Analyzing Sentiment while Accounting for Negation Scope and Strength

Alexander Hogenboom Paul van Iterson Bas Heerschop
Flavius Frasincar Uzay Kaymak

Erasmus University Rotterdam
P.O. Box 1738, NL-3000 DR, Rotterdam, The Netherlands
hogenboom@ese.eur.nl, {paulvaniterson, basheerschop}@gmail.com, {frasincar, kaymak}@ese.eur.nl

The full version of this paper, entitled Determining Negation Scope and Strength in Sentiment Analysis appeared in: Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2011), IEEE Computer Society, Forthcoming 2011.

Abstract

Recent developments in automated sentiment analysis show a tendency of accounting for various aspects other than word frequencies. One of these aspects is negation. 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 turns out to be to consider the first two words, following a negation keyword, to be negated by that keyword. Additionally, we propose to optimize the sentiment modification in case of negation to a value of –1.27 rather than –1.

1 Introduction

The Web offers a vast amount of textual data, containing traces of valuable information – sentiment. Typically, the goal of automated sentiment analysis is to automatically determine the polarity of natural language texts. Most existing approaches are based on frequencies of positive and negative words. Yet, one may also account for other aspects like negation by, e.g., inverting the polarity of negated words. A major challenge in this respect is determining which words are negated to what extent by a negation keyword. Several approaches to optimizing the scope of influence of a negation keyword have already been proposed, yet so far, the impact of these approaches as such has neither been assessed nor compared. Therefore, we aim to provide measurements and comparisons of these methods and additionally account for negation strength.

2 Framework

In order to address our research goals, we propose a basic sentence-level sentiment analysis framework which uses word-level sentiment scores ranging from -1 (negative) to 1 (positive) to classify sentences as either positive or negative. First, we determine POS types, lemmas, and word senses for each word in a sentence. For the Word Sense Disambiguation (WSD) process, we propose to use a freely available Lesk algorithm [2], which iteratively selects the word sense that is semantically most similar to the context. We then retrieve sentiment scores from SentiWordNet [1], i.e., by subtracting negativity scores from positivity scores. Then, we handle negation by modifying (i.e., inverting) word-level sentiment scores if their associated words are influenced by a negation keyword. Finally, if the sum of word-level sentiment scores is smaller than 0, the sentence is classified as negative, else, the sentence is classified as positive.

We consider several negation scope determination approaches. 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. Third, we consider negating only the sentiment of the Next Non-Adverb (NNA) following a negation keyword, while assuming adverbs to typically modify other (sentiment carrying) words. Fourth, 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. Additionally, as negation may modify sentiment strength, we also support Modified Inversion Strength (MIS), where the inversion factor for any negation scope determination method may be anything in the range $[-2, 0]$.

3 Evaluation

In order 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 our framework in C#. For POS tagging, we use a commercial tagger provided to us by Teezir[1]. For lemmatization and WSD, we use the open-source WordNet.Net API[2]. The sentiment lexicon used in our framework is SentiWordNet 3.0 [1].

We evaluated the performance of our considered approaches on a collection of $10{,}662$ sentences from English movie reviews, which have been rated for sentiment [3]. We selected $2{,}285$ sentences (930 positive and $1{,}355$ negative) that contain one or more of our considered negation keywords. We used 60% of this set for training and 40% for testing purposes. Each approach was assessed on our test set 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 considered $k \in \{1, 2, 3, 4\}$. Additionally, for the best performing approach, we optimized the sentiment inversion factor by means of a hill-climbing procedure on our training set. We compared the performance of all considered approaches to the performance of a baseline without any support for negation.

The best performing method turns out to be to consider the first two words following a negation keyword to be negated by that keyword. When using this method, overall accuracy and macro-level F_1 significantly increase with 5.5% and 6.2%, respectively, 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 increase in accuracy and macro-level F_1 of 7.0% and 8.0%, respectively, compared to not taking into account negation.

4 Conclusions and Future Work

Properly accounting for negation in automated sentiment analysis may help improve the performance of classifying text as carrying either positive or negative sentiment. On our data set of English movie review sentences, the best approach turns out to be to consider two words following a negation keyword to be negated by that keyword. Optimizing the sentiment modification of negated words to a value of -1.27 rather than -1 further improves performance. As future research, we would like to incorporate a deeper understanding of semantics, as the interpretation of negation may be context-dependent – "not missing" something bad can imply a positive sentiment, whereas "not missing" something good may be negative.

Acknowledgments

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

References

[1] A. Esuli, S. Baccianella, 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)*, pages 2200–2204. ELRA, 2010.

[2] 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)*, pages 24–26. ACM, 1986.

[3] 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)*, pages 271–280. ACL, 2004.

[1] http://www.teezir.com/

[2] http://opensource.ebswift.com/WordNet.Net/