Determining Negation Scope and Strength in Sentiment Analysis

Alexander Hogenboom, Paul van Iterson, Bas Heerschop, Flavius Frasincar, and Uzay Kaymak

Erasmus University Rotterdam
Rotterdam, The Netherlands
hogenboom@ese.eur.nl, {paulvaniterson, basheerschop}@gmail.com, {frasincar, kaymak}@ese.eur.nl

***Abstract*—A key element for decision makers to track is their stakeholders' sentiment. Recent developments show a tendency of including various aspects other than word frequencies in automated sentiment analysis approaches. One of these aspects is negation, which can be accounted for in various ways. We compare several approaches to accounting for negation in sentiment analysis, differing in their methods of determining the scope of influence of a negation keyword. On a set of English movie review sentences, the best approach is to consider two words, following a negation keyword, to be negated by that keyword. This method yields a significant increase in overall sentiment classification accuracy and macro-level F_1 of 5.5% and 6.2%, respectively, compared to not accounting for negation. Additionally optimizing sentiment modification of negated words to a value of –1.27 rather than –1 yields a significant 7.1% increase in accuracy and a significant 8.0% increase in macro-level F_1.**

***Index Terms*—Sentiment analysis, negation scope, negation strength.**

I. Introduction

Information monitoring tools are of paramount importance to decision makers in today's complex systems. Such tools may support decision makers by identifying issues and patterns that matter as well as by tracking and predicting emerging events. A key element for decision makers to track is their stakeholders' sentiment, as has been demonstrated for, e.g., economic systems [1], financial markets [2], politics [3], organizations [4], and reputation or brand management [5].

The Web offers an overwhelming amount of textual data, containing traces of sentiment. Such data may be published through, e.g., blogs, reviews, or Twitter. Analyzing these textual data can enable extraction of the information tailored to the needs of decision makers. Yet, the analysis of sentiment in this overwhelming amount of data is challenging at best.

Sentiment analysis refers to a broad area of natural language processing, computational linguistics, and text mining. Typically, the goal is to determine the polarity of natural language texts. Existing, straightforward approaches are typically statistical ones, based on frequencies of positive and negative words. More recently, researchers have been exploring ways of accounting for various other aspects of content, such as structural or semantic aspects.

One of the areas of focus is accounting for negation, which is typically done by inverting the polarity of negated words. For instance, the negative sentiment associated with the word "bad" would typically be inverted into a positive sentiment for the phrase "not bad". In previous work [6], we assessed the impact of a simple way of accounting for negation, i.e., inverting the polarity of sentiment carrying words directly following negation keywords. Performance improvements compared to not taking into account negation turned out to be somewhat marginal, which inspired us to explore more complex approaches to handling negation.

A major challenge in this respect is determining which words are negated by a negation keyword. Several approaches to optimizing the scope of influence of a negation keyword have already been proposed, usually within a larger framework for sentiment analysis. Yet so far, the impact of these approaches as such has niether been assessed nor compared. Therefore, the aim of this paper is to provide measurements and comparisons of these methods. Additionally, we investigate the impact of introducing a notion of the extent to which a word's sentiment is negated, as the sentiment associated with, e.g., "not bad" may not necessarily be as positive as the sentiment associated with "good". Alternatively, negation may be used in order to stress the disparity between the negation phrase and its non-negated counterpart, thus resulting in amplification of the sentiment of the negation phrase.

The remainder of the paper is organized as follows. In Section II, we briefly discuss related work on accounting for negation in sentiment analysis. We then elaborate on our framework for assessing different ways of accounting for negation in sentiment analysis in Section III. Our findings are discussed in Section IV and we conclude in Section V.

II. Negation in Sentiment Analysis

As we have shown in [6], most approaches to sentiment analysis essentially adhere to more or less similar frameworks, which typically consist of a sentiment lexicon creation phase and a subsequent lexicon-based sentiment scoring phase. Despite adhering to similar frameworks, document sentiment analysis approaches have several characteristic features distinguishing them from one another. For instance, sentiment may be scored on document level, sentence level, or window level. In matching words in a text with words in a sentiment lexicon, some approaches as lemmatization are designed to cope with syntactical variations. Additionally, some algorithms attempt to identify subjective phrases or phrases relevant to the topic considered in order to boost sentiment analysis performance.

