one set in order to be used for training. The training data of 2014 also suffer from the class imbalance problem (Figure 2.4), since they are the union of 2013's train and

Test Set	Description	Positive	Negative	Neutral
LJ_{14}	1142 sentences from LIVEJOURNAL.	427	304	411
SMS_{13}	SMS test data from 2013.	492	394	1207
TW_{13}	Twitter test data from 2013.	1572	601	1640
TW_{14}	1853 new tweets.	982	202	669
$TWSARC_{14}$	86 tweets containing sarcasm.	33	40	13

Table 2.1: Test sets of 2014.

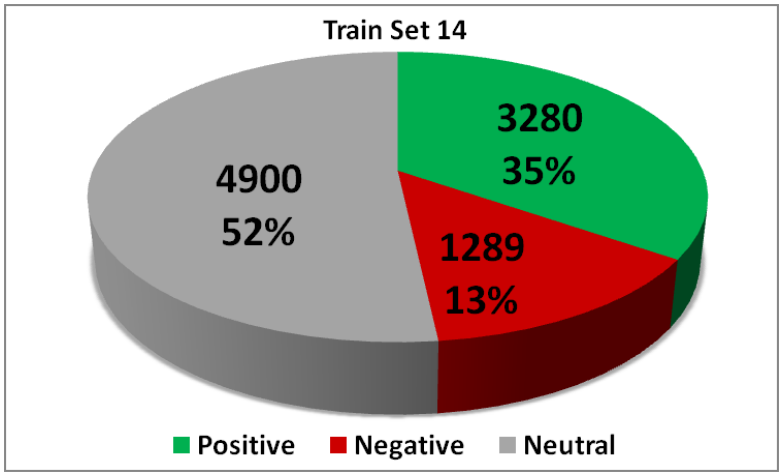

Figure 2.4: Train data class distribution in SEMEVAL-2014.

development set. Moreover, the test data of 2013 were provided as development data for 2014. The train set and the Twitter development set were distributed as tweet IDs and were downloaded by the participants using the Python script of 2013 due to Twitter's privacy policy, while the SMS set was available for direct download as it did not have privacy issues. The test set of 2014 consisted of 8987 messages (3506 positive, 1531 negative, 3940 neutral) and was downloaded by the participants via FTP. It consists of five subsets as shown in Table 2.1. Figure 2.5 shows the class distribution of the new test sets. It is worth noting that the new test sets have different a class distribution than the train set. In TW_{14} the positive class is the majority class. On the other hand in $TWSARC_{14}$ the negative class is the majority class, while LIVEJOURNAL has almost balanced class distribution. As shown in Table 2.1, TW_{13} and SMS_{13} were used both

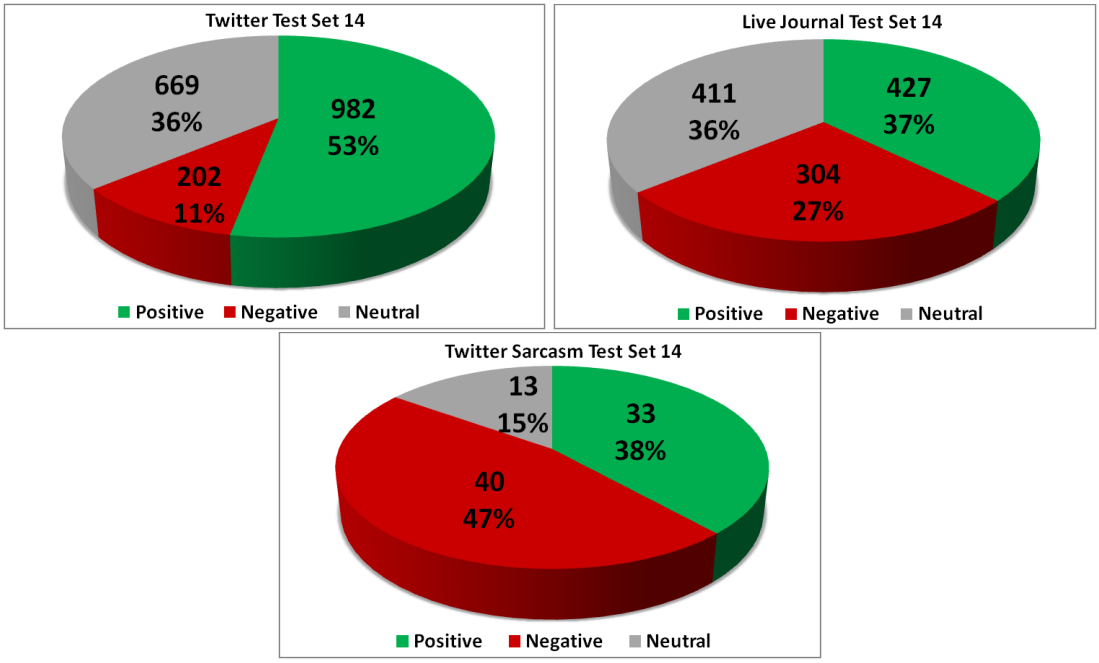

Figure 2.5: Test data class distribution in SEMEVAL-2014.

as development and test data in 2014. Possibly partly due to this, some systems suffered from overfitting and did not generalize well on other sets.

2.3 System architecture

Our system follows a two–stage approach. We have used the simplified taxonomy of classes of Figure 2.1. During the first stage we perform subjectivity detection, hence we detect whether a message expresses some sentiment or not. In the second stage, we perform polarity detection for the 'subjective' messages found in the first stage and classify them as 'positive' or 'negative' (Figure 2.6). Both stages utilize a Support Vector Machine (SVM) (Vapnik, 1998) classifier with a linear kernel.[5] Such a two–stage approach has also been suggested by Pang and Lee (2004) to improve sentiment classification of reviews by discarding objective sentences, by Wilson et al. (2005) for phrase-level sentiment analysis, and by Barbosa and Feng (2010) for sentiment analysis on Twitter messages. The main reason for choosing to use this architecture is that it

[5]We used the LIBLINEAR implementation (Fan et al., 2008).

helps addressing the class imbalance problem (Figures 2.2 and 2.4), since the classifier of each stage is trained on almost balanced data. It also allows us to focus on each stage separately. Especially, subjectivity detection can be useful for other tasks, such as opinion mining where we might want to detect subjective sentences in reviews that express an opinion for a particular product.

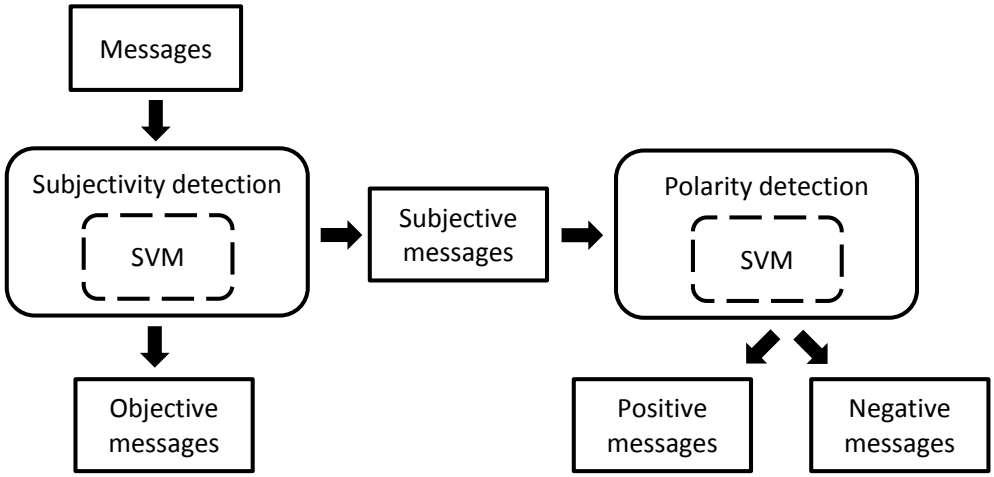

Figure 2.6: Our two–stage message-level sentiment classifier.

