Sentiment Mining for Natural Language Documents

Alexander O'Neill

COMP3006 Project Report

Department of Computer Science
Australian National University

November 2009

Abstract

The wide variety of possible applications for sentiment mining has made it the focus of considerable research in recent years. High accuracy classification has been achieved by using a variety of techniques, most of which are heavily reliant on machine learning.

At its core, sentiment mining involves correct interpretation of natural language, and all the complexities and challenges involved in such a task. Despite this, the published work on the topic shows a surprising lack of work on the application of natural language based techniques to the problem.

This report details an attempt to apply Natural Language Processing (NLP) techniques to the problem of accurately classifying the sentiment a document contains relating to a certain subject. Common NLP methods that are implemented in pursuit this goal include part of speech tagging, grammatical structure parsing, and coreference resolution.

Combining these NLP approaches gave classification accuracy of 83% on the test corpus, and suggest with improved coreference resolution handling this could be improved on even further.

1 Introduction

The automation of extracting opinions from text has become an area of growing interest in recent years. Due to the increasing amount of user-generated content available on the Web, the ability to accurately gauge opinions has more practical applications than ever before. Businesses spend huge amounts of money in an attempt to determine how consumers feel towards products and services, and any unfulfilled desires of theirs. An automated system that was able to provide such information would both save money and provide greater reach. The individual could benefit from the aggregated opinions of others on different products, services, etc. Stock market analysis[1], political polling, and advertisement targeting are other examples of applications where sentiment analysis could be used.

This report limits its scope to one of the subfields of sentiment analysis, and focuses on sentiment classification, the classification of a subjective document as expressing a positive, neutral or negative sentiment towards some target (product, person, idea, etc.)[2].

Sentiment analysis is a complex field. As it involves the processing and interpretation of natural language, it must deal with natural language's inherently ambiguous nature, the importance of context, and other complications that do not lend themselves to automation. To give an idea of the challenge involved, below are outlined some of the issues that need to be dealt with.

> *"Just go read the book."*

This demonstrates how important context can be. If mentioned regarding a book, this could be considered a recommendation, while if it is in reference to a film adaption of a book, it would seem to suggest the film is not worth watching.

> *"Tarantino's latest film is an instant classic."*

Once again context dependent, as "Tarantino's latest film" will likely change over time.

> *"The new Nokia is so small, it will fit in any pocket."*
>
> *"The serving sizes there are tiny."*

Whether being small is a desirable property is another example of context dependence.

> *"I enjoyed this play like I enjoy a trip to the dentist."*

In common English usage a trip to the dentist has negative connotations, but this would be difficult for a computer to know.

"It is the Ferrari of the motorcycling world."

This involves a comparison, the sentiment of which is impossible to analyse without some kind of knowledge about Ferrari. Since possibilities for comparison are practically infinite, dealing with cases such as this are difficult for a computer.

"The new chassis looks so sick."

Slang terms come and go, and their sentiment can be difficult to determine.

"... leaving the government with a huge budget deficit. Looking forward to a tax rise? I know I am."

"This, the ultimate street racer, outputs over 50kW from it's high-tech 1.3L Korean engine."

Detecting sarcasm is a problem that would seem to be well beyond current technologies.

There are a number of different approaches that have been used in an attempt to 'solve' the problem of sentiment classification. One of the more widely used methods involves classifying single words or phrases with a sentiment, and then calculating an overall sentiment rating for a target based using some weighting. Probably the most widespread and best performing method involve the use of machine learning techniques to classify unknown documents. For a topic so intrinsically linked with natural language, the use of linguistics in sentiment classification is surprisingly limited, though there are some cases where it has been applied successfully[3]. It should be noted though that even when linguistic methods are used, it is often in combination with machine learning techniques.

This report details the implementation of a classifier that uses a variety of Natural Language Processing (NLP) techniques in an attempt to improve classification accuracy. These techniques will be used to classify a test set of 100 natural language documents and the results compared with a baseline classifier, to determine whether the use of underlying language structure can provide an advantage in sentiment classification.

