Semantically Oriented Sentiment Mining in Location-Based Social Network Spaces

Carlone, Domenico; Ortiz-Arroyo, Daniel

Published in:
Lecture Notes in Computer Science

DOI (link to publication from Publisher):
10.1007/978-3-642-24764-4_21

Publication date:
2011

Document Version
Early version, also known as pre-print

Semantically Oriented Sentiment Mining in Location-Based Social Network Spaces

Domenico Carlone and Daniel Ortiz-Arroyo

Computational Intelligence and Security Laboratory,
Department of Electronic Systems,
Aalborg University,
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
dcarlo09@student.aau.dk, do@cs.aaue.dk

Abstract. In this paper we describe a system that performs sentiment classification of reviews from social network sites using natural language techniques. The pattern-based method used in our system, applies classification rules for positive or negative sentiments depending on its overall score, calculated with the aid of SentiWordNet. We investigate several classifier models created from a combination of different methods applied at word and review levels. Our experimental results show that using part-of-speech helps to achieve better accuracy.

Keywords: Opinion Mining, Sentiment Classification, SentiWordNet, Social Networks.

1 Introduction

Sentiment analysis or opinion mining is an emerging discipline within the fields of information retrieval and natural language processing (NLP). Sentiment analysis consists in detecting the subjectivity and sentiments contained in general opinions. Opinions are expressions that describe the emotions and feelings of people regarding a subject, entity or event [1]. Conversely, facts are objective descriptions.

Sentiment analysis has many applications. For instance, it can be applied to understand people's attitudes for marketing analysis purposes. Moreover, the automatic detection of opinions can be used to substitute surveys and questionnaires. Finally, the Internet can be used as a source of information on people's opinions about products, services, events, or political topics. For this purpose, social network sites provide a convenient way to share opinions.

Many studies have been carried out on sentiment-based classification within the field of sentiment analysis. However, few of these studies have been performed in the domain of reviews posted in social network sites. In this paper we present an algorithm for mining opinions from some social network sites such as Foursquare, Yelp, Qype, Where, CitySearch. These sites are mainly concerned with describing "interesting" places within cities. In these social networks users

H. Christiansen et al. (Eds.): FQAS 2011, LNAI 7022, pp. 234–245, 2011.

post opinions about clubs, events or restaurants and some of their features such as food quality, customer satisfaction or atmosphere.

Our system is capable of collecting and classifying user's opinions by identifying their semantic orientation. Reviews retrieved from social networks sites are classified based on the presence of certain terms that are likely to express sentiments. Opinions are classified as belonging to one of two opposing polarities: positive or negative [3]. In order to apply our classification method, the text from reviews is preprocessed using Natural Language Processing (NLP) techniques.

Since opinions frequently express the strength of a person's feelings with respect to some subject, our method associates a degree of positivity or negativity to each review/comment. This is done to obtain a ranked list of reviews for the best places. The effectiveness of the proposed system is evaluated in terms of precision, recall, F-measure and overall accuracy.

This paper is organized as follows. Section 2 presents a summary of related work. Section 3 describes in detail our system and in section 4 we present some experimental results. Finally section 5 provides our conclusions and describes future work.

2 Related Work

Some recent research work in sentiment analysis focuses on designing methods to determine the sentiment contained in documents. Other research focuses on more specific tasks, such as finding the sentiments of words [4] or searching for subjective expressions [5].

Machine learning and semantic orientation analysis are some of the methods applied in sentiment analysis. The former employs, for instance, well known probabilistic algorithms such as Naive Bayes (NB) [6]. The latest is a rule-based (or pattern-based) approach that applies Natural Language Processing (NLP) [2] techniques and external linguistic resources. One of the main differences between these two approaches lies in the need to use a training phase, in the case of supervised machine learning. The two approaches have been combined in a hybrid solution as is described in [8] and [9].