2.3.1 Data preprocessing

The preprocessing of the data is an important part of the system and can greatly influence its performance. For tokenization and POS tagging we have used a Twitter-specific tokenizer and part-of-speech (POS) tagger (Owoputi et al., 2013). For the preprocessing of tweets we use the following algorithm:

1. Each message is passed through the tokenizer and a list of tokens is produced.
2. The POS tag of each token is computed.
3. A slang dictionary[6] is utilized to replace any slang expression with the corresponding non-slang expression.

[6]See `http://www.noslang.com/dictionary/`.

4. Each token of the message is normalized using a simple text normalization algorithm:

   (a) We check if the token contains English characters. If not or if the token is an abbreviation, then we return the original token.

   (b) We replace special characters such as '@' with the corresponding character(s) of Table 2.2 and we reutilize the slang dictionary.

   (c) We squeeze characters that are repeated more than two times in a row (e.g., loooove) and we reutilize the slang dictionary.

   (d) If the token is not present in a general purpose English dictionary then we replace it with the most similar word of the dictionary.[7] To measure the similarity of words we employ Levenshtein edit distance, a trie data structure (De La Briandais, 1959) and dynamic programming to do the computation more efficiently (Karampatsis, 2012).

5. For each token that was replaced by the slang dictionary and/or the English dictionary, its POS tag is recomputed.

2.3.2 Sentiment lexicons preprocessing

Our system uses various lexicons which can be divided in two categories:

1. Lexicons that contained scores.

2. Lexicons that did not contain scores.

2.3.2.1 Sentiment lexicons with scores

HL (Hu and Liu, 2004): A list of 2006 positive and a list of 4783 negative opinion words for English. Positive words have a score of 1 and negative words a score

[7] We used the OPENOFFICE dictionary `https://www.openoffice.org/lingucomponent/download_dictionary.html`.

Original	Replacement
0	o
3	e
@	a
#	h
8	ate
4	for
!	i
$	s
1	i
2	to
5	s
7	t

Table 2.2: Character replacement mapping.

of -1. Both lists were merged into a subjectivity list where every word has a score of 1.

SENTIWORDNET lexicon with POS tags (Baccianella et al., 2010): Contains all WordNet synsets and assigns to each one three sentiment scores, positivity, negativity, objectivity.[8] We discarded every synset whose first sense's sum of positive and negative scores equals 0. For each word, we take the sentiment score of the first sense for the appropriate POS tag.

SENTIWORDNET (Baccianella et al., 2010): The same as SENTIWORDNET with POS tags, but we average the sentiment scores of all the word's POS tags.

AFINN (Nielsen, 2011): A list of 2477 English words and phrases, rated for valence

[8]In WordNet, which is a large lexical database of English, nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets), each expressing a distinct sense. Synsets are interlinked by means of conceptual-semantic and lexical relations.

with an integer in the range $[-5, 5]$.

NRC Emotion lexicon (Mohammad and Turney, 2013): This lexicon has annotations for 14,177 words. These annotations show whether a word is positive or negative and whether it has associations with eight basic emotions (joy, sadness, anger, fear, surprise, anticipation, trust, disgust). For each word, we compute three scores: the number of positive emotions (joy, surprise, anticipation, trust) associated with the word, plus one if the word is positive; the number of negative emotions (sadness, fear, anger, disgust), plus one if the word is negative; and finally a subjectivity score, which is the sum of our positive and negative scores. We use only the 6,464 words which have at least one non zero score.

NRC Hashtag lexicon (Mohammad et al., 2013): Contains three lists. One of 54,129 word unigrams, one of 316,531 word bigrams, and one of 480,000 pairs of unigrams and bigrams (i.e., unigram–unigram, unigram–bigram, bigram–unigram, or bigram–bigram pairs). For each entry of the lexicon there are three sentiment scores. The first score (a real number) is the association of the word with positive sentiment minus the association of the word with negative sentiment. For a word w this score is computed as $PMI(w, positive) - PMI(w, negative)$, where PMI stands for pointwise mutual information (Church and Hanks, 1990). A positive value indicates positive sentiment and a negative value indicates negative sentiment.[9] The second score is the number of times the term (or the pair) co-occurred with a positive marker (e.g., positive emoticon or hashtag). The third score is the number of times the term (or the pair) co-occurred with a negative marker. We have only used the first score.

NRC S140 lexicon (Mohammad et al., 2013): A list of words with associations to positive and negative sentiments. This lexicon uses identical format to the NRC

[9]The association of a word with the positive (or negative) sentiment was measured from 775K tweets by the pointwise mutual information score of the word for each category. The tweets were annotated using a list of 32 positive and 36 negative hashtagged seed words, e.g., #good, #bad, etc. The same seed words were used to collect the 775K tweets.

Hashtag lexicon, but it was created from 1,6 million tweets, which were labeled as positive or negative according to the emoticons that they contained.

2.3.2.2 Sentiment lexicons without scores

Three lexicons created from the SEMEVAL-2013 training data by (Malakasiotis et al., 2013): To create these lexicons we selected the 100 most important words per class from the training set by Chi Squared (χ^2) (Liu and Setiono, 1995) feature selection. We manually removed tokens such as days, months and years. This resulted in a lexicon of 94 terms for the positive class, a lexicon of 96 terms for the negative class, and a lexicon of 94 terms for the neutral class.

MPQA (Wilson et al., 2005): A list of 8222 words. For each word there are various annotations. The prior polarity of the word (positive, negative or neutral), its subjectivity intensity (weak or strong) and its POS tag. A word/expression is considered 'strong' subjective if it expresses sentiment in most contexts, otherwise it is considered 'weak' subjective (sometimes it is subjective). We split the lexicon into various sub–lexicons. Firstly, we split the lexicon with respect to the subjectivity intensity of the contained terms, resulting in two sublexicons, one containing strong and one containing weak subjective words. Then, we further divided these sublexicons into smaller ones with respect to the prior polarity of their words. This process leads to eight MPQA-based (sub)lexicons:

S_+ : Contains strong subjective words with positive prior polarity (e.g., 'charismatic').

S_- : Contains strong subjective words with negative prior polarity (e.g., 'abase').

$S_\pm$: Contains strong subjective words with either positive or negative prior polarity.

S_0 : Contains strong subjective words with neutral prior polarity (e.g., 'disposition').

W_+ : Contains weak subjective words with positive prior polarity (e.g., 'drive').

W_- : Contains weak subjective words with negative prior polarity (e.g., 'dusty').

$W_\pm$: Contains weak subjective words with either positive or negative prior polarity.

W_0 : Contains weak subjective words with neutral prior polarity (e.g., 'duty').

2.3.3 Subjectivity detection

In this first stage of the system we want to detect subjective messages (i.e., messages with sentiment) and discard objective or neutral sentences (i.e., messages without sentimen). We will next discuss the features employed in stage 1. The discussion will focus on 2014's system as it is the evolution of 2013's system. At the end of this Section we will briefly discuss the differences of the two systems. Our system employs several types of features based on morphological attributes of the messages, POS tags, and the sentiment lexicons of Section 2.3.2.[10]

2.3.3.1 Morphological features

- A Boolean feature indicating the existence (or absence, before squeezing) of elongated tokens (e.g., 'baaad'), in the message being classified.
- The number of elongated tokens in the message.
- The existence (or absence) of date expressions in the message (Boolean feature).
- The existence of time expressions (Boolean feature).
- The number of tokens of the message that are fully capitalized (i.e., contain only upper case letters).
- The number of tokens that are partially capitalized (i.e., contain both upper and lower case letters).
- The number of tokens that start with an upper case letter.

[10] All the features are normalized to $[-1, 1]$.