2 Background

2.1 Sentiment Analysis

Given recent trends, it is interesting that one of the earliest influential papers on the subject involved the use of NLP techniques. The paper by Turney[4] used part of speech tagging (an NLP technique) to estimate a 'semantic orientation' of meaningful phrases in Epinions reviews. The entire review was then classified as either positive or negative by averaging the semantic orientation of the phrases within. This approach gave a classification accuracy of 74%. Yi et al[5] also outlines a classifier that uses part of speech tagging, with good results on a wide range of different sentiment sources.

Pang et al.[6] studied the application of a number of different machine learning techniques to sentiment classification. A Naive Bayes classifier, maximum entropy and support vector machines were used classify movie reviews as either positive or negative. Using unigrams, the support vector machine gave nearly 83% accuracy, and was further improved on in a later paper[7]. Logistic regression and clustering are among the other machine learning techniques that can be found in sentiment analysis literature.

Others have approached the sentiment mining problem from a different angle, believing that the complexity of natural language make the existence of a general solution to sentiment problems unlikely. These instead focus on specific sentence or text types, such as in the case of Liu et al [3] who use linguistics in an attempt to classify for 'conditional sentences', once again in combination with machine learning techniques.

While machine learning methods have been found to produce good results, there are associated disadvantages. Machine learning classification is dependent on the training set used, so while a number of papers detail classifiers with high accuracy, they are often tested on only a single kind of sentiment source (movie review is the most common). In such cases there is little indication of how the classifier would perform in more general cases. The gathering of such a training set is also arduous, as it involves the gathering and human classification of a huge number of different documents. The impracticality of gathering such a data set is the reason no machine learning techniques are offered for comparison in this report.

2.2 Natural Language Processing

There are a number of different language analysis techniques that fall under the umbrella of Natural Language Processing (NLP), of which only a very limited subset have occured with regularity in sentiment mining literature. Part of speech tagging is the most commonly encountered, though there are also papers detailing classifiers using coreference resolution[8], and even the use of a full syntactic parse tree[9].

Part of speech (POS) tagging is the process of labelling word occurrences with its 'world class', for example whether a word is occuring as an adjective, noun or verb. Effective tagging requires knowledge of not just the word but also its context, such as position within the sentence and surrounding words. Hidden Markov models are a common technique used in POS tagging, though the Stanford POS tagger uses a maximum entropy technique[10].

To avoid constantly referencing a subject by name, natural languages usually contain alternative words that can be used when referring to a previously mentioned subject. For example:

"John has proven what a great actor he is, once again playing his role perfectly."

Here, the words "he" and "his" both relate to John. Automating the process of connecting such references is called coreference resolution, and previous studies have shown that effectively applying the technique to sentiment analysis can improve classification accuracy by as much as 10%[8].

Creating parse trees for natural language sentences is another central area of study in NLP. Due to the ambiguity of natural language there is often multiple valid parse trees for a given sentence, so require probabilistic techniques to be used. Parsing is related to POS tagging as determining sentence structure requires knowledge of which sense words are being used in. 'Chunking', a simplified form of parsing that does not analyse sentences in as much depth, can be used in place of parsing for some applications. Nakusawa et al[9] propose one method of how parsing can be applied to sentiment analysis.

2.3 Tools

2.3.1 The Stanford Parser

The Stanford parser[11] is a natural language parser developed by The Stanford Natural Language Processing Group. It uses probabilistic methods to work out parse trees for sentences, to represent their grammatical structure. The parser requires training, using some collection of pre-annotated data. One commonly used data set is available from the Penn Treebank Project[12]. The POS tags used by the Penn Treebank Project are shown in Table 1.