Methods based on the application of NLP-based techniques, extract phrases containing opinions using predefined part-of-speech (POS) patterns. In [7], Turney et al. use POS tagging to extract two-words phrases from reviews containing at least one adjective or one adverb. The semantic orientation is estimated assigning a score. Then an average is calculated with the scores obtained with the sentences and phrases contained in the reviews. Turney's work and others such as [4] found that there is a high correlation between the presence of adjectives and a sentence's subjectivity.

Other studies demonstrate that other parts of speech such as nouns and verbs are also good indicatives of sentiment [10]. In a similar work Pang et al. [6], examine three different machine learning methods for sentiment classification: Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy. They found that best performance was obtained when SVM was used in combination with unigrams, reaching a maximum accuracy of 83%.

The two techniques most commonly used in sentiment classification based on a semantic orientated approach are corpus-based and dictionary-based techniques. Within the former approach, Turney in [7] calculated the semantic orientation of a phrase using point-wise mutual information. This method essentially calculates the probability of collocations between the terms contained in a phrase and two reference words such as excellent and poor, that are representative of positive and negative polarities. Conversely, dictionary-based techniques utilize dictionaries and sentiment lexicons that provide information about semantic relations between words and terms' sentiment properties to determine overall sentiment of opinions.

The problem of identifying sentiment in text can be addressed by determining the subjectivity or semantic orientation (i.e. polarity) it contains. Lexicons addressing the former tasks are called *subjectivity lexicons* as they provide lists of subjective words. An example of this approach is introduced in [5]. Other lexicons include the prior polarity of words, such as Harvard General Inquirer (GI), Micro-WNOp, and SentiWordNet [11]. The first two include prior polarities together with indicators (i.e. adjectives) of term attitudes (e.g. "strong negative" or "weak positive"). SentiWordNet on the other hand, determines the degrees of words' polarities within the range [0,1]. SentiWordNet includes an evaluation no only of the positivity and negativity of a word but also its objectivity.

Other research work has applied SentiWordNet to the problem of automatically classifying sentiment. For instance, Pera, Qumsiyeh and Ng [12] introduced a domain independent sentiment classifier which categorizes reviews on the base of their semantic, syntactic, and sentiment content. To calculate the overall sentiment score of a review, the proposed classifier determines first the polarity score of each word contained in it; thereafter, it calculates the review's sentiment orientation by subtracting the sum of its negative words scores from the sum of its positive words scores.

Thet et al. in [13], proposed a linguistic approach for sentiment analysis of message posts on discussion boards, in which it is performed clause-level sentiment analysis. Firstly, they calculate the prior words' sentiment scores, employing SentiWordNet in combination with a lexicon from the domain of movie reviews especially built for the purpose. Then, they determine the contextual sentiment score for each clause by analyzing grammatical dependencies of words (through dependency trees) and handling pattern-rules.

In [14] Denecke tested rule-based and machine learning models in a multi-domain, classification scenario. Their results confirmed that the lexicon-based approach that made use of SentiWordNet had limited accuracy compared to the machine learning method.

Few studies have combined semantic orientation and machine learning approaches to improve Sentiment Classification performance. Ohana and Tierney [15] compared two approaches to assess the performance of using of SentiWordNet to the task of sentiment classification at document level on film reviews. In the first method, the lexicon was applied to count the positive and negative terms found in a document. Sentiment orientation was determined based on which class received the highest score, similarly as was done in the methods described in [6] and [16]. Later, term scores were used to determine sentiment orientation. The second method in [15] employed SentiWordNet as a source of positive and negative features. These features were used to train a SVM supervised learning algorithm that showed an improvement in accuracy.

3 System Description

Our sentiment classification system for location-based social network sites performs a series of steps, starting with the collection of reviews from social network sites and ending up performing sentiment analysis and classification. Fig. 1 illustrates the steps performed by our system.

The dataset used in our experiments was extracted directly from social network sites, given that most of the datasets available in the domain of sentiment analysis belong to movie reviews and that no dataset was available in our domain. In this work we used "Yelp" and "Foursquare" sites as the data sources. Our dataset consists in geo-coded place reviews collected from these sites.