- The number of exclamation marks in the message.
- The number of question marks.
- The sum of exclamation and question marks.
- The number of tokens containing only exclamation marks.
- The number of tokens containing only question marks.
- The number of tokens containing only exclamation or question marks.
- The number of tokens containing only ellipsis (...).
- The existence of a subjective (i.e., positive or negative) emoticon at the message's end.
- The existence of an ellipsis and a link (URL) at the message's end. News tweets, which are often objective, often contain links of this form.
- The existence of an exclamation mark at the message's end.
- The existence of a question mark at the message's end.
- The existence of a question or an exclamation mark at the message's end.
- The existence of slang, as detected by using the slang dictionary (Section 2.3.1).

2.3.3.2 POS based features

- The number of adjectives in the message being classified.
- The number of adverbs.
- The number of interjections (e.g., 'hi', 'bye', 'wow', etc.).
- The number of verbs.
- The number of nouns.
- The number of proper nouns.

– The number of URLs.

– The number of subjective emoticons.

2.3.3.3 POS bigrams features

– The average F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.

– The maximum F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.

– The minimum F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.

For a POS-tag bigram b and a class c, F_1 is calculated over all the training messages as:

$$F_1(b,c) = \frac{2 \cdot Pre(b,c) \cdot Rec(b,c)}{Pre(b,c) + Rec(b,c)} \tag{2.1}$$

where:

$$Pre(b,c) = \frac{\#\text{messages of } c \text{ containing } b}{\#\text{messages containing } b} \tag{2.2}$$

$$Rec(b,c) = \frac{\#\text{messages of } c \text{ containing } b}{\#\text{messages of } c} \tag{2.3}$$

2.3.3.4 Sentiment lexicon based features

For each subjectivity lexicon of Section 2.3.2 we use the following seven features based on the scores provided by the lexicon for each word present in the message:[11]

– Sum of the scores.

– Maximum of the scores.

– Minimum of the scores.

[11] We removed from SENTIWORDNET any instances having positive and negative scores equal to zero. For the lexicons that do not provide scores, we assume that each word's score is equal to 1.

– Average of the scores.

– The count of the words of the message that appear in the lexicon.

– The score of the last word of the message that appears in the lexicon.

– The score of the last word of the message.

If a word does not appear in the lexicon, it is assigned a score of 0 and it is not considered in the calculation of the average, maximum, minimum and count scores. We use the absolute values of the scores as both positive and negative scores are considered subjective.

We also created features based on the precision and F_1 scores of the words of MPQA and the words of the lexicons generated from the training data (Malakasiotis et al., 2013). For each word w of each lexicon, we calculate the precision ($Pre(w,c)$), recall ($Rec(w,c)$) and F_1 ($F_1(w,c)$) of w with respect to class c (Equations 2.4, 2.5, and 2.6). The scores are computed on the training data.

$$Pre(w,c) = \frac{\text{\#messages that contain word } w \text{ and belong in class } c}{\text{\#messages that contain word } w} \tag{2.4}$$

$$Rec(w,c) = \frac{\text{\#messages that contain word } w \text{ and belong in class } c}{\text{\#messages that belong in class } c} \tag{2.5}$$

$$F_1(w,c) = \frac{2 \cdot P(w,c) \cdot R(w,c)}{P(w,c) + R(w,c)} \tag{2.6}$$

Having assigned a precision and F_1 score to each word of each lexicon (MPQA and lexicons generated from training data), we then use the following features:

– The average precision score of the words of the message for the subjective class.

– The maximum precision score of the words of the message for the subjective class.

– The minimum precision score of the words of the message for the subjective class.

– The average F_1 score of the words of the message for the subjective class.

- The maximum F_1 score of the words of the message for the subjective class.
- The minimum F_1 score of the words of the message for the subjective class.
- The average precision score of the words of the message for the neutral class.
- The maximum precision score of the words of the message for the neutral class.
- The minimum precision score of the words of the message for the neutral class.
- The average F_1 score of the words of the message for the neutral class.
- The maximum F_1 score of the words of the message for the neutral class.
- The minimum F_1 score of the words of the message for the neutral class.

2.3.3.5 Miscellaneous features

Negation: Negation is a good subjectivity indicator. We use one feature indicating the existence of negation in the message being classified, via a list of English words related to negation (e.g., 'don't').

Negation preceding lexicons: We also use two more features indicating the existence of negation in the message before (up to a distance of 5 tokens) words from lexicons $S_{\pm}$, and $W_{\pm}$. We have not implemented these features for other lexicons, but they might be a good addition to the system.

Carnegie Mellon University's Twitter clusters (Owoputi et al., 2013): CMU released a dataset of 938 clusters containing words coming from tweets. Words of the same clusters share similar attributes (e.g., they may be near-synonyms, or they may be used in similar contexts). For instance soo, sooo, soooo, etc., belong to the same cluster. We exploit these clusters by adding 938 Boolean features, each one indicating if any of the message's words appear (or not) in the corresponding cluster.

2.3.4 Polarity detection

During this stage our system classifies the subjective messages of stage 1 as positive or negative. We use similar features to those of stage 1 as described below.

2.3.4.1 Morphological features

- A Boolean feature indicating the existence (or absence, before squeezing) of elongated tokens (e.g., 'baaad'), in the message being classified.
- The number of elongated tokens in the message.
- The existence (or absence) of date expressions in the message (Boolean feature).
- The existence of time expressions (Boolean feature).
- The number of tokens of the message that are fully capitalized (i.e., contain only upper case letters).
- The number of tokens that are partially capitalized (i.e., contain both upper and lower case letters).
- The number of tokens that start with an upper case letter.
- The number of exclamation marks in the message.
- The number of question marks.
- The sum of exclamation and question marks.
- The number of tokens containing only exclamation marks.
- The number of tokens containing only question marks.
- The number of tokens containing only exclamation or question marks.
- The number of tokens containing only ellipsis (...).
- The existence of a positive emoticon at the message's end.

- The existence of a negative emoticon at the message's end.
- The existence of an ellipsis and a link (URL) at the message's end. News tweets, which are often objective, often contain links of this form.
- The existence of an exclamation mark at the message's end.
- The existence of a question mark at the message's end.
- The existence of a question or an exclamation mark at the message's end.
- The existence of slang, as detected by using the slang dictionary (Section 2.3.1).

2.3.4.2 POS based features

- The number of adjectives in the message being classified.
- The number of adverbs.
- The number of interjections (e.g., 'hi', 'bye', 'wow', etc.).
- The number of verbs.
- The number of nouns.
- The number of proper nouns.
- The number of URLs.
- The number of positive emoticons.
- The number of negative emoticons.

2.3.4.3 POS bigrams features

- The average precision score of the message's POS-tag bigrams for the positive and negative classes.
- The maximum precision score of the message's POS-tag bigrams for the positive and negative classes.

– The minimum precision score of the message's POS-tag bigrams for the positive and negative classes.

– The average F_1 score of the message's POS-tag bigrams for the positive and negative classes.

– The maximum F_1 score of the message's POS-tag bigrams for the positive and negative classes.

– The minimum F_1 score of the message's POS-tag bigrams for the positive and negative classes.

For the definition of the F_1 score used see Equation 2.1.

2.3.4.4 Sentiment lexicon based features

For each polarity lexicon of Section 2.3.2 we use the following seven features based on the scores provided by the lexicon for each word present in the message.[12]

– Sum of the scores.

– Maximum of the scores.

– Minimum of the scores.

– Average of the scores.

– The count of the words of the message that appear in the lexicon.

– The score of the last word of the message that appears in the lexicon.

– The score of the last word of the message.

[12]We removed from SENTIWORDNET any instances having positive and negative scores equal to zero. Moreover, the MPQA lexicon does not provide scores, so, for each positive word in the lexicon we assume a score equal to 1 and for each negative a score equal to -1.

If a word does not appear in the lexicon, it is assigned a score of 0 and it is not considered in the calculation of the average, maximum, minimum and count scores.

For each lexicon of Section 2.3.1 we use seven different features based on the scores provided by the lexicon for each word present in the message.