978-1-4577-0653-0/11/$26.00 ©2011 IEEE

Other helpful techniques include taking into account amplification or negation of sentiment carrying words [7].

Negation in linguistics is the process of turning an affirmation into its opposite, or vice versa. There are many forms of negation, which can be divided into two main groups: local and long distance negation [8]. In case of local negation, negation keywords syntactically apply to sentiment carrying words (e.g., “I do not like something”). Conversely, long distance negation implies that negation does not directly apply to sentiment carrying words, but to, e.g., a contradictory clause (e.g., “I really love this, but I do not feel the same about that”). In our current endeavors, we focus on local negation, as is the case in most existing sentiment analysis approaches.

A major challenge when accounting for negation is determining the words that are affected by a negation keyword – the negation scope [9]. Sophisticated approaches to optimizing this scope involve complex rules [9], machine learning methods [10], or identifying negated words through processing compositional semantics of phrases [11]. However, many existing sentiment analysis approaches have relatively straightforward conceptualizations of the scope of negation keywords.

For instance, in early work [12], words in between a negation keyword and the first punctuation mark following it are considered to be in that negation keyword’s scope. Other work elaborates on experiments with considering the remainder of a sentence as well as with considering the first sentiment carrying word following a negation keyword [13]. Yet, most recent work exhibits a tendency of considering words near negation keywords to be in the scope of those keywords. These approaches typically differ in the considered number of words following or around negation keywords, as well as in the types of considered words. Some approaches only consider specific types of words such as adjectives [14], [15] or adjectives and adverbs [16], whereas other approaches consider any sentiment carrying word [17], [18], [19], [20], [21]. The impact of such recent, relatively simple approaches is yet to be assessed.

III. Framework

In order to assess the impact of several approaches to sentiment negation, we propose a basic sentence-level sentiment analysis framework. This framework uses word-level sentiment scores in the range $[-1, 1]$ (anywhere in between negative and positive, respectively) derived from a sentiment lexicon in order to classify sentences as either positive (1) or negative (-1).

The sentiment lexicon utilized in our approach is based on a large, widely used, readily available (semantic) lexical resource: WordNet [22], the design of which is inspired by psycholinguistic theories of human lexical memory. WordNet is designed to be used under program control and enables the distinction between different word forms and word meanings. WordNet is organized into sets of synonyms – synsets – which can be differentiated based on their Part-of-Speech (POS) type. Each synset expresses a distinct concept, which has a natural language description (gloss) and is linked to other synsets through different kinds of relations (e.g., synonymy, antonymy, hyponymy, or meronymy). In SentiWordNet [23], each WordNet synset σ has been assigned scores in the range $[0, 1]$ on objectivity $Obj(\sigma)$, positivity $Pos(\sigma)$, and negativity $Neg(\sigma)$, the sum of which always equals 1. $Obj(\sigma) > 0$ means a less subjective word and thus weaker sentiment scores in $Pos(\sigma)$ and $Neg(\sigma)$. We ignore the objectivity score, as it implicitly influences the positive and negative scores, and define our own word sentiment score as a single real number computed by subtracting $Neg(\sigma)$ from $Pos(\sigma)$, which results in a real number in the interval $[-1, 1]$, representing sentiment scores in the range from negative to positive, respectively.