Reviews and other information contained in Yelp and Foursquare's repositories were extracted by sending requests to their Web Services APIs. The retrieved information was about reviews on certain interesting places and locations. Then the data was stored in a database to ease its access.

Reviews were then processed through several stages using NLP steps as depicted in Fig. 1. First, tokens were extracted one at a time and then normalized using rules specifically designed for the English language. For instance, short forms' expansion was employed to eliminate contractions. Terms were also transformed to lowercase for easing the searching for entries in the SentiWordNet database. For the same previous reason, words were brought to their base form through lemmatization.

Tokens were then tagged so that they could be used in the SentiWordNet lexicon. POS tagging was used to identify words, corresponding to parts-of-speech, that are good predictors of the sentiment expressed in sentences. If a lexicon entry corresponding to the analyzed token was found in SentiWordNet, the token score algorithm was applied. Then, the resulting token score was sent to our *Prior-Polarity Classifier* to be used in the calculation of the review's sentiment Score. Next, the token score algorithm used by our classifier model was applied. Finally, our classifier determines if the processed review was positive, negative or objective. Following sections describe the most important stages of our system with more detail.

3.1 Classification Algorithm

Our classification algorithm takes as input all the normalized tokens coming out from the pre-processing phase, together with the related part-of-speech tags assigned. The collection of terms involved in calculating a review score is reduced to tokenized words that belong to one of the four POS classes of SentiWordNet (adjectives, adverbs, nouns, verbs).

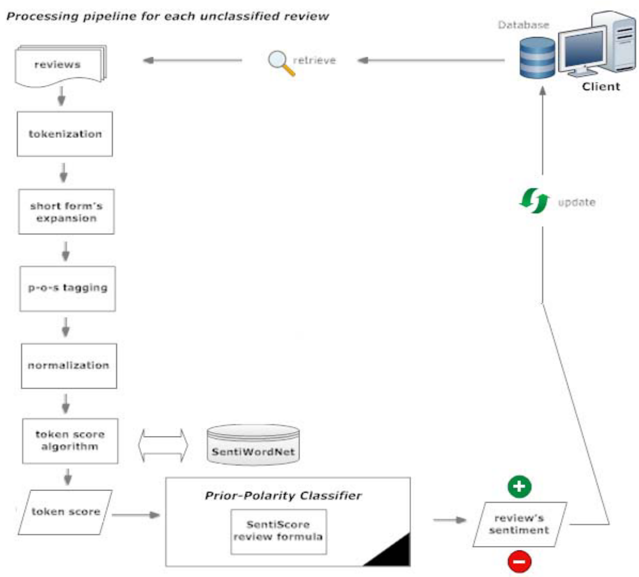

Fig. 1. Sentiment Analysis Pipeline

Words in Natural Language can be polysemous and because of their multiple meanings, tokens can have multiple entries in SentiWordNet. Consequently, in order to assign the polarity score to a word, it is first necessary to perform *Word Sense Disambiguation (WSD)*. However, our system in the current state does not applies any Word Sense Disambiguation method. Alternatively, to determine what is the effect of the different meanings of a word in the overall sentiment, we used a simple statistical approach. For each word, all possible senses are collected together with the three corresponding polarity scores: positive, negative and objective. Then, we applied and evaluated the following three strategies for the calculation of the final triple of token scores, namely:

- *Random Sense*
- *All Senses Arithmetic Mean*
- *POS matching Senses Arithmetic Mean*

The first method consists in the random selection of a sense among all possible senses of a word. This is the simplest approach that intuitively should show worst performance. The second method is the arithmetic mean of each of the three polarity scores computed, on all the possible senses, i.e. is an average of the sentiment entries of a word for all possible POS taggings. The third method is also an average of the sentiment entries of a word, but in this case the entries used to calculate the average are only those that match the POS tag assigned in the pre-processing phase. Therefore, in this method not all senses are considered but only the senses of the words found in SentiWordNet that match the computed

POS tag; if more than one sense belongs to the subset obtained after the POS tagging filtering, then the arithmetic mean is applied.