Similarly to Stage 1, we created features based on the precision and F_1 scores of the words of MPQA and the words of the lexicons generated from the training data (Malakasiotis et al., 2013). For each word w of each lexicon, we calculate the precision ($Pre(w,c)$), recall ($Rec(w,c)$) and F_1 ($F_1(w,c)$) of w with respect to class c (Equations 2.4, 2.5 and 2.6). Having assigned a precision and F_1 score to each word of each lexicon (MPQA and lexicons generated from training data), we then use the following features:

– The average precision score of the words of the message for the positive class.

– The maximum precision score of the words of the message for the positive classategory.

– The minimum precision score of the words of the message for the positive class.

– The average F_1 score of the words of the message for the positive class.

– The maximum F_1 score of the words of the message for the positive class.

– The minimum F_1 score of the words of the message for the positive class.

– The average precision score of the words of the message for the negative class.

– The maximum precision score of the words of the message for the negative class.

– The minimum precision score of the words of the message for the negative class.

– The average F_1 score of the words of the message for the negative class.

– The maximum F_1 score of the words of the message for the negative class.

– The minimum F_1 score of the words of the message for the negative class.

2.3.4.5 Miscellaneous features

Negation. Negation may change the polarity of a message (i.e., 'He is not a bad singer'). We use the same list of words related to negation that we used in Stage 1.

Negation preceding lexicons: We also use 4 more features indicating the existence of negation in the message before (up to a distance of 5 tokens) words from lexicons S_+, S_-, W_+ and W_-. We have not implemented these features for other lexicons, but they might be a decent addition to the system.

Carnegie Mellon University's Twitter clusters (Owoputi et al., 2013): These features are similar to those employed in Stage 1.

2.3.5 Feature selection

To allow our model to better scale on unseen data we have performed feature selection. More specifically, we first merged the training and development data of SEMEVAL-2013 Task 2. Then, we ranked the features with respect to their information gain (Quinlan, 1986). To obtain the best set of features, we started with a set containing the top 50 features and we kept adding batches of 50 features until we had added all of them. At each step we evaluated the corresponding feature set on the TW_{13} and SMS_{13} datasets. To evaluate the performance of the system we used the weighted average $F_1(\pm)$ (Equation 2.7) of the TW_{13} and SMS_{13} datasets; the $F_1(\pm)$ of each dataset is weighted according to the size of each dataset. For simplicity we have chosen to optimize each stage separately, however it could be a good improvement to optimize the two stages simultaneously. We eventually chose the feature set with the best performance for each stage. This resulted in a system which used the top 900 features for Stage 1 and the top 1150 features for Stage 2.

2.3.6 Differences between our systems of SEMEVAL-2013 and SEMEVAL-2014

The system described in the previous sections competed in the challenge of SEMEVAL-2014. We will refer to this system as AUEB_{14}. In the challenge of SEMEVAL-2013 we competed with a simpler version of AUEB_{14}. We will refer to this system as AUEB_{13}. Below we describe the changes that made to AUEB_{13} to create AUEB_{14}. AUEB_{13} was improved in three ways:

1. We added more lexicons. In detail AUEB_{13} does not use HL, SENTIWORDNET, AFINN, NRC Emotion, NRC Hashtag, and NRC S140.

2. We focused on each stage separately. The features used in AUEB_{13} were based on three classes, positive, negative, and neutral (Malakasiotis et al., 2013), while in 2014 our features were computed seperately for the two classes of the first stage (i.e., subjective vs. neutral) and the two classes of the second stage (i.e., positive vs. negative).

3. Bag of words features (i.e., features showing the presense of specific words or terms) were removed, since preliminary feature selection indicated that they did not add much to the other features.

2.4 Evaluation measures

To evaluate our system we used the official measure of the challenge. This is the average F_1 score of the positive and negative classes (Equations 2.7 - 2.10). [13]

$$F_1(\pm) = \frac{F_1(+) + F_1(-)}{2} \tag{2.7}$$

[13] As noted by the SEMEVAL organisers, this measure does not make the task binary. Rather, the neutral class is considered less important than (and is being indirectly evaluated through) the positive and negative classes.

$$F_1(c) = \frac{2 \cdot P(c) \cdot R(c)}{P(c) + R(c)} \tag{2.8}$$

$$Pre(c) = \frac{\text{\#messages that belong in class } c \text{ and were classified as c}}{\text{\#messages that were classified as c}} \tag{2.9}$$

$$Rec(c) = \frac{\text{\#messages that belong in class } c \text{ and were classified as c}}{\text{\#messages that belong in class } c} \tag{2.10}$$

2.5 Experimental results

2.5.1 SEMEVAL-2013

Table 2.3 lists the $F_1(\pm)$ scores of our system along with the scores of a majority baseline, the median score of the participating systems, and the best $F_1(\pm)$ in each test set of SEMEVAL-2013. On TW_{13}, our 2013 system achieved an $F_1(\pm)$ score of 58.91% and it was ranked 17th/51 systems (i.e., both constrained and unconstrained). The best system achieved a score of 69.02% and a majority baseline released by the organizers achieved only 29.19%. On SMS_{13}, our 2013 system achieved an $F_1(\pm)$ score of 55.28% and it was ranked 11th/45 systems. The best system achieved 68.46% and the baseline 19.03%.

Test Set	**AUEB**	**Baseline**	**Median**	**Best Reported Score**	**AUEB Ranking**
TW_{13}	58.91	29.19	54.56	69.02	17/51
SMS_{13}	55.28	19.03	51.08	68.46	11/45

Table 2.3: $F_1(\pm)$ scores and ranking per dataset of our system in SEMEVAL-2013.

2.5.2 SEMEVAL-2014

Table 2.4 illustrates the $F_1(\pm)$ score achieved by our 2014 system for each evaluation dataset, along with the median score of the participating systems, the best score and the

rankings of our system.[14] We have also computed AVG_{all} which is the macro $F_1(\pm)$ (average $F_1(\pm)$ with equal weighting) across the five datasets and AVG_{14} which is the macro $F_1(\pm)$ across the 2014 test datasets (LJ_{14}, TW_{14} and $TWSARC_{14}$). Our system always beats the median system. We ranked 6th by AVG_{all} and 5th by AVG_{14} among the 50 participating systems of the 2014 competition. One should note that our best results were achieved on the new test sets (LJ_{14}, TW_{14}, $TWSARC_{14}$) meaning that our system has a good generalization ability. Recall that in 2014 the test data comprised tweets (TW_{14}), tweets including sarcasm ($TWSARC_{14}$), tweets from 2013 (TW_{13}), SMS messages from 2013 (SMS_{13}), and sentences from LIVEJOURNAL (LJ_{14}).

Test Set	**AUEB**	**Median**	**Best Reported Score**	**AUEB Ranking**
LJ_{14}	70.75	65.48	74.84	9/50
SMS_{13} *	64.32	57.53	70.28	8/50
TW_{13} *	63.92	62.88	72.12	21/50
TW_{14}	66.38	63.03	70.96	14/50
$TWSARC_{14}$	56.16	45.77	58.16	4/50
AVG_{all}	64.31	56.56	68.78	6/50
AVG_{14}	64.43	57.97	67.62	5/50

Table 2.4: $F_1(\pm)$ scores and ranking per dataset of our system in SEMEVAL-2014.

2.5.3 Ceiling analysis

It is often very useful to individually evaluate each stage of the system, in order to decide which stage one should try to improve. In more detail, when a pipeline approach is used, ceiling analysis is employed to evaluate the impact of each module to the overall performance of the system. This can help us decide which module is worthy of investing our time to improve. For this analysis we will use error rate instead of $F_1(\pm)$, because in the case of a hypothetical perfect polarity system there is no way to know into which category (positive or negative) the system would classify objective instances, thus being

[14] Stars indicate datasets that were also used as development data in SEMEVAL-2014 feature selection.

