Analyzing Sentiment in a Large Set of Web Data while Accounting for Negation

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

Abstract As virtual utterances of opinions or sentiment are becoming increasingly abundant on the Web, automated ways of analyzing sentiment in such data are becoming more and more urgent. In this paper, we provide a classification scheme for existing approaches to document sentiment analysis. As the role of negations in sentiment analysis has been explored only to a limited extent, we additionally investigate the impact of taking into account negation when analyzing sentiment. To this end, we utilize a basic sentiment analysis framework – consisting of a wordbank creation part and a document scoring part – taking into account negation. Our experimental results show that by accounting for negation, precision on human ratings increases with 1.17%. On a subset of selected documents containing negated words, precision increases with 2.23%.

Key words: Sentiment analysis, negation, wordbank creation, document scoring

1 Introduction

With the advent of the Web, traces of human activity and communication have become ubiquitous, partly in the form of written text. In recent years, virtual utterances of opinions or sentiment have become increasingly abundant through messages on Twitter, on-line customer reviews, etcetera. The information contained in this ever-growing data source of the Web is invaluable to key decision makers, e.g., those making decisions related to reputation management or marketing. An understanding of what is going on in their particular markets is crucial for decision makers, yet the analysis of sentiment in an overwhelming amount of data is far from trivial.

Bas Heerschop, Paul van Iterson, Alexander Hogenboom, Flavius Frasincar, and Uzay Kaymak
Erasmus University Rotterdam, PO Box 1738, NL-3000 DR, Rotterdam, The Netherlands, e-mail: {basheerschop,paulvaniterson}@gmail.com, {hogenboom,frasincar,kaymak}@ese.eur.nl

Sentiment analysis refers to a broad area of natural language processing, computational linguistics and text mining. In general it aims to determine the attitude, evaluation, or emotions of the author with respect to the subject of the text. The basis of sentiment analysis is determining the positive-negative polarity of a text. The research area of sentiment analysis is relatively new, with many aspects being currently explored. The most promising areas of focus are word sentiment scoring (i.e., learning the sentiment scores of single words), subject/aspect relevance filtering (i.e., determining the subject and/or aspect a sentiment carrying word is relevant to), subjectivity analysis (i.e., determining whether a sentence is subjective or objective), and sentiment amplification and negation (i.e., modifying sentiment strength on amplifying words and reversing sentiment scores on negated words).

Some researchers have already suggested to account for negation when analyzing sentiment in texts. Yet so far, the impact of taking into account negation when analyzing sentiment has not been demonstrated. Therefore, we present our first steps towards insight in the impact of negation on sentiment analysis. The remainder of this paper is organized as follows. First, we classify existing sentiment analysis approaches and assess the extent to which they account for negation in Sect. 2. Then, we describe and utilize our framework for assessing the impact of negation in sentiment analysis in Sect. 3. We conclude in Sect. 4.

2 Sentiment Analysis

In recent years, several approaches to sentiment analysis (i.e., classification) of documents have been proposed. Most approaches essentially adhere to more or less similar frameworks. One of such frameworks is the basic framework proposed by Liu [11, 12], consisting of an algorithm for creating a wordbank (i.e., a list of words and their associated sentiment) from a training corpus, along with a document-level scoring function. Ceserano et al. [3] propose a similar framework, which has been used by other researchers as well [1, 2]: OASYS. OASYS provides two word scoring algorithms based on supervised learning and three sentence-level document scoring algorithms with topic relevance filtering. Despite adhering to similar frameworks, document sentiment analysis approaches have several characteristic features distinguishing them from one another. We consider the following features.

Wordbank (WB) Most approaches rely on a wordbank, typically containing per-word sentiment scores. Creation methods include supervised learning on a set of manually rated documents, learning through related word expansion (expanding a small, manually created set of words by exploiting word relationships such as synonyms, antonyms, and hypernyms), completely manual creation, or a combination of these methods. The target of the wordbank (e.g., general or domain-specific) may also differ amongst approaches, as well as the differentiation between part-of-speech variations of a word.