Each of the previous three scoring methods is applied to the three possible polarities: positive, negative, objective. At the end of this step, for each token we will have the following nine different scores:

– *Random Sense Score: Pos, Neg, Obj*
– *All Senses Arithmetic Mean Score: Pos, Neg, Obj*
– *POS matching Senses Arithmetic Mean Score: Pos, Neg, Obj*

The last six arithmetic mean scores are calculated with the formula:

$$score_{pol}(T) = \frac{1}{n}\sum_{s=1}^{n} score_{pol(s)} \tag{1}$$

where $pol \in \{pos, neg, obj\}$, and n is the number of the *s senses (synsets)* corresponding to the SentiWordNet entries for the *token T*. As was explained before, in the case of POS matching senses arithmetic mean score, n is reduced to the subset of all the senses in SentiWordNet that match the computed POS tag.

We applied the three methods obtaining a final triple score for positive, negative and objective scores. The approach we applied is similar to the one reported in [14] where positive and negative SentiWordNet scores for a term are compared. If the positivity (or negativity) is larger, the word is considered positive (or negative, respectively) and its strength is represented by its positivity (or negativity) score. If both values are equal, the word is ignored, since the interest is toward opinionated words. The objective value is only taken into account in the case we want to apply a cutoff value in order to exclude, from the computation of the overall sentiment review score, words that are too "objective".

To calculate the overall sentiment score of a review R we subtracted the sum of the scores of its negative words from the sum of its positive words scores as is shown in equation 2:

$$SentiScore(R) = \frac{\sum_{pos=0}^{j} Score(Token_{pos}) - \sum_{neg=0}^{k} Score(Token_{neg})}{j+k} \tag{2}$$

where j and k are the number of positive and negative words in R respectively, *Token* is a word in R, *Score(Token)* is the highest SentiWordNet score of the word considering the positive and negative scores.

Since large reviews can contain more or less positive or negative words, the different numbers may impact the sentiment score. For this reason *SentiScore* is normalized by dividing it by the number of sentiment words in R. Normalization keeps values within the interval [-1,+1].

Finally if the *SentiScore(R)* obtained by equation 2 is higher (lower) than zero, then a review R is labeled as positive (negative); when *SentiScore(R)* is zero, it means that the score of positive words equals the score of negative words, in this case the review is considered objective.

3.2 Classifier Models

Our classifier employs the *SentiScore* equation 2 to classify reviews. However, we decided to apply several classifier models to investigate their effect in accuracy. The first model considers the inclusion of nouns in the estimation of the SentiScore. Words in SentiWordNet are partitioned into adjectives, adverbs, nouns and verbs. Sometimes nouns are judged to be objective words and in some research work they are completely excluded.

The second model consists in applying a cutoff to the objective score of a token to exclude, from the computation of the *SentiScore*, words that have a high degree of objectivity. It has to be noted that in SentiWordNet a word can be simultaneously positive, negative and objective. In fact, most SentiWordNet's words have an objective score grater than zero, even if they are positives or negatives. We decided that a word is considered polarized if its $ObjScore(T) < 1 - cutoff$. The reason to use this condition is because in SentiWordNet the summation of the positive, negative, objective scores for a term is 1, and the objective score results from the complement of positive plus negative scores. For instance applying a cutoff of 0.3 will exclude those words whose objective score is higher than 0.7. With this cutoff value we expressly allow to include words whose polarity is objective in the computation. Since our reviews are very short, applying a high cutoff limit together with the condition of POS tag matching, may reduce the number of words considered polarized to either a very small number or even zero. For words that pass the cutoff condition, the algorithm compares its positive and negative scores with the *SentiScore* formula.