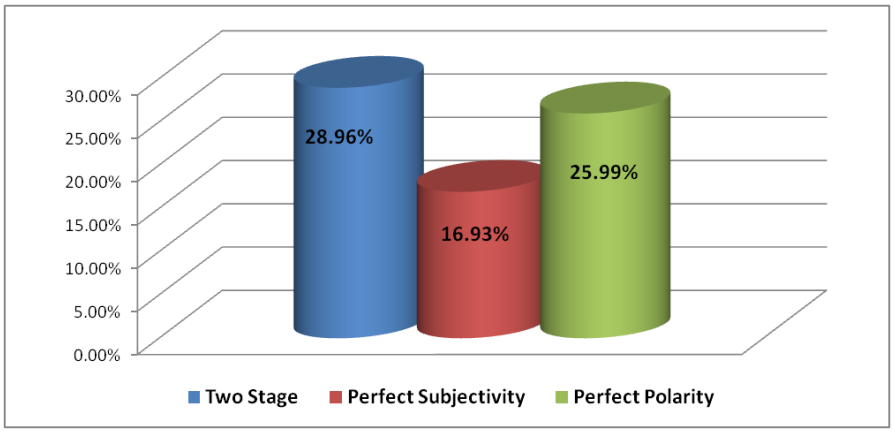

Figure 2.7: Ceiling analysis of AUEB_{14} trained on 2013's train set and tested on 2013's development set.

impossible to compute the F_1 of the positive and negative categories. To perform ceiling analysis we have to evaluate each stage assuming that the preceding stages are perfect. In our case Stage 1 does not have a preceding stage. For Stage 2 we assume a perfect input coming from Stage 1, i.e., all the truly subjective messages are passed to Stage 2. For the analysis we have trained Stage 2 of our system (polarity classification) on 2013's train set and tested it on 2013's development set. Figure 2.7 illustrates that even if we improve Stage 2 and make it perfect the total error of the system will only drop from 28.96% to 25.99%. On the other hand if we invest our time in creating a perfect Stage 1 system then the system's total error will drop dramatically to only 16.93%. Hence, it is by far a better choice to try to improve Stage 1.

2.6 Conclusions and future work

In this chapter, we presented the system we designed and implemented for message-level sentiment estimation. Our system participated, and achieved good ranks, in the Message Polarity Classification subtask of the Sentiment Analysis in Twitter Task of SEMEVAL-2013 and even better ranks in 2014. We proposed a two–stage pipeline approach, which first detects sentiment and then decides about the polarity of that sentiment, using two separate SVM classifiers. The results indicate that our system handles

well the class imbalance problem and has a good generalization ability over different types of messages (tweets, SMSs, blog posts from LIVEJOURNAL).

As shown in Section 2.5.3, it is a very good option to invest time into improving Stage 1 of the system. A first step in this direction might be to include as a feature the output of a vagueness classifier (Alexopoulos and Pavlopoulos, 2014), the idea being that vagueness correlates with subjectivity. We have already done some preliminary experiments in order to study the corellation of vagueness and subjectivity. However, since additional experiments are needed to study this correlation, we intend to study this issue in future work. Another, promising feature would be to check the existence of certain types of named entities such as products, footbal or basketball clubs etc. since posts containing them are often subjective.

Chapter 3

A Sentiment Analysis System For Greek Social Network Messages

3.1 Introduction

Although sentiment analysis in Twitter and social networks in general has been a popular topic recently, there is little or no work that has studied this issue for the Greek language. For this reason we created a sentiment analysis system which focuses on posts written in the Greek language. As a case study we have used Twitter. The data we have used, the methodology applied, the experimental results of the system and possible improvements are discussed in this chapter.

3.2 Datasets

Table 3.1 illustrates the data used in this chapter. They were collected and annotated by the Greek company Qualia.[1] We would like to thank them and express our gratitude for their help. We have split the data in three sets. Namely, the training, the development, and the test set, consisting of 70%, 10% and 20% of the original data, respectively.

As shown in Figure 3.1, the data suffer from a class imbalance problem. Specif-

[1] See `http://www.qualia.gr/`.

Set	Description	Positive	Negative	Neutral
Train	Greek tweets train data.	2069	1922	4527
Development	Greek tweets devlopment data.	295	274	646
Test	Greek tweets test data.	592	550	1295

Table 3.1: Greek data details.

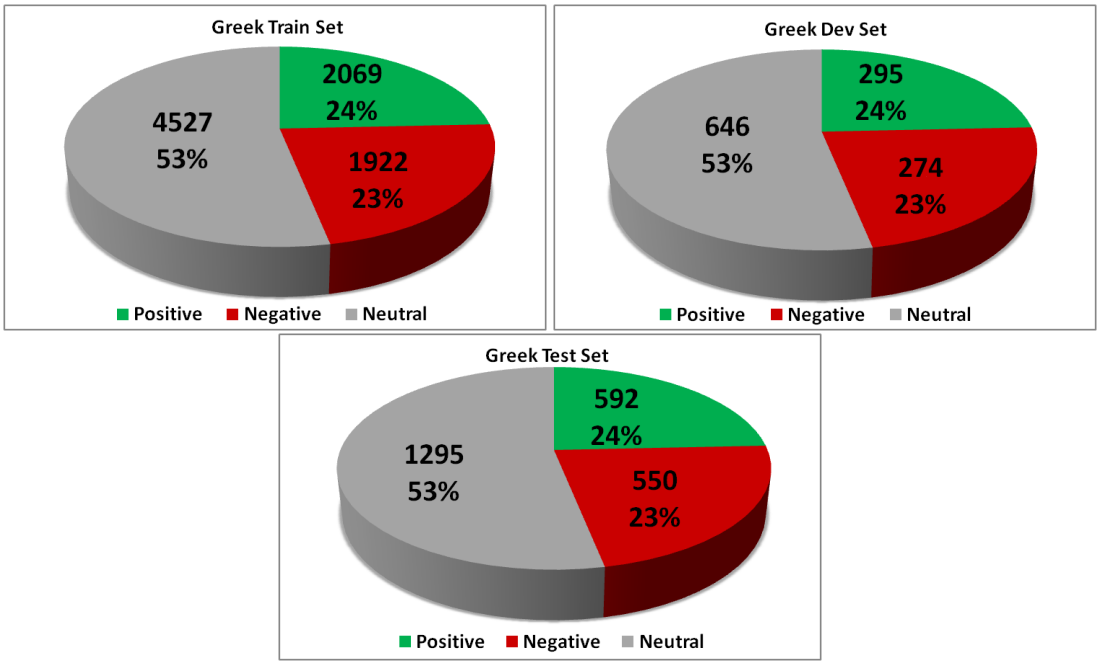

Figure 3.1: Train, development, and test data class distribution .

ically, the neutral class greatly outnumbers the other two. Recall that a similar class imbalance problem was also present in the data of SemEval (Section 2.2). Using, therefore, our two–stage approach (Figure 2.6) alleviates the problem since we have to deal with two balanced classification problems.

3.3 System architecture

3.3.1 Data preprocessing

Preprocessing can considerably affect the performance of the system. A very important problem we had to address is the lack of sufficient tools for the preprocessing of

Greek social network posts. At first this might not seem as a major issue. However, it has been shown that the performance of state-of-the-art NLP tools such as POS taggers drops dramatically when they are applied to Twitter data (Alan et al., 2011). For instance the accuracy of the Stanford POS tagger (Toutanova et al.,2003) drops from about 0.97 on news to 0.80 on tweets. Due to time constraints we chose not to create such tools from scratch, but to modify and adapt existing ones.

3.3.1.1 Greeklish conversion

It is very common for Greek users of social networks to write their posts in neither Greek nor English. Instead Greeklish is used. Greeklish (also known as Grenglish), is a transliteration of Greek using the Latin alphabet (i.e., users of Greeklish write words of the Greek language using the Latin alphabet). Writting words of a language using the alphabet of another one might seem realy confusing but it was invented by Greek Internet users because older operating systems did not support the Greek language in order to fulfill their need to write in Greek. A major issue with Greeklish is the lack of a standard concerning its use. This results in the existence of many different forms of Greeklish, thus making difficult their automatic conversion to Greek. To address the existence of tweets in Greeklish we have used the free online service "Greeklish-to-Greek",[2] which is based on an automatic Greeklish to Greek transliteration system (Chalamandaris et al., 2006) developed by the Institute of Language and Speech Processing.[3]

3.3.1.2 Tokenization

For tokenization we extended the Twitter-specific tokenizer and (POS) tagger (Owoputi et al., 2013) that was introduced in Section 2.3.1 to allow it to identify Greek characters and words.[4] This tokenizer is based on regular expressions patterns. To support the Greek language we modified several of these patterns in order to support Greek characters. Example patterns that we had to modify are URLs and acronyms (e.g., O.H.E,

[2]See `http://services.innoetics.com/greeklish/`.