Sentiment scoring level (SSL) One could consider sentiment analysis to be performed at document level, sentence level, or window level.

Topic relevance filtering (TRF) Taking into account the way in which sentiment carrying words are tied to their subject results in allegedly irrelevant phrases being filtered out of further processing. More advanced methods look at the surrounding words of a sentiment carrying word, the subject of sentences that contain sentiment, or specific features of a subject.

Subjectivity filtering (SF) Subjective sentences carry sentiment, whereas objective sentences only carry factual information. Ignoring objective sentences is crucial for some sentence-level and window-level algorithms with aggregation functions averaging sentiment expressed in the subparts. Including objective sentences here would decrease the impact of subjective sentences.

Part-of-speech tagging (POS) Annotating words with their corresponding parts-of-speech (POS) – e.g., noun, verb, adjective, subject, or object – can help algorithms making better decisions. For example, “I like A”, where “like” is a verb carrying high positive sentiment, is very different from “A is like B” where “like” is an adverb carrying no sentiment. POS tagging can also be used to identify the subject of a sentence to which a sentiment carrying adjective or verb applies. Additionally, words that cannot carry sentiment can be filtered; only adjectives, adverbs, verbs and nouns carry sentiment [2, 12, 15].

Negation (NEG) Linguistic negation is the process that turns an affirmative statement (“I like A”) into the opposite denial (“I do not like A”). In general, negation is done by the inclusion of a negation keyword (e.g., “not” or “never”), but negation can also be achieved using clauses like “but” (“Feature A is excellent, but feature B . . . ”). An important aspect in negation is the identification of the sentiment carrying word the negation applies to [12].

Amplification (AMP) The process of increasing or decreasing the sentiment score of a word, when it is combined with an amplification word, is typically done by multiplying the sentiment score of the word by the amplification score of the amplification word. For example, the positive sentiment score of “beautiful” would be increased by multiplying it with the amplification score of “very” in “very beautiful”.

Comparison (COMP) An author’s sentiment on a topic, relative to his sentiment on another topic can be determined by means of comparison (e.g., “A is better than B”). In sentiment analysis, the absolute sentiment of the author on a topic is typically extracted (“A is good”). Relative comparative sentiment analysis can only be converted to absolute values if an absolute sentiment analysis can be done. For example, if the sentiment score on A can be determined, and if we know that “A is better than B”, we can deduce that the sentiment score on B must be lower than the sentiment score on A, with an amount depending on the strength of the comparison.

Syntactical variants (SYN) Reducing the variability in the forms of words as much as possible can increase the accuracy of word counts. Words can, for example, have alternative spellings or spelling errors. Also verbs, adjectives, and adverbs can be transformed grammatically. Stemming and lemmatizing are techniques to bring back transformed words to their base form (e.g., bring “loved” in “I loved it” back to its stem “love”).

Based on these features, the state-of-the-art in sentiment analysis can be characterized. Table 1 presents an overview of several recent approaches. All three OASYS document scoring algorithms introduced by Ceserano et al. [3] use a wordbank created through supervised learning. All methods do TRF; the Topic-Focussed (TF) algorithm only handles sentences that contain a reference to the topic, the Distance-Weighted Topic-Focussed algorithm (DTWF) gives more weight to sentiment near topic keywords (and is hence a window-level approach), and the Template-Based algorithm (TB) only handles sentences that match a certain template (e.g.., sentence structure or keywords). Besides POS tagging, the OASYS algorithms do not support any other features.

Lerman et al. [10] propose three sentence-level sentiment summarization algorithms: Sentiment Match (SM), Sentiment Match and Aspect Coverage (SMAC), and Sentiment Aspect Match (SAM). The algorithms compute a textual summary of the input document, where sentences are selected to maximize total sentiment in the summary. The algorithms use a wordbank, which is created by related word expansion from a manually annotated base collection using WordNet [6]. By selecting sentences with the highest absolute sentiment score, objective sentences are filtered out. All algorithms filter for topic relevance, where SMAC and SAM use a more advanced, feature-based approach.