The algorithm used to compute the semantic orientation of a word is shown below. POS can be restricted to just {verbs, adverbs, adjectives} in case we choose not to consider noun's POS senses.

```
for each Token = POS
consider the Score Triple calculated using a chosen score Method

  if ObjScore(T) > 1-(cutoff):
   do not include word in the SentiScore computation

  else
   if PosScore(T) > NegScore(T):
   add Token,Scores(Token) to positive set

   if NegScore(T) > PosScore(T):
   add Token,Scores(Token) to negative set

   if PosScore(T) = NegScore(T):
   do not include word in the SentiScore computation

end for each
Perform Sentiscore computation using tokes' positive and negative scores
```

Note that if the positivity and negativity values of a word are equal, the difference between the polarity scores will be zero; therefore the word will not be included in the computation of the *SentiScore* formula since we are interested in opinionated words.

The algorithm that is most similar to our approach is presented in [8], where the overall sentiment score is calculated applying a classification rule. Conversely, in [14] the number of positive, negative and objective words is calculated and their values compared to classify a review. In both, the strategy for the calculation of the token scores consists in the arithmetic mean executed on the triAple scores for all the term's senses found. A cutoff value is applied in [3].

4 Experimentation

In order to discover the best classification model, several criteria were applied at both, token and review levels. Similarly, we used several cutoff points, as was done in [3], where the best accuracy was reached with a 0.8 $cutoff$. In [3] only words that have a positive or negative polarity greater than the established cutoff are considered. However, when a cutoff point of 0.8 is used, the size of the SentiWordNet lexicon is reduced from $52,902$ to 924. This approach is too strict to be applied in the short reviews we have in our dataset. Therefore, we decided to experiment using two lower cut-off values of 0.3 and 0.5. The rule we applied is that a token T, belonging to a review R, is considered in the computation of its *SentiScore(R)* if $ObjScore(T) < 1 - cutoff$.

As a result of our experiments, we obtained 18 different sentiment scores for each review, corresponding to the 18 classifier models produced by combining the 3 token scoring methods with 6 different review scoring methods.

The system was implemented in Python using MySQL database and the open source library Natural Language ToolKit (NLTK)[1] for tokenization and part-of-speech tagging.

4.1 Dataset

Experiments were conducted on a dataset consisting of both 400 and 200 positive reviews additionally to 200 negative reviews. The reviews used are a subset of the whole data collected during the opinion extraction phase of our system.

It must be noted that Yelp's reviews are rated on a 5-point scale, with 1 being the most negative and 5 being the most positive. We decided to convert these favorability ratings into a polarity corresponding to one of the three sentiment categories (positive, negative, neutral), to being able to use them during testing. Since each review has a rating based on the number of stars (1 to 5), we decided to use 1 or 2 as a negative rating and 4 or 5 as positive one. Opinions marked with 3 stars are considered neutral (objective) and therefore excluded from the evaluation. As it is suggested in [6] and [10], ratings in terms of the number of stars, can be used as indicator of the overall sentiment of reviewers.

4.2 Evaluation Metrics and Results

The effectiveness of the system was evaluated in terms of Precision, Recall, F-measure and overall Accuracy.

The contingency Table 1 shows true positives and true negatives as the correct classifications. Precision and recall metrics are split in *Positive Precision* ($Prec_p$)

[1] `http://www.nltk.org/`

Table 1. Relevance/Retrieval contingency table

	Relevant	Nonrelevant
Retrieved	*true positives (tp)*	*false positives (fp)*
Not Retrieved	*false negatives (fn)*	*true negatives (tn)*

Table 2. Equations for Precision and Recall

	Positive	Negative
Precision	$Prec_p = tp/(tp + fp)$	$Prec_n = tn/(tn + fn)$
Recall	$Rec_p = tp/(tp + fn)$	$Rec_n = fn/(tp + fn)$