[3]See `http://www.ilsp.gr/`.

[4]The tool also has a POS tagging component, but we did not use it.

Δ.E.H.). The result of these modifications is a Twitter specific tokenizer for both English and Greek, which can tokenize both English and Greek messages, but also mixed tweets such as tweets written in Greeklish or tweets containing both English and Greek words.

3.3.1.3 POS tagging

For POS tagging we have used the Greek POS tagger developed by Koleli (2011). However, this tagger does not support Twitter or social network specific tags such as usernames, hashtags and URLs. To address this problem we identify these tags using appropriate regular expressions. To perform POS tagging on a tweet we apply the following steps:

- We tokenize the tweet using our extended Twitter specific tokenizer.
- We identify all categories of tags that are not supported by the tagger via regular expressions.
- We use the identified tags as delimiters to split the text in parts.
- We normalize each part of the text the using the text normalization algorithm presented in Section 3.3.1.5.
- For each part, we use the Greek POS tagger to tag its tokens.

3.3.1.4 Stemming

According to Manning et al. (2008) stemming is the reduction of inflectional forms and sometimes derivationally related forms of a word to a common base form (stem). According to the results of the European CLEF evaluations, stemming has been especially helpful for languages with rich morphology, such as Greek, Finnish, Swedish and French. We used stemming in order to be able to match words of the sentiment lexicons regardless of their case or gender. For instance, a sentiment lexicon may contain the masculine gender of an adjective but not the feminine one. Thus, although both should

have the same sentiment score, only the masculine genre would participate in the calculation of features. To compensate with this problem we have used stemming and more specifically the Greek stemmer of Ntais (2006) which is included in Lucene.[5]

3.3.1.5 Text normalization

Before we proceed with the calculation of features for each tweet we normalize its text to correct mispelled words. The text normalization algorithm used is very similar to that of Section 2.3.1. We normalize the text of each tweet by replacing every token (excluding punctuation, abreviations etc.). However, the existence of tokens from both languages (Greek and English) in some tweets requires some changes in the original algorithm:

1. We use a slang dictionary[6] to replace any slang expression with the corresponding non-slang expression.

2. We check if the token contains any Greek characters. If no Greek characters are present then we use the English normalization algorithm (Section 2.3.1).

3. For each token found containing Greek characters if there is one or more 'ς' (end-of-word sigma) that are not the last character, then we replace them with 'σ' and if there is a 'σ' as the last character, then we replace it with 'ς'.

4. We replace special characters such as '@' with the corresponding character(s) of Table 3.2.

5. We squeeze characters that are repeated more than two times in a row (e.g., όόόχι).

6. We add all the possible accents (each one separately) to tokens that do not have any accent, and check if the accented token exists in the lexicon. If it exists, the initial token is replaced by the accented one and we skip the next step.

[5] http://lucene.apache.org/

[6] See http://www.noslang.com/dictionary/.

Original	Replacement
0	ο
3	ε
@	α
#	η
8	θ
4	α
!	ι
$	ς
1	ι
9	θ
5	ς
7	τ

Table 3.2: Character replacement mapping.

7. The text of each message is normalized by replacing every token (excluding punctuation, abbreviations etc.) that is not present in a general purpose Greek dictionary with the most similar word of the dictionary.[7] To measure the similarity of words we employ edit distance, a trie data structure (De La Briandais, 1959) and dynamic programming to do the computation more efficiently (Karampatsis, 2012).

3.3.2 Sentiment lexicons preprocessing

Similar to our English system, the Greek system uses various lexicons which can be divided in two categories:

1. Lexicons that contained scores.

[7] We used the Greek version of the OPENOFFICE dictionary `https://www.openoffice.org/lingucomponent/download_dictionary.html`.

2. Lexicons that did not contained scores.

3.3.2.1 Sentiment lexicons with scores

Qualia Sentiment: A list of 1363 Greek words, each annotated with an integer in the range $[-5,5]$ representing its sentiment score. This lexicon was created by Qualia.

Qualia Sentiment Stems: A list of 1363 stems of Greek words (i.e., we use the stems of the words of Qualia Sentiment), each annotated with an integer in the range $[-5,5]$ representing its sentiment score.

3.3.2.2 Sentiment lexicons without scores

Three lexicons created from the Greek training data similar to those of the English system: To create these lexicons we selected the 100 most important words per class from the training set by Chi Squared feature selection.

Qualia Mood: A list of 1953 words created by Qualia. Each word is annotated with its prior polarity (positive, negative, or neutral). We divided this lexicon into four sublexicons in a similar way to MPQA (Section 2.3.2.2):

Q_+ : Contains words with positive prior polarity.

Q_- : Contains words with negative prior polarity.

$Q_\pm$: Contains words with either positive or negative prior polarity.

Q_0 : Contains words with neutral prior polarity.

Qualia Mood Stems: A list of 1953 stems of words created by Qualia (i.e., the stems of the words of Qualia Mood are used). Preprocessing is identical to that used for Qualia Mood.

3.3.3 Subjectivity detection

Similarly to our English system, our Greek system employs several types of features based on morphological attributes of the messages, POS tags, and the sentiment lexicons of Section 3.3.2.

3.3.3.1 Morphological features

- A Boolean feature indicating the existence (or absence, before squeezing) of elongated tokens (e.g., 'καλόόόό') in the message being classified.
- The number of elongated tokens in the message.
- The existence (or absence) of date expressions in the message (Boolean feature).
- The existence of time expressions (Boolean feature).
- The number of tokens of the message that are fully capitalized (i.e., contain only upper case letters).
- The number of tokens that are partially capitalized (i.e., contain both upper and lower case letters).
- The number of tokens that start with an upper case letter.
- The number of exclamation marks in the message.
- The number of question marks.
- The sum of exclamation and question marks.
- The number of tokens containing only exclamation marks.
- The number of tokens containing only question marks.
- The number of tokens containing only exclamation or question marks.
- The number of tokens containing only ellipsis (...).

- The existence of a subjective (i.e., positive or negative) emoticon at the message's end.
- The existence of an ellipsis and a link (URL) at the message's end. News tweets, which are often objective, often contain links of this form.
- The existence of an exclamation mark at the message's end.
- The existence of a question mark at the message's end.
- The existence of a question or an exclamation mark at the message's end.
- The existence of slang, as detected by using the slang dictionary (Section 3.3.1).

3.3.3.2 POS based features

- The number of adjectives in the message being classified.
- The number of adverbs.
- The number of verbs.
- The number of nouns.
- The number of pronouns.
- The number of URLs.
- The number of subjective emoticons.

3.3.3.3 POS bigrams features

- The average F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.
- The maximum F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.

– The minimum F_1 score of the message's POS-tag bigrams for the subjective and neutral classes.

For the definition of F_1 see Equations 2.1-2.3 in Section 2.3.3.3.

3.3.3.4 Sentiment lexicon based features

For the subjectivity lexicon of Section 3.3.2.1 we use the following seven features based on the scores provided by the lexicon for each word present in the message:[8]

– Sum of the scores.

– Maximum of the scores.

– Minimum of the scores.

– Average of the scores.

– The count of the words of the message that appear in the lexicon.

– The score of the last word of the message that appears in the lexicon.

– The score of the last word of the message.

If a word does not appear in the lexicon, it is assigned a score of 0 and it is not considered in the calculation of the average, maximum, minimum and count scores.