A. Sentiment Classification

In order to retrieve sentiment scores of words (simple or compound) from SentiWordNet, POS types, lemmas, and word senses need to be determined first for each word in a sentence. For the word sense disambiguation process, we propose to use a freely available Lesk algorithm [24] implementation for WordNet [25]. The algorithm, described in Algorithm 1, selects the word sense that is semantically most similar to the words in the context (i.e., the other words in the sentence). Having disambiguated the word senses, sentiment scores can be retrieved from SentiWordNet. However, in our approach, these SentiWordNet sentiment scores may not be sufficient, as we aim to process negation as well. Our negation processing approach relies on occurrences of (English) negation keywords defined in [9]: “no”, “not”, “-n’t”, “never”, “less”, “without”, “barely”, “hardly”, “rarely”, “no longer”, “no more”, “no way”, “no where”, “by no means”, “at no time”, and “not (...) anymore”. Each negation keyword is assumed to have a scope of influence of the negation. This scope can be determined in many ways, as further detailed in Section III-B.

Having determined the scope of negation keywords using any of the considered methods, the sentiment scores associated with the words in the negation keywords’ scope can be inverted. To this end, we introduce per-word sentiment modifiers, which are initialized at a value of 1, indicating that the sentiment score retrieved from the sentiment lexicon is considered to be the true sentiment score associated with that word in the considered context. In case a word is negated, the sentiment modifier may be multiplied with an inversion factor i. Initially, we assume this factor to be equal to -1. However, as we hypothesize that negated sentiment may not necessarily be as strong as its non-negated counterpart (compare, e.g., “not bad” and “good”), our framework also supports Modified Inversion Strength (MIS), where the inversion factor may be anything in the range $[-2, 0]$.

Finally, when all word scores have been determined while accounting for negation, sentences can be classified as either positive or negative. To this end, we use a sentence scoring function. If the sum of word-level sentiment scores in a sentence produces a number smaller than 0, the sentence is classified as negative, else, the sentence is classified as a positive sentence.

Algorithm 1: Word Sense Disambiguation.

```
input : The to be disambiguated word w and the sentence s that
        contains the word
output: The sense sense of w with the highest semantic similarity to
        the words in the context
tSenses = ∅; // Senses of the target word w
tGlosses = ∅; // Glosses of the senses for w
senseScores = ∅; // Scores of the senses for w
bestSense = ∅; // Best sense for w
bestScore = −1; // Score for best sense for w
k = 8; // Considered context around w
// Retrieve the sequence of words starting k/2
// words to the left of w and ending k/2 words
// to the right of w, excluding w
context = getContext(s, w, k);
// Look up and add all senses of POS noun and
// verb for w
tSenses = getSenses(w);
foreach sense in tSenses do
    // Retrieve the gloss of the sense and the
    // glosses connected to it through hypernym,
    // hyponym, meronym, and troponym relations
    tGlosses[sense] = {tGlosses, getGlosses(sense)};
end
foreach word in context do
    // Look up and add all senses of POS noun and
    // verb for word
    senses = getSenses(word);
    foreach sense in senses do
        // Retrieve the gloss of the sense and the
        // glosses connected to it through
        // hypernymy, hyponymy, meronymy, and
        // troponymy
        glosses = getGlosses(sense);
        foreach gloss in glosses do
            foreach tGloss in tGlosses[sense] do
                // Each overlap with N consecutive
                // words contributes N^2 to the gloss
                // sense combination score
                overlapScore = overlap(gloss, tGloss);
                senseScores[tGloss] += overlapScore;
            end
        end
    end
end
foreach sense in tSenses do
    if senseScores[getGloss(sense)] > bestScore then
        bestScore = senseScores[getGloss(sense)];
        bestSense = sense;
    end
end
return bestSense;
```

Our framework is formalized in Algorithm 2. The input to this algorithm is a sentence and the output is its sentiment classification. Additional inputs include – if applicable – the desired negation scope determination method, sentiment inversion factor, scope direction, and window size. For each identified negation keyword, the sentiment modifier of the words within the scope of this keyword is multiplied with the sentiment inversion factor. This sentiment modifier initially equals 1, indicating that no inversion is applied. When all negation keywords have been processed, the sentence is scored by summing the (modified) sentiment scores of all words in the sentence. The resulting sentiment score is then used to classify the sentence.