Table 3. Evaluation of results: Precision

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
$Prec_n$	cutoff=0	48,5%	55%	56,5%
	cutoff=0, no-nouns=true	49%	53%	53,5%
	cutoff=0.3	52,5%	56%	57,5%
	cutoff=0.3, no-nouns=true	48,5%	52%	55%
	cutoff=0.5	52%	54%	50,5%
	cutoff=0.5, no-nouns=true	49%	50%	48,5%
$Prec_p$	cutoff=0	65,5%	68%	65%
	cutoff=0, no-nouns=true	61%	68%	65,5%
	cutoff=0.3	49,5%	55%	53%
	cutoff=0.3, no-nouns=true	44%	47%	52%
	cutoff=0.5	32%	32%	35%
	cutoff=0.5, no-nouns=true	29,5%	26,5%	35%

and *Positive Recall* (Rec_p), *Negative Precision* ($Prec_n$), and *Negative Recall* (Rec_n) as shown in Table 2.

The *Accuracy* of the system is given by:

$$\textbf{Accuracy} = (tp + tn)/(tp + fp + fn + tn) \tag{3}$$

Precision and Recall are combined in the *F-measure*:

$$\textbf{F-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Tables 3, 4, 5, 6, summarize classifier's performance in terms of precision, recall, F-measure, and overall accuracy respectively. These tables show the results we obtained using different methods at token and review level and that the best results were obtained using no cut-off and a token score method based on POS matching senses.

The differences in performance between reviews' classification on positive and negative opinions, as measured by precision, recall and F-Measure (e.g. Table 3, Table 4, Table 5), may be attributed to a cause mentioned in [15]. [15] describes that reviewers generally include negative remarks on positive opinions to provide

Table 4. Evaluation results: Recall

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
Rec_n	cutoff=0	44%	39,82%	40,092%
	cutoff=0, no-nouns=true	45,54%	40,87%	41,518%
	cutoff=0.3	48,97%	44,44%	44,5%
	cutoff=0.3, no-nouns=true	53,93%	50,526%	46,4%
	cutoff=0.5	60%	41,02%	58,58%
	cutoff=0.5, no-nouns=true	63,35%	65,36%	59,5376%
Rec_p	cutoff=0	55,98%	60,18%	59,907%
	cutoff=0, no-nouns=true	54,464%	58,8745%	58,482%
	cutoff=0.3	51%	55,555%	55,497%
	cutoff=0.3, no-nouns=true	46,07%	49,47%	53,608%
	cutoff=0.5	40%	41,025%	41,42%
	cutoff=0.5, no-nouns=true	36,646%	34,64%	40,462%

Table 5. Evaluation results: F-Measure

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
$F - score_n$	cut-off=0	46,14%	46,195%	46,9%
	cut-off=0, no-nouns=true	47,2%	46,15%	46,75%
	cut-off=0.3	50,67%	49,555%	50,17%
	cut-off=0.3, no-nouns=true	51,071%	51,252%	50,335%
	cut-off=0.5	55,714%	44,9%	54,24%
	cut-off=0.5, no-nouns=true	55,26%	56,657%	53,455%
$F - score_p$	cut-off=0	60,37%	63,8514%	62,35%
	cut-off=0, no-nouns=true	57,547%	63,106%	61,8%
	cut-off=0.3	50,24%	55,276%	54,22%
	cut-off=0.3, no-nouns=true	45,011%	48,203%	52,79%
	cut-off=0.5	35,555%	35,955%	37,94%
	cut-off=0.5, no-nouns=true	32,7%	30,03%	37,53%
$F - score$	cutoff=0	53,255%	55,0232%	54,625%
	cutoff=0, no-nouns=true	52,373%	54,628%	54,275%
	cutoff=0.3	50,455%	52,4155%	52,195%
	cutoff=0.3, no-nouns=true	48,041%	49,7275%	51,5625%
	cutoff=0.5	45,6345%	40,4275%	46,09%
	cutoff=0.5, no-nouns=true	43,98%	43,3435%	45,4925%

Table 6. Evaluation results: Accuracy

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
Accuracy	cutoff=0	57%	61,5%	60,75%
	cutoff=0, no-nouns=true	55%	60,5%	59,5%
	cutoff=0.3	51%	55,5%	55,25%
	cutoff=0.3, no-nouns=true	46,25%	49,5%	53,5%
	cutoff=0.5	42%	43%	42,75%
	cutoff=0.5, no-nouns=true	39,25%	38,25%	41,75%

a more balanced assessment. Additionally reviewers may choose to build up the expectation of a general good view to ended up giving a negative impression.