We also created features based on the precision and F_1 scores of the words of Qualia Mood, the stems Qualia Mood Stem, and the words of the lexicons generated from the training data. These features are similar to the ones used in the English system. For the definition of the precision and F_1 scores see Equations 2.4-2.6 in Section 2.3.3.4.

3.3.3.5 Miscellaneous features

Negation: Negation is a good subjectivity indicator. We use one feature indicating the existence of negation, in the message being classified, via a list of Greek words related to negation (e.g., 'δεν').

[8]For the lexicon containing stems of words we have used the stems of the tweets' tokens to calculate the scores.

3.3.3.6 English subjectivity features

Using Google translate[9] we translated the normalized text of each message from Greek to English. Then, we used our English system to calculate the 900 features that we use for subjectivity detection in the English system. Finally, we append these features to the feature vector of our Greek system.

3.3.4 Polarity detection

3.3.4.1 Morphological features

- A Boolean feature indicating the existence (or absence, before squeezing) of elongated tokens (e.g., 'καλόόόό'), in the message being classified.
- The number of elongated tokens in the message.
- The existence (or absence) of date expressions in the message (Boolean feature).
- The existence of time expressions (Boolean feature).
- The number of tokens of the message that are fully capitalized (i.e., contain only upper case letters).
- The number of tokens that are partially capitalized (i.e., contain both upper and lower case letters).
- The number of tokens that start with an upper case letter.
- The number of exclamation marks in the message.
- The number of question marks.
- The sum of exclamation and question marks.
- The number of tokens containing only exclamation marks.
- The number of tokens containing only question marks.

[9] https://translate.google.gr/#el/en/

- The number of tokens containing only exclamation or question marks.
- The number of tokens containing only ellipsis (...).
- The existence of a positive emoticon at the message's end.
- The existence of a negative emoticon at the message's end.
- The existence of an ellipsis and a link (URL) at the message's end. News tweets, which are often objective, often contain links of this form.
- The existence of an exclamation mark at the message's end.
- The existence of a question mark at the message's end.
- The existence of a question or an exclamation mark at the message's end.
- The existence of slang, as detected by using the slang dictionary (Section 2.3.1).

3.3.4.2 POS based features

- The number of adjectives in the message being classified.
- The number of adverbs.
- The number of verbs.
- The number of nouns.
- The number of pronouns.
- The number of URLs.
- The number of positive emoticons.
- The number of negative emoticons.

3.3.4.3 POS bigrams features

- The average precision score of the message's POS-tag bigrams for the positive and negative classes.
- The maximum precision score of the message's POS-tag bigrams for the positive and negative classes.
- The minimum precision score of the message's POS-tag bigrams for the positive and negative classes.
- The average F_1 score of the message's POS-tag bigrams for the positive and negative classes.
- The maximum F_1 score of the message's POS-tag bigrams for the positive and negative classes.
- The minimum F_1 score of the message's POS-tag bigrams for the positive and negative classes.

For the definition of the F_1 score used see Equation 2.1 in Section 2.3.3.3.

3.3.4.4 Sentiment lexicon based features

For each polarity lexicon of Section 3.3.2.1, we use the following seven features based on the scores provided by the lexicon for each word present in the message. As we did in AUEB_{14}, we use the absolute values of the scores as both positive and negative scores are considered subjective tweets.

- Sum of the scores.
- Maximum of the scores.
- Minimum of the scores.
- Average of the scores.
- The count of the words of the message that appear in the lexicon.

– The score of the last word of the message that appears in the lexicon.

– The score of the last word of the message.

If a word does not appear in the lexicon, it is assigned a score of 0 and it is not considered in the calculation of the average, maximum, minimum and count scores.

For each lexicon of Section 2.3.1, we use seven different features based on the scores provided by the lexicon for each word present in the message.[10]

Similarly to Stage 1, we created features based on the precision and F_1 scores of the words of MPQA and the words of the lexicons generated from the training data (Malakasiotis et al., 2013). For each word w of each lexicon, we calculate the precision ($Pre(w,c)$), recall ($Rec(w,c)$) and F_1 ($F_1(w,c)$) of w with respect to class c (Equations 2.4, 2.5 and 2.6). Having assigned a precision and F_1 score to each word of each lexicon (MPQA and lexicons generated from training data), we then use the following features:

– The average precision score of the words of the message for the positive class.

– The maximum precision score of the words of the message for the positive classategory.

– The minimum precision score of the words of the message for the positive class.

– The average F_1 score of the words of the message for the positive class.

– The maximum F_1 score of the words of the message for the positive class.

– The minimum F_1 score of the words of the message for the positive class.

– The average precision score of the words of the message for the negative class.

– The maximum precision score of the words of the message for the negative class.

[10]We removed from SENTIWORDNET any instances having positive and negative scores equal to zero. Moreover, the MPQA lexicon does not provide scores, so, for each word in the lexicon we assume a score equal to 1.

- The minimum precision score of the words of the message for the negative class.
- The average F_1 score of the words of the message for the negative class.
- The maximum F_1 score of the words of the message for the negative class.
- The minimum F_1 score of the words of the message for the negative class.

3.3.4.5 Miscellaneous features

Negation. The existence of negation may change the polarity of a message. The same word list as in Stage 1 is used.

3.3.4.6 English polarity features

Similarly to Section 3.3.3.6, we append to the feature vector the 1150 features of our English polarity detection system.

3.4 Evaluation measures

To evaluate the Greek system, we used the same measures that we used in Section 2.4. See Equations 2.7 - 2.10.

3.5 Experimental results

Table 3.3 illustrates the $F_1(\pm)$ score achieved by different versions of our system on the development set. Firstly, we used only the lexicon features of each stage. This baseline achieved an $F_1(\pm)$ of 41.08%. Next, in both stages we added the morphological features and miscellaneous features. The $F_1(\pm)$ was increased to 42.4%. Subsequently, we added the POS based and POS bigrams features. This addition had a negative impact on the system's performance. Thus, we rejected these features. Lastly, we added the features from our English system, thus creating a hybrid system, we will refer to this system as $AUEB\_GR_{14}$. The new features increased remarkably the system's $F_1(\pm)$ on

the development data (49.27%). The same configuration achieved an $F_1(\pm)$ score of 49,89% on the test data. The most important groups of features of $AUEB\_GR_{14}$ are the sentiment lexicons and the features of the English system.

Features Used	**Development $F_1(\pm)$**
Lexicons	41.08
Lexicons + Morphological + Misc	42.47
Lexicons + Morphological + Misc + POS	42.11
$AUEB\_GR_{14}$	49.27

Table 3.3: $F_1(\pm)$ score of different versions of our system on the development set.

We also experimented with the creation of a sentiment lexicon using the training data and the words of the Openoffice Greek dictionary. However, this approach led to overfitting of the training data due to the small size of the training dataset; by contrast, similar approaches have employed millions of tweets (Mohammad et al., 2013) to create a lexicon.

3.5.1 Ceiling analysis and confusion matrices

Similarly to Section 2.5.3 we performed a ceiling analysis in order to decide which stage we should try to improve. We also analyzed the confusion matrices of the system.

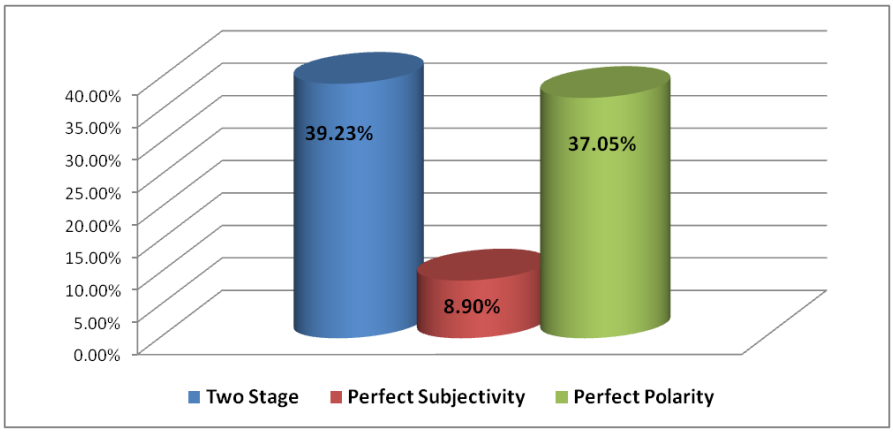

Figure 3.2: Ceiling analysis of $AUEB\_GR_{14}$ tested on the development set.