The Adverb-Adjective Combinations (AACs) proposed by Benamara et al. [2] use a linguistic analysis of adverbs of degree, which modify adjectives (e.g., “very beautiful”). The algorithms – Variable Scoring (VS), Adjective Priority Scoring (APS), and Adverb First Scoring (AFS) – vary in how they weight the adverb amplification scores. The AAC framework builds on the OASYS framework. The differences between the original OASYS implementation and the AAC implementation are that the AAC implementation supports negation and amplification, and requires a second wordbank containing adverb amplification scores.

Ding et al. [5] propose a holistic lexicon-based sentence-level sentiment analysis approach. Their Opinion Observer (OO) handles context-dependent opinion words and deals with many special words, phrases and language constructs which impact opinions through their linguistic patterns. OO uses a wordbank that is created using related work expansion (via WordNet) on a small set of manually annotated words. Subjectivity filtering is done by ignoring sentences that do not contain sentiment-carrying words. Linguistic negation is recognized through negation words, which include traditional keywords (e.g., “not”) and pattern-based negations such as “stop” + verb + “ing” (e.g., “stop liking”).

Rather than using a wordbank with absolute word sentiment scores, the Class Sequential Rules (CSR) approach proposed by Jindal and Liu [9] uses sequential pattern mining to identify sub-sequences of text that occur more often than a minimum support threshold. The patterns used as features consist of POS tags and one or more comparative key phrases. The sentiment orientation of the key phrases determines the orientation of a sequential pattern. For example, the phrase “Intel is better than AMD” yields the comparative pattern {{proper noun} {third person singular present tense verb} {“better”, comparative adjective} {subordinating preposition or conjunction} {proper noun}}.

Table 1 Classification of algorithms.

Algorithm	WB	SSL	TRF	SF	POS	NEG	AMP	COM	SYN
TF [3]	yes	sentence	yes	no	yes	no	no	no	no
DWTF [3]	yes	window	yes	no	yes	no	no	no	no
TB [3]	yes	sentence	yes	no	yes	no	no	no	no
SM [10]	yes	sentence	yes	yes	no	no	no	no	no
SMAC [10]	yes	sentence	yes	yes	no	no	no	no	no
SAM [10]	yes	sentence	yes	yes	no	no	no	no	no
VS [2]	yes	sentence	yes	no	yes	yes	no	no	no
APS [2]	yes	sentence	yes	no	yes	yes	no	no	no
AFS [2]	yes	sentence	yes	no	yes	yes	no	no	no
OO [5]	yes	sentence	yes	yes	yes	yes	no	no	no
CSR [9]	no	document	yes	no	yes	no	no	yes	no
EVAL [11]	yes	document	no	no	no	no	no	no	no

The Evaluate document algorithm (EVAL) proposed by Liu [11] implements a very basic sentiment analysis framework. It works on the document level and sums up all the individual word sentiment scores, stored in a wordbank, to compute the document score. It does not support any of our classification properties.

Most approaches agree that adjectives and adverbs carry the most sentiment. The role of negations has been explored only to a limited extent. Therefore, we propose to shed some light onto the impact of accounting for negation in sentiment analysis.

3 Sentiment Negation

In order to assess the impact of sentiment negation, we propose a very simple sentiment analysis framework, similar to Liu [11]. This framework consists of wordbank creation and subsequent lexicon-based document scoring. Both parts have optional support for sentiment negation. We classify a document as either positive (1), neutral (0), or negative (-1). The score range of individual words is [-1, 1]. Our framework focuses on adjectives, as adjectives are the best indicators of sentiment [2, 12, 15].