The techniques employed by the parser are further detailed by Klein & Manning[13, 14]. For an example of the output provided by the Stanford parser refer to Figure 1.

```
(ROOT
  (S
    (S
      (NP (PRP I))
      (VP (VBD thought)
        (SBAR
          (S
            (NP (DT the) (JJ first) (NN season))
            (VP (VBD was)
              (ADJP (JJ outstanding)))))))
    (, ,)
    (S
      (NP (DT the) (JJ second) (NN season))
      (VP (VBD was)
        (ADJP (JJ mediocre)
          (PP (IN at)
            (ADJP (JJS best))))))
    (CC and)
    (S
      (NP (DT the) (JJ third) (NN season))
      (VP (VBD was)
        (ADJP (RB no) (JJ good))
        (PP (IN at)
          (NP (DT all)))))
    (. .)))
```

Figure 1: An example parse tree produced by the Stanford parser

Tag	Part of speech
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential *there*
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possesive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	*to*
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possesive wh-pronoun
WRB	Wh-adverb

Table 1: Penn Treebank part of speech tags

2.3.2 SentiWordNet

The classification techniques detailed in this report require the ability to calculate sentiment on a per word basis. SentiWordNet[15] provides this functionality. Based on WordNet[16], SentiWordNet associates three scores with each synset (usually a word plus POS tag or a simple phrase) in WordNet: objectivity, positivity and negativity.

2.3.3 Coreference resolution

As with many areas of NLP, the tools available for coreference resolution are far from perfect. While both Lingpipe[17] and OpenNLP[18] provide coreference resolution methods, testing revealed that their performance is very limited outside of resolving references to people. Due to this, both implementations were discarded as options for use in general purpose sentiment classification.

3 Approach

Each of the different classifiers implemented require the assigning of sentiment to individual words, for which SentiWordNet was used.

All of the NLP techniques implemented make use of information provided by using the Stanford parser to parse the text according to its grammatical structure.

3.1 Baseline

The baseline classifier has been implemented as a benchmark for NLP based classifiers. It uses a simple neighbourhood proximity based approach, whereby each word in the text is assigned a sentiment by SentiWordNet. As this approach is not intended to make use of NLP techniques, SentiWordNet averages sentiment entries of the word for all possible POS taggings.

For each occurrence of the target word in the text, every word within a given range has its sentiment assigned, and a weight calculated based on the distance from the target. A Gaussian curve is used to calculate a weighting for each distance,better emphasising the importance to be placed on words occuring closer to the target. The sentiment classification is given by the sum of the weighted individual word sentiments for each target occurrence.

Figure 2 gives an example of how weightings would be assigned. It should be noted that sentiment is calculated for each target occurrence independently before summation, so if a word occurs within range of two target occurrences the two weightings will be additive. This use of Gaussian weighting is used in all classification methods that do not determine dependency based on the sentence parse tree.

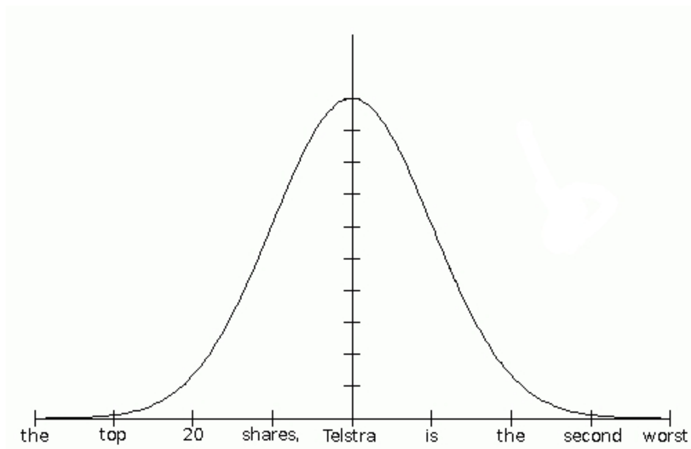

Figure 2: Gaussian weighting with target Telstra

3.2 Part-of-Speech Tagging

Parsing by the Stanford parser provides full POS tagging. The Penn Treebank POS tags were split into groups that correspond to adjectives, nouns and verbs. Using these tags, individual word sentiment calculation by SentiWordNet could be refined to only consider meanings of words that matched their POS tag.

In addition, words with POS tags that did not fall into any of these categories could be disregarded when calculating sentiment, as they would be highly unlikely to express any meaningful sentiment.