Figure 3.2 clearly indicates that we should spend all of our time to improve Stage 1, since if it were perfect then the total error of the system would have a huge decrease from 38.22% to 8.9%. By contrast, if we improve Stage 2 until it is perfect, then the total error will drop slightly to 37.05%.

Category	Predicted Subjective	Predicted Neutral
Subjective	650	492
Neutral	411	884

Table 3.4: Confusion matrix for Stage 1 (subjectivity detection).

Category	Predicted Positive	Predicted Negative
Positive	468	124
Negative	93	457

Table 3.5: Confusion matrix for Stage 2 (polarity classification).

Table 3.4 shows that most of the errors of Stage 1 occur because subjective instances are falsely classified as neutral. Table 3.5 shows that the most common error of Stage 2 is the classification of positive instances as negative.

3.6 Conclusions and future work

In this chapter, we presented the system we designed and implemented for message-level sentiment estimation of Greek social networks. We used the same two–stage pipeline approach proposed in Section 2.3, which first detects sentiment and then decides about the polarity of that sentiment, using two separate SVM classifiers. Experimental results indicate that our system is in need of improvement, while ceiling analysis clearly shows that our time should be spent on Stage 1 (subjectivity detection) as many subjective instances are falsely classified as objective.

Section 3.5 illustrated the performance of different versions of the system. As shown POS based features have a negative impact to the system's performance. The most

possible reason for this is that we have not used a POS tagger trained on Twitter data but on news articles (Koleli, 2011). To improve the performance of the system we believe that Greek natural language tools focusing on Twitter (e.g., a Twitter POS tagger) and more sentiment lexicons should be created. Moreover, other types of features such as BOW, the existence of certain types of named entities, and the use of Latent Dirichlet Allocation (Blei et al., 2003) could be experimentally tested. Finally, it might be a good choice to use feature selection via information gain and perform experiments using the most important features.

Chapter 4

Conclusions

4.1 Summary and contribution of this thesis

During this thesis a sentiment analysis system for English social media messages was developed. The system was trained on annotated tweets and participated in two international challenges where its performance was evaluated on different collection of test data such as tweets and SMS messages. The evaluation process indicated that although the system was trained only on tweets it has a good generalization ability and achieves similar results for other genres of messages. Also, using existing tools that we expanded and existing resources, we created and evaluated a Greek version of our system.

4.1.1 Message-level sentiment estimation for social networks

We decomposed the problem of sentiment analysis for social network messages into two stages and built a system based on this decomposition. During the first stage we perform subjectivity detection, (i.e., we detect whether a message expresses some sentiment or not). In the second stage, we perform polarity detection for the 'subjective' messages found in the first stage and classify them as 'positive' or 'negative'. We have used various types of features such as morphological features, twitter clusters and scores from sentiment lexicons. In addition, we propose a simple method for creating

scores for sentiment lexicons without scores and an algorithm for text normalization to improve the recall of sentiment lexicons. Using feature selection we achieved better generalization of our model. To evaluate our system we competed in two international challenges (i.e., SEMEVAL-2013 Task 2 and SEMEVAL-2014 Task 9). During the first challenge (SEMEVAL-2013) we used a simpler version of the system we used in the second challenge. In the first challenge we achieved good rankings while in the second we achieved even better rankings. We also used ceiling analysis to show that we should focus on improving Stage 1. To improve it we propose including as a feature the decision of a vagueness classifier, for which we have already done some preliminary experiments or using the existence of certain types of named entities such as products, footbal or basketball clubs etc.

4.1.2 A sentiment analysis system for Greek social network messages

We applied the methodology that we used for our English system to create a similar system for the Greek language. However, due to the lack of sufficient natural language tools for social media messages in Greek, we had to modify existing tools built for more formal text genres (i.e., news). These tools do not perform well on social messages (e.g., tweets) due to their morphology (i.e., spelling variation, slang, special tokens, Greeklish, etc.). To convert from Greeklish to Greek, we used the free Greeklish to Greek converter of Innoetics. For tokenization we expanded the Twitter specific tokenizer of Carnegie Mellon University to support both the Greek and English language. For POS tagging we utilized regular expressions to capture Twitter specific POS tags, an improved version of our text normalization algorithm, and the Greek POS tagger of AUEB's Natural Language Processing group. For stemming we used the Greek stemmer of Ntais and our text normalization algorithm. To evaluate the system we used the metrics used in the sentiment analysis challenges of SEMEVAL. Ceiling analysis indicated that there is huge potential for the system if we improve its first stage. Finally, we propose possible improvements for future work. These include the creation of Greek

natural language tools focusing on social media, the creation of more sentiment lexicons, the addition of other feature types such as BOW, using LDA, and using feature selection.

References

R. Alan, C. Sam, Mausam, and E. Oren. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of EMNLP*, pages 1524–1534, Edinburgh, U.K.

P. Alexopoulos and J. Pavlopoulos. 2014. A vague sense classifier for detecting vague definitions in ontologies. In *Proceedings of EACL*, pages 33–37, Gothenburg, Sweden.

C. O. Alm. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *In Proceedings of HLT/EMNLP*, pages 347–354.

S. Baccianella, A. Esuli, and F Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*, Valletta, Malta.

L. Barbosa and J. Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of COLING*, pages 36–44, Beijing, China.

J. Bellegarda. 2010. Emotion analysis using latent affective folding and embedding. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 1–9, Los Angeles, CA, June.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

A. Chalamandaris, A. Protopapas, P. Tsiakoulis, and S. Raptis. 2006. All Greek to me! An automatic Greeklish to Greek transliteration system.

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March.

R. De La Briandais. 1959. File searching using variable length keys. In *Proceedings of the Western Joint Computer Conference*, pages 295–298, San Francisco, California.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

A. Go, R. Bhayani, and L. Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177, Seattle, WA, USA.

R. M. Karampatsis, J. Pavlopoulos, and P. Malakasiotis. 2014. AUEB: Two stage sentiment analysis of social network messages. In *Proceedings of SemEval*, SemEval '14, Dublin, Ireland.

R. M. Karampatsis. 2012. Named entity recognition in Greek texts of social media. BSc thesis, Athens University of Economics and Business, Greece.

E. Koleli. 2011. A new Greek part-of-speech tagger, based on a maximum entropy classifier. BSc thesis, Athens University of Economics and Business, Greece.

H. Liu and R. Setiono. 1995. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the 7th International Conference on Tools with Artificial Intelligence*, pages 388–391.

B. Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

P. Malakasiotis, R. M. Karampatsis, K. Makrynioti, and J. Pavlopoulos. 2013. nlp.cs.aueb.gr: Two stage sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 562–567, Atlanta, Georgia, June.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

S. M. Mohammad, S. Kiritchenko, and X. Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of SemEval*, Atlanta, Georgia, USA.

F. A. Nielsen. 2011. A new anew: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of ESWC*, pages 93–98, Heraclion, Greece.

G. Ntais. 2006. Development of a stemmer for the Greek language. Master's thesis, Stockholm University, Sweden.

O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*, Atlanta, Georgia.

B. Pang and L. Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, Barcelona, Spain.

B. Pang and L. Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124, Ann Arbor, MI, USA.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation*, SemEval '14, Dublin, Ireland.

M. Tsytsarau and T. Palpanas. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514.

V. Vapnik. 1998. *Statistical Learning Theory*. Wiley.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354, Vancouver, BC, Canada.

T. Wilson, Z. Kozareva, P. Nakov, S. Rosenthal, V. Stoyanov, and A. Ritter. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, Atlanta, Georgia.