3.1 Framework

The first part of our framework facilitates wordbank creation, involving scoring sentiment of individual words (adjectives) w in a training corpus D_{train}. Our word scoring function is based on a pseudo-expected value function [3]. The sentiment score of any adjective w, $\text{score}(w)$, is based on its total relative influence on the sentiment over all documents $d \in D_w$, where $D_w \subseteq D$, with each document containing w:

$$\text{score}(w) = \frac{\sum_{d \in D_w} \text{score}(d) \times \inf(w, d, neg)}{|D_w|}, \tag{1}$$

where $\text{score}(d)$ is an individual document d's manually assigned score, $|D_w|$ is the number of documents in D_w, and $\inf(w,d,neg)$ is the relative influence of an adjective w in document d, with a Boolean *neg* indicating whether to account for negation or not. This influence is calculated as the count $\text{freq}(w,d,neg)$ of w in d in relation to the total frequency $\sum_{w' \in d} \text{freq}(w',d,neg)$ of all sentiment carrying words w' in d:

$$\inf(w,d,neg) = \frac{\text{freq}(w,d,neg)}{\sum_{w' \in d} \text{freq}(w',d,neg)}. \tag{2}$$

In order to support negation in our framework, we use a variation of Hu and Liu's method [7] of negation. Yet, even though optimizing the scope of influence of negation words [8, 13, 14] or exploitation of compositional semantics in sentiment-bearing expressions [4] has its merits, we first focus on a one-word scope for negation words in an attempt to tease out the effects of accounting for even the simplest forms of negation, as opposed to not accounting for negation at all. We only handle negation words that precede a sentiment word, as larger distances might cause noise in our results due to erroneously negated words. Support for negation is considered in the frequency computations by subtracting the number of negated occurrences of word w in d from the number of non-negated occurrences of w in d.

In the second part of our framework, the score $\text{eval}(d)$ of a document d containing n adjectives $\{w_1, w_2, \ldots, w_n\}$ is simply computed as the sum of the scores of the individual adjectives (the same adjective can appear multiple times), as determined using (1) and (2). In case negation is accounted for, we propose to use a document scoring function based in the scoring function presented by Liu [11]:

$$\text{eval}(d) = \sum_{w_i \in d} (-1)^{\text{negated}(w_i,d)} \times \text{score}(w_i), \tag{3}$$

where $\text{negated}(w_i,d)$ is a Boolean indicating whether the ith adjective in w is negated in d (1) or not (0). Using (3), the classification $\text{class}(d)$ of a document d can finally be determined as follows:

$$\text{class}(d) = \begin{cases} 1 & \text{if eval}(d) > 0.002, \\ 0 & \text{if } -0.021 \leq \text{ eval}(d) \leq 0.002, \\ -1 & \text{if eval}(d) < -0.021. \end{cases} \tag{4}$$

In order to determine the optimal thresholds for (4), we have experimented with different values for the upper and lower threshold. For the upper threshold we have experimented with values between 0.001 and 0.5 with a step-size of 0.005. For the lower threshold, we have experimented with values between -0.001 and -0.5 with a step of 0.005. The ranges between which we tested were determined by manual analysis, in which we found results to decrease rapidly outside interval $[-0.5, 0.5]$.

3.2 Implementation

We have implemented our framework in C#, combined with a Microsoft SQL Server database. We have used a corpus of 13,628 human-rated Dutch documents on 40 different topics. Sentiment in these documents is classified as positive, negative, or neutral. In order to be able to asses the impact of negation, we have implemented two versions of our framework. The first version has no support for negation, whereas the second version supports negation both in the wordbank creation and in the document scoring part. Our framework only handles adjectives for sentiment analysis and uses a commercial part-of-speech tagger (based on OpenNLP and trained on Dutch corpora) to identify adjectives in the corpus.

We have used 60% of our documents for training and 40% for testing. The training set was used to create wordbanks and to determine the best threshold level for document classification. Our software first uses Algorithm 1 to retrieve all adjectives from the training corpus, where multiple occurrences of an adjective are not allowed. The list of adjectives thus extracted is subsequently used for creating a wordbank, hereby following Algorithm 2, which scores all adjectives occurring more than once in the training set with word scoring function (1). A Boolean variable is used to turn the support for negation on or off. Algorithm 3, in which support for negation can also be toggled, is subsequently used to score documents in accordance with document scoring functions (3) and (4).