Algorithm 2: Sentiment Analysis.

```
input : A sentence s and – if applicable – a negation scope
        determination method m, an inversion factor i, a scope
        direction d, and FWL window size k
output: The sentiment classification of sentence s
// Retrieve and process negation keywords
keys = getNegKeyWords(s);
foreach key in keys do
    // Update sentiment modifiers of words in the
    // negation scope of this keyword
    scope = getScope(s, key, m, d, k);
    foreach word in scope do
        // Initial sentiment modifier equals 1,
        // i.e., by default, sentiment of words is
        // not negated to any extent
        word.mod = word.mod × i;
    end
end
// Compute sentence sentiment score as sum of
// individual words' sentiment scores
sentenceScore = 0;
foreach word in sentence do
    pos = getPOS(word, sentence);
    lemma = getLemma(word, pos);
    sense = getWordSense(word, sentence, pos);
    score = getWordScore(lemma, sense, pos) × word.mod;
    sentenceScore = sentenceScore + score;
end
// Determine sentiment classification
class = 1;
if sentenceScore < 0 then
    class = −1;
end
return class;
```

B. Negation Scope Determination

Inspired by the existing, relatively straightforward approaches mentioned in Section II, we consider four negation scope determination approaches (see Table I). First, we consider the Rest of the Sentence (RoS), following or around a negation keyword, to be negated. Second, we negate the sentiment of the First Sentiment-carrying Word (FSW) following or around a negation keyword. The third method we consider involves negating only the sentiment of the first word following a negation keyword. However, when the first word is an adverb, we negate the sentiment of the word following that adverb, as we assume adverbs to typically modify other (sentiment carrying) words, thus making the Next Non-Adverb (NNA) following a negation keyword a more viable candidate for negation. In our fourth method, we consider a Fixed Window Length (FWL) of k words following or $2k$ words around a negation keyword to be in the scope of that keyword.

TABLE I
CONSIDERED APPROACHES TO NEGATION SCOPE DETERMINATION.

Acronym	Negation scope
RoS	Rest of the sentence, following or around a negation keyword
FSW	First sentiment-carrying word following or around a negation keyword
NNA	Next non-adverb following a negation keyword
FWL	A fixed window length of words following or preceding and following a negation keyword

IV. Evaluation

In order to be able to assess the effects of distinct methods of determining negation scope as well as of our proposed method for accounting for negation strength, we have implemented the framework presented in Section III. The implementation was done in C#.Net in combination with a Microsoft SQL Server database. For lemmatization and word sense disambiguation, we use functionalities provided by the open-source C# WordNet.Net WordNet API[1]. The word sense disambiguation approach utilized by this API is the Lesk-based approach discussed in Algorithm 1. Our POS tagger – with an accuracy of 98.7% [26] – is based on SharpNLP[2] and has been provided to us by Teezir[3]. The sentiment lexicon used in our framework is SentiWordNet 3.0 [23].

The performance of our considered sentiment lexicon approaches was evaluated on a collection of $1{,}000$ positive and $1{,}000$ negative English movie reviews[4], which have been extracted from movie review web sites by Pang and Lee [27]. The review classifications have been derived from the accompanying numerical review scores. Pang and Lee supply an extract of this data set – a collection of $10{,}662$ sentences from the original texts, which have been rated for sentiment as well. Of these $10{,}662$ sentences, we marked $2{,}285$ sentences that contain one or more of our considered negation keywords [9]. Of these $2{,}285$ sentences (with an average length of 25 words), 930 have been classified as positive (1), whereas $1{,}355$ have been classified as negative (-1).

On this corpus, we have evaluated the performance of our sentiment analysis framework when using different methods for determining the scope of negation keywords: ROS, FSW, NNA, and FWL. Each approach, where applicable, has been assessed with the direction of the scope set to (a subset of) the words following, as well as around identified negation keywords. For the FWL method, we also experimented with window sizes $k \in \{1, 2, 3, 4\}$. In our experiments, we initially assigned the sentiment inversion factor a value of -1. For the best performing approach, we subsequently optimized the sentiment inversion factor by means of a hill-climbing procedure, starting from four random starting points in the interval $[-2, 0]$ and iteratively increasing or decreasing the inversion factor with 0.01. The objective of this procedure was to maximize the resulting model's overall performance. The performance of all considered approaches has been compared to the performance of our baseline approach, which does not have any support for negation.