One of the reasons for the low accuracies in Table(6) may be due to the limited number of opinionated words contained in the short reviews collected. Our results also show that classifier's performance decreases when the cutoff value is increased. The reason may be also due to the short reviews given that applying a high cutoff, together with the condition of POS tag matching, reduces the number of words considered polarized to either zero or a very small number. This reduces the information at classifier's disposal to correctly determine a review's sentiment orientation.

5 Conclusions and Future Work

In this paper we have described a rule-base classifier model that exploits SentiWordNet. Our model was applied to classify reviews of interesting places extracted from social network sites. Our method achieved an accuracy in classification comparable to those obtained by previous similar systems e.g. [8], [14], [3]. However, since these systems were evaluated using different datasets in different domains, no direct comparison can be performed at this time.

A number of factors affect the performance of our classifier. For instance, the use of ironic words or colloquial language makes difficult to determine the polarity of a review. Additionally, most of the errors we obtained came from the wrong assignment of prior sentiment scores to words. For instance, words that have certain polarity in SentiWordNet may have a different polarity within the context of a review. Other inaccuracies come from the assignment of part-of-speech tags; for example, in a phrase such as "What a cool place", the term cool is wrongly tagged as proper noun, and consequently identified by SentiWordNet as being objective, instead of being positive (adjective). More imprecisions may come from SentiWordNet itself since it has been found that some words have the wrong scores assigned. As future work we consider using word sense disambiguation and combining our rule-based method with a machine-learning approach.

References

1. Liu, B.: Sentiment analysis and subjectivity. In: Indurkhya, N., Damerau, F.J. (eds.) Handbook of Natural Language Processing, 2nd edn. CRC Press, Taylor and Francis Group (2010) ISBN 978-1420085921
2. Nasukawa, T., Yi, J.: Sentiment analysis: capturing favorability using natural language processing. In: K-CAP 2003: Proceedings of The 2nd International Conference on Knowledge Capture, pp. 70–77. ACM, New York (2003)
3. Mejova, Y.: Tapping into sociological lexicons for sentiment polarity classification. In: 4th Russian Summer School in Information Retrieval, pp. 14–27 (September 2010)

4. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics, pp. 174–181 (1997)
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT 2005: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354 (2005)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP 2002: Proceedings of the ACL-2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
7. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424 (2002)
8. Denecke, K.: Using sentiwordnet for multilingual sentiment analysis. In: ICDE Workshops 2008, pp. 507–512 (2008)
9. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. Journal of Informetrics 3(2), 143–157 (2009)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. 2(1-2), 1–135 (2008)
11. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta (May 2010)
12. Pera, M.S., Qumsiyeh, R., Ng, Y.-K.: An unsupervised sentiment classifier on summarized or full reviews. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 142–156. Springer, Heidelberg (2010)
13. Thet, T.T., Na, J.-C., Khoo, C.S.G., Shakthikumar, S.: Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In: Proceeding of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA 2009, pp. 81–84. ACM, New York (2009)
14. Denecke, K.: Are sentiwordnet scores suited for multidomain sentiment classification? In: Fourth IEEE International Conference on Digital Information Management, ICDIM 2009, November 1-4, pp. 33–38. University of Michigan, IEEE, Ann Arbor, Michigan (2009)
15. Ohana, B., Tierney, B.: Sentiment classification of reviews using SentiWordNet. In: 9th. IT & T Conference, p. 13 (2009)
16. Kennedy, A., Inkpen, D.: Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. Computational Intelligence 22(2), 110–125 (2006)