3.3 Evaluation

In order to evaluate the human judgements, we took a random sample of 224 documents and rated these for sentiment. We observed 56% strong agreement and 99% weak agreement between our judgement and the human annotations, where strong agreement means an exact match and weak agreement means that one rating is positive or negative, whereas the other is neutral. Interestingly enough, in 17% of the cases where our ratings do not strongly agree, human raters appear to tie sentiment to the consequences of facts, which we call "factual sentiment". For example, the

Algorithm 1: Creating a list of adjectives.

input : A training corpus D_{train}
output: A list *wordList* of all adjectives in D_{train}

$wordList = \emptyset$;
foreach d *in* D_{train} **do**
    $adjList$ = `getAdj(`d`); // Retrieve all adjectives in` d
    foreach adj *in* $adjList$ **do**
        if $adj \notin wordList$ **then** $wordList = \{wordList, adj\}$;
    end
end

Algorithm 2: Creating a wordbank.

```
input : A training corpus D_train, a list wordList of all adjectives in D_train, and a Boolean neg
        indicating whether to account for negation
output: A list wordbank containing all adjectives in D_train with their scores
wordbank = ∅;
foreach w in wordList do
    |D_w| = 0; // Number of documents containing w
    sumWContr = 0; // Sum of all contributions of w in D_w
    foreach d in D_train do
        // Retrieve frequency of w in d, minus negated
        // occurrences, if neg
        freqWD = freq(w, d, neg); // Number of occurrences of w in d
        scoreD = getScore(d); // Human annotators' score for d
        if freqWD > 0 then
            |D_w| = |D_w| + 1;
            if scoreD ≠ 0 then
                sumAllWD = 0; // Count of all words of wordList in d
                foreach w' in wordList do
                    sumAllWD = sumAllWD + freq(w', d, neg);
                end
                infWD = freqWD / sumAllWD; // Influence of w in d
                sumWContr = sumWContr + (infWD × scoreD);
            end
        end
    end
    if |D_w| > 1 then
        scoreW = sumWContr / |D_w|;
        wordbank = {wordbank, {w, scoreW}};
    end
end
```

objective and hence neutral statement "Stock prices for our company went down 2% today" is judged as carrying (negative) sentiment. Another explanation for the discrepancies between ratings are interpretation differences. It is for instance difficult for humans to pick up on subtle cases of sentiment, which can be expressed in irony and tone. The interpretation of such subtle uses of sentiment can differ from person to person. The two cases of strong disagreement are due to misinterpretation of the text.

Additionally, we have evaluated the performance of our framework against human ratings in two set-ups: one with support for negation and one without support for negation. Precision improves with 1.17% from 70.41% without taking into account negation to 71.23% when accounting for negation. This observed improvement is even more evident when our framework is applied to a subset of the corpus, each document of which contains negated words (not necessarily adjectives). On this subset, precision increases with 2.23% from 69.44% without accounting for negation to 70.98% when taking into account negation. These results are notable given that only 0.85% of the sentences in the original corpus contain negations.

Algorithm 3: Scoring a document.

input : A list *wordbank* containing all adjectives in the training corpus with their scores, an upper threshold *utreshold* indicating the score above which a document is considered to be positive, a lower document score threshold *lthreshold* below which a document is considered to be negative, a Boolean *neg* indicating whether to account for negation, and a document *d*

output: A document score *result*

```
result = 0; // Final score for document d, initialized as neutral
docScore = 0; // Score for document d
adjList = getAdj(d); // Retrieve all adjectives in d
foreach adj in adjList do
    if adj ∈ wordbank then
        if neg then docscore = docscore + (−1)^isNegated(adj) × score(adj);
        else docscore = docscore + score(adj);
    end
end
if docscore > uthreshold then result = 1;
else if docscore < lthreshold then result = −1;
```

4 Conclusions and Future Work

The main contribution of this paper is two-fold. First of all, we have provided a characterization of current approaches to sentiment analysis, based on their wordbank type, sentiment scoring level, topic relevance filtering, subjectivity filtering, part-of-speech tagging, negation, amplification, comparison, and type variations. In this analysis, it has become apparent that the role of negations in sentiment analysis has been explored only to a limited extent.