In our evaluation, several performance measures have been taken into account. For both the positive documents and the negative documents, we report precision, recall, and the F_1 measure. Precision is the percentage of the positively (negatively) classified documents which have an actual classification of positive (negative). Recall is the percentage of the actual positive (negative) documents which is also classified as such. The F_1 measure is the harmonic mean of precision and recall. We also report some statistics on our full corpus. We report the macro-level F_1 measure, which is the average of the F_1 scores of the two classifications, and the accuracy, which is the total percentage of correctly classified documents. In our comparisons of our considered approaches, we assess the statistical relevance of observed differences in performance using a paired, two-sided Wilcoxon signed-rank test, which evaluates the hypothesis that the differences between paired observations are symmetrically distributed around a median equal to 0. If this null hypothesis is rejected, the compared samples are significantly different. Our results are reported in Tables II and III.

Table II shows that accounting for negation with a sentiment inversion factor equal to -1 appears to predominantly have a significant, positive effect on the classification of negative sentences in our corpus, mainly in terms of recall. Conversely, the recall of positive sentence classifications is negatively affected by all considered approaches, thus nullifying the performance improved classification of negative sentences for most considered methods for negation handling. Additionally, accounting for negation does not appear to have a significant impact on the precision of classification of both positive and negative sentences.

Overall, the worst performing approach is ROS, which even performs worse than the baseline when the direction of the scope is set to the words around negation keywords. NNA and FWL considering words around a keyword to be in the scope do not appear to perform much better than the baseline either in terms of overall accuracy and macro-level F_1. FSW on the other hand exhibits a significantly increased accuracy and macro-level F_1 compared to the baseline when considering the first sentiment-carrying word following a negation keyword to be in the keyword's scope. However, the best performing approach turns out to be FWL, considering two words following a negation keyword. With respect to the baseline, accuracy significantly increases with 5.5% from 49.9% to 52.7% and macro-level F_1 significantly increases with 6.2% from 49.4% to 52.4%.

However, the performance of FWL considering two words following a negation keyword can be further improved by optimizing the sentiment inversion factor. Table III shows the performance of the sentiment analysis framework, given a sample of values for the sentiment inversion factor i. Sentiment inversion factor values closer to 0 (i.e., those indicating weakened inversion) tend to improve the performance on positive sentences at the cost of the performance on negative sentences. Conversely, stronger inversion typically yields the opposite effect. The sentiment inversion factor exhibiting the optimal overall performance on our corpus equals -1.27, which yields a 7.1% increase in accuracy from 49.9% to 53.5% and an 8.0% increase in macro-level F_1 from 49.4% to 53.3% with respect to the baseline. These increases constitute statistically significant improvements with respect to not optimizing the negation strength.

[1]http://opensource.ebswift.com/WordNet.Net/
[2]http://sharpnlp.codeplex.com/
[3]http://www.teezir.com/
[4]http://www.cs.cornell.edu/People/pabo/movie-review-data/

TABLE II

EXPERIMENTAL RESULTS FOR ALL APPROACHES WITH SENTIMENT INVERSION FACTOR -1. BOLDFACE INDICATES MAXIMUM VALUES FOR PERFORMANCE MEASURES. PERFORMANCE MEASURES MARKED WITH * SIGNIFICANTLY DIFFER FROM THE BASELINE AT $p < 0.05$, WHEREAS PERFORMANCE MEASURES MARKED WITH ** SIGNIFICANTLY DIFFER FROM THE BASELINE AT $p < 0.01$.

			Positive			Negative			Overall	
Method	Direction	k	Precision	Recall	F_1	Precision	Recall	F_1	Accuracy	Macro F_1
Baseline	-	-	43.1%	**72.3%**	53.9%	64.2%	34.6%	44.9%	49.9%	49.4%
RoS	Following	-	42.3%	59.9%**	49.4%**	61.6%	44.1%**	51.3%**	50.5%	50.3%
RoS	Around	-	39.3%**	50.7%**	44.1%**	57.8%**	**46.4%****	**51.4%****	48.1%	47.7%
FSW	Following	-	44.3%*	69.4%*	54.0%	65.9%	40.5%**	50.1%**	52.3%**	52.0%**
FSW	Around	-	43.2%	64.5%**	51.7%	63.2%	42.0%**	50.4%**	51.2%	51.0%
NNA	Following	-	43.6%	71.4%	54.0%	65.2%	36.8%**	46.9%**	50.9%	50.4%
FWL	Following	1	43.9%*	71.6%	54.3%	65.7%	37.5%**	47.6%**	51.3%*	51.0%*
FWL	Following	2	**44.8%****	70.8%	**54.7%**	**66.7%**	40.3%**	50.1%**	**52.7%****	**52.4%****
FWL	Following	3	44.1%	69.2%*	53.8%	65.5%	40.2%**	49.7%**	52.0%*	51.7%**
FWL	Following	4	44.2%	68.3%*	53.6%	65.5%	41.2%**	50.5%**	52.2%*	52.0%**
FWL	Around	1	43.6%	71.0%	53.9%	65.0%	37.2%*	47.2%*	50.9%	50.5%
FWL	Around	2	43.2%	67.3%**	52.5%	63.7%	39.5%**	48.7%*	50.8%	50.6%
FWL	Around	3	43.0%	64.9%**	51.7%	63.2%	41.3%**	49.9%**	50.9%	50.8%
FWL	Around	4	42.9%	64.1%**	51.3%*	62.8%	41.7%**	50.0%*	50.8%	50.6%

TABLE III

EXPERIMENTAL RESULTS FOR MIS WITH FWL CONSIDERING TWO WORDS FOLLOWING A NEGATION KEYWORD. MAXIMUM VALUES FOR EACH PERFORMANCE MEASURE ARE PRINTED IN BOLD. PERFORMANCE MEASURES MARKED WITH * SIGNIFICANTLY DIFFER FROM THE BEST PERFORMING FWL METHOD WITH SENTIMENT INVERSION FACTOR -1 AT $p < 0.05$, WHEREAS ** INDICATES SIGNIFICANT DIFFERENCES AT $p < 0.01$.

	Positive			Negative			Overall	
i	Precision	Recall	F_1	Precision	Recall	F_1	Accuracy	Macro F_1
0.00	43.7%*	**74.6%****	**55.0%**	65.8%	34.2%**	44.9%**	50.6%**	49.9%**
-0.25	44.5%	72.4%*	55.0%	66.5%	38.1%**	48.4%**	52.1%	51.7%
-0.50	44.8%	71.8%	55.0%	66.9%	39.4%	49.5%	52.6%	52.3%
-0.75	44.5%	70.4%	54.5%	66.2%	40.1%	49.9%	52.4%	52.2%
-1.00	44.8%	70.8%	54.7%	66.7%	40.3%	50.1%	52.7%	52.4%
-1.25	45.2%*	69.7%*	54.7%	**67.0%**	42.2%**	51.7%**	53.4%**	53.2%**
-1.27	**45.2%***	69.5%*	54.7%	67.0%	42.5%**	**51.9%****	**53.5%****	**53.3%****
-1.50	44.9%	69.0%**	54.3%	66.4%	42.1%**	51.4%*	53.0%	52.9%
-1.75	44.8%	68.0%**	53.9%	66.0%	**42.8%****	51.9%**	53.0%	52.9%
-2.00	44.7%	67.7%**	53.7%	65.7%	42.6%**	51.7%*	52.8%	52.7%