The second contribution of this paper lies in our reported endeavors of shedding some light onto the impact of accounting for negation in sentiment analysis. Firstly, we have found that human raters tend to rate the consequences of factual information as carrying sentiment; an observation that may be taken into account in future work. Furthermore, our experiments with a basic sentiment analysis framework show that a relatively straightforward approach to accounting for negation already helps to increase precision with 1.17%. On a subset of selected documents containing negated words, precision increases with 2.23%. This is a notable result if we consider the fact that negation is sparsely used in our data set.

Nevertheless, it appears to be worthwhile to investigate the effects of optimizing the scope of influence of negation words in order to obtain more detailed insights in the impact of negation in sentiment analysis. We would also like to experiment with other types of words in our wordbank (e.g., adverbs, possibly combined with adjectives). Finally, we plan on taking into account degrees of negation. For instance, "not bad" is not necessarily "good", yet more likely slightly less positive than "good". All in all, a rather simple way of accounting for negation in sentiment analysis already helps to improve performance, yet we envisage that future work in the suggested directions could advance the state-of-the-art in sentiment analysis.

Acknowledgements

We would like to thank Teezir (`http://www.teezir.com`) for their technical support, fruitful discussions, and for supplying us with data for this research.

References

1. Bautin, M., Vijayarenu, L., Skiena, S.: International Sentiment Analysis for News and Blogs. In: 2nd International Conference on Weblogs and Social Media (ICWSM 2008), pp. 19–26. AAAI Press (2008)
2. Benamara, F., Cesarano, C., Reforgatio, D.: Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone. In: 1st International Conference on Weblogs and Social Media (ICWSM 2007), pp. 203–206. AAAI Press (2007)
3. Cesarano, C., Dorr, B., Picariello, A., Reforgiato, D., Sagoff, A., Subrahmanian, V.: OASYS: An Opinion Analysis System. In: AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (CAAW 2006). pp. 21–26, AAAI Press (2006)
4. Choi, Y., Cardie, C.: Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis. In: 13th Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 793–801, ACL (2008)
5. Ding, X., Lu, B., Yu, P.S.: A Holistic Lexicon-Based Approach to Opinion Mining. In: 1st ACM International Conference on Web Search and Web Data Mining (WSDM 2008), pp. 231–240. ACM (2008)
6. Fellbaum, C.D.: WordNet: An Electronic Lexical Database. MIT Press (1998)
7. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 168–177. ACM (2004)
8. Jia, L., Yu, C., Meng, W.: The Effect of Negation on Sentiment Analysis and Retrieval Effectiveness. In: 18th ACM Conference on Information and Knowledge Management (CIKM 2009), pp. 1827–1830. ACM (2009)
9. Jindal, N., Liu, B.: Identifying Comparative Sentences in Text Documents. In: 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006), pp. 244–251. ACM (2006)
10. Lerman, K., Blair-Goldensohn, S., McDonald, R.: Sentiment Summarization: Evaluating and Learning User Preferences. In: 12th Conference of the European Chapter of the ACL (EACL 2009), pp. 514–522. ACL (2009)
11. Liu, B.: Web Data Mining. Springer (2007)
12. Liu, B.: Handbook of Natural Language Processing, chap. Sentiment Analysis and Subjectivity, pp. 627–667. CRC Press, 2nd edn. (2010)
13. Morante, R., Liekens, A., Daelemans, W.: Learning the Scope of Negation in Biomedical Texts. In: 13th Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 715–724. ACL (2008)
14. Morante, R., Daelemans, W.: A Metalearning Approach to Processing the Scope of Negation. In: Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009), pp. 21–29. ACL (2009)
15. Wiebe, J.M.: Learning Subjective Adjectives from Corpora. In: 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI 2000), pp. 735–740. AAAI Press (2000)