Yet, even after optimizing the sentiment inversion factor, the assessed approaches to processing negation when analyzing sentiment appear to have a rather limited effect on overall performance in terms of changes in precision, recall, accuracy, and F_1. A closer look at the data reveals that when accounting for negation by means of any of our considered methods, sentiment scores of approximately 20–80% and classifications of approximately 10–30% of the sentences change with respect to not taking into account negation. For our best performing method, 38.3% of the sentiment scores are influenced, resulting in 16.4% of the sentences to be classified differently. Juxtaposing these numbers with the relatively low increase in performance, we conclude that our considered methods for handling negation have a high impact, yet are far from perfect. Apparently, negation handling not only improves sentiment classification of sentences, but introduces new classification errors as well – albeit to a lesser extent.

This observation suggests that the considered relatively simple methods for handling negation are not subtle enough to deal with complex natural language. Common phrases may be hard to process for straightforward negation handling methods. For instance, the sentence "I like that Smith; he's not making fun of these people, he's not laughing at them." contains the common phrases "making fun of people" and "laughing at people", which typically have a negative connotation in spite of their positive sentiment carrying words. Accounting for negation in this sentence does not help at all when the semantics of the common phrases are ignored. Furthermore, in the sentence "You could nap for an hour and not miss a thing.", the phrase "miss" carries a negative sentiment, but accounting for its negation would yield a positive sentiment. However, the sentiment here comes from the meaning of napping – and inherently missing a part of the movie – which, combined with the common phrase "not missing a thing", should yield a negative sentiment. Therefore, a deeper understanding of the semantics may be required to correctly classify the sentiment of such sentences.

V. Conclusions and Future Work

Our analysis shows that properly accounting for negation when analyzing sentiment in natural language texts may help improve the performance of classifying unseen natural language text as carrying either positive or negative sentiment. However, some approaches to accounting for negation in sentiment analysis have proven to be more effective than others. The relatively simple approaches considered in our current study typically differ in their methods of determining the scope of influence of a negation keyword. On our data set consisting of English review sentences, the best performing method is considering two words following a negation keyword to be negated by that keyword. When using this method, overall accuracy significantly increases with 5.5% and macro-level F_1 significantly increases with 6.2%, compared to not accounting for negation. Optimizing the sentiment modification in case of negation to a value of -1.27 rather than -1 yields a significant 7.1% increase in accuracy and a significant 8.0% increase in macro-level F_1.

As distinct values for the extent to which sentiment is negated clearly have different effects on the performance of classification of positive and negative sentences, an interesting direction for future research would be to explore the applicability of distinct sentiment inversion factors for positive and negative words. Additionally, one could explore ways of incorporating the position of a negation keyword into the sentiment analysis process, as for instance a negation keyword at the end of a sentence is likely to affect preceding rather than following words. Another interesting direction for future research would be to explore ways of incorporating a deeper understanding of the semantics in the negation handling process in order to be able to cope with common phrases or context-dependent interpretations.

Acknowledgments

We would like to thank Teezir for their technical support and fruitful discussions.

References

[1] S. Ludvigson, "Consumer Confidence and Consumer Spending," *The Journal of Economic Perspectives*, vol. 18, no. 2, pp. 29–50, 2004.

[2] I. Arnold and E. Vrugt, "Fundamental Uncertainty and Stock Market Volatility," *Applied Financial Economics*, vol. 18, no. 17, pp. 1425–1440, 2008.

[3] D. Baron, "Competing for the Public through the News Media," *Journal of Economics and Management Strategy*, vol. 14, no. 2, pp. 339–376, 2005.

[4] C. Holton, "Identifying Disgruntled Employee Systems Fraud Risk Through Text Mining: A Simple Solution for a Multi-Billion Dollar Problem," *Decision Support Systems*, vol. 46, no. 4, pp. 853–846, 2009.

[5] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter Power: Tweets as Electronic Word of Mouth," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[6] B. Heerschop, P. van Iterson, A. Hogenboom, F. Frasincar, and U. Kaymak, "Analyzing Sentiment in a Large Set of Web Data while Accounting for Negation," in *7th Atlantic Web Intelligence Conference (AWIC 2011)*. Springer, 2011, pp. 195–205.

[7] M. Taboada and K. Voll, "Extracting Sentiment as a Function of Discourse Structure and Topicality," Simon Fraser University, Tech. Rep. 20, 2008, available online, ftp://ftp.fas.sfu.ca/pub/cs/TR/2008/CMPT2008-20.pdf.

[8] T. Wilson, J. Wiebe, and P. Hoffman, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in *Human Language Technology Conference and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. ACL, 2005, pp. 347–354.

[9] L. Jia, C. Yu, and W. Meng, "The Effect of Negation on Sentiment Analysis and Retrieval Effectiveness," in *18th ACM Conference on Information and Knowledge Management (CIKM 2009)*. ACM, 2009, pp. 1827–1830.

[10] R. Morante, A. Liekens, and W. Daelemans, "Learning the Scope of Negation in Biomedical Texts," in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. ACL, 2008, pp. 715–724.

[11] Y. Choi and C. Cardie, "Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis," in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. ACL, 2008, pp. 793–801.

[12] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. ACL, 2002, pp. 79–86.

[13] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie and Product Reviews using Contextual Valence Shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[14] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," in *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. ACM, 2004, pp. 168–177.

[15] D. Milea, N. Sharef, R. Almeida, U. Kaymak, and F. Frasincar, "Prediction of the MSCI EURO Index Based on Fuzzy Grammar Fragments Extracted from European Central Bank Statements," in *2010 International Conference on Soft Computing and Pattern Recognition (SoCPaR 2010)*. IEEE, 2010, pp. 231–236.

[16] F. Benamara, C. Cesarano, and D. Reforgatio, "Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone," in *1st International Conference on Weblogs and Social Media (ICWSM 2007)*. AAAI Press, 2007, pp. 203–206.

[17] A. Agarwal, F. Biadsy, and K. McKeown, "Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-Grams," in *Conference of the European Chapter of the ACL (EACL 2009)*. ACL, 2009, pp. 24–32.

[18] M. Dadvar, C. Hauff, and F. de Jong, "Scope of Negation Detection in Sentiment Analysis," in *11th Dutch-Belgian Information Retrieval Workshop (DIR 2011)*, 2011, pp. 16–19.

[19] X. Ding, B. Liu, and P. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining," in *1st ACM International Conference on Web Search and Data Mining (WSDM 2008)*. ACM, 2008, pp. 231–240.

[20] G. Grefenstette, Y. Qu, J. Shanahan, and D. Evans, "Coupling Niche Browsers and Affect Analysis for an Opinion Mining Application," in *7th International Conference on Computer-Assisted Information Retrieval (RIAO 2004)*, 2004, pp. 186–194.

[21] S. Kim and E. Hovy, "Determining the Sentiment of Opinions," in *20th International Conference on Computational Linguistics (COLING 2004)*. ACL, 2004, pp. 1367–1371.

[22] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[23] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in *7th Conference on International Language Resources and Evaluation (LREC 2010)*. ELRA, 2010, pp. 2200–2204.

[24] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," in *5th Annual International Conference on Systems Documentation (SIGDOC 1986)*. ACM, 1986, pp. 24–26.

[25] T. Dao and T. Simpson, "Measuring Similarity between Sentences," WordNet.Net, Tech. Rep., 2005, available online, http://wordnetdotnet.googlecode.com/svn/trunk/Projects/Thanh/Paper/WordNetDotNet_Semantic_Similarity.pdf.

[26] E. Buyko, J. Wermter, M. Poprat, and U. Hahn, "Automatically adapting an NLP Core Engine to the Biology Domain," in *9th Bio-Ontologies Meeting and the Joint Linking Literature Information and Knowledge for Biology (ISMB 2006)*. Oxford University Press, 2006, pp. 65–68.

[27] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on Minimum Cuts," in *42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*. ACL, 2004, pp. 271–280.