3.3 Dependency Analysis

Dependency analysis using the parse tree was used to determine what scope different words and phrases had within a sentence.

With target occurrences occuring in noun phrases, words were only considered as relating to the target if they satisfied at least one of the following:

1. The word is contained within a noun phrase where the target occurs.

2. The word is contained within a phrase with the same parent node as a noun phrase where the target occurs, and the phrase occurs after the target in the text.

All words that are marked as referring to the target are given equal weight, and the summation of individual word sentiments is given as the overall sentiment for the target.

Figure 3 shows an example of this technique in practice. The parse tree has been produced by the Stanford parser. Using dependency analysis would give blue words as referring to Alan, red words referring to Barry and green words as referring to Charles.

```
(ROOT
  (S
    (S
      (S
        (NP (NNP Alan))
        (VP (VBZ is)
          (ADJP (JJ useless))))
      (CC and)
      (S
        (NP (NNP Barry))
        (VP (VBZ is)
          (ADJP (RB no) (JJ good)))
        (. .)))
    (NP
      (NP (NNP Charles) (POS '))
      (NN performance))
    (VP (VBD was)
      (ADJP (JJ excellent))
      (ADVP (RB however)))
    (. .)))
```

Figure 3: Dependency analysis example

The obvious drawback of this method is the difficulty of connecting words that occur before the target. A new method would need to be developed to extend the dependency analysis to such cases. An example of a sentence that would require this is given in Figure 4.

```
(ROOT
  (S
    (PP (IN For)
      (NP (PRP me)))
    (NP (DT the) (JJS best) (NN movie))
    (NP (DT this) (NN year))
    (VP (MD would)
      (VP (VB be)
        (NP (DT The) (NNP Departed))))
    (. .)))
```

Figure 4: Example of where dependency analysis can fail

3.4 Negation

Negation was used in classifiers both with and without parse tree based dependency analysis. It uses a list of words that are considered to negate sentiment.

When used in a neighbourhood proximity based approach, the occurrence of a negating word reverses the polarity of all words occuring afterwards within the scope of a sentence.

When used with dependency analysis, the occurrence of a negating word reverses the polarity of all words for which the parent of the negating word is an ancestor (here the parent of the negating word is taken to mean the parent of the node containing the POS tag of the negating words). In such cases nested negating words are capable of reversing polarity of sentiment again, back to its original polarity.

This method does not deal with some of the more difficult cases of natural language negation, as some words that negate sentiment are not included in the list as the negation is dependent on their usage. For example

> "*The film avoids the usual cliches found in the genre.*"

While in this instance avoids has the effect of negating the sentiment of cliches, it also occurs in many sentences that does not modify sentiment.

An example of negation in practice can be seen in Figure 5, where negated sentiment words are shown in red.

```
(ROOT
  (S
    (NP (DT The) (NNP Mona) (NNP Lisa))
    (VP (VBZ is)
      (NP (DT no) (NN masterpiece)))
    (. .)))
```

Figure 5: An example of negation being used

3.5 Coreference resolution

As noted previously, the available tools tested were insufficient for coreference resolution in cases not relating to people. Such serious shortcomings in even state of the art NLP technologies gives an idea of how difficult coreference resolution is.

As the technology has been shown to provide significant performance improvements in cases where it can be applied[12] an attempt was made to implement a basic coreferencing system. It makes use of a list of words that indicate a reference to another subject, and the POS tags that correspond to such usages. It does not take into account context, so coreferencing a subject other than the target is not detected. It's a very crude method, used purely to demonstrate the possible improvements that coreferencing could make. Refinements could be made to this technique to improve it, but without use of context it would still be inherently limited in the complexity it could deal with.

4 Experiments

4.1 Test Corpus

The test corpus required human gathering and classification. It contains 100 different files, labelled with a single target and the sentiment the text expresses towards that target. 50 of the files have a positive sentiment, 50 have a negative sentiment. When gathering the corpus care was taken to ensure all files have at least one reference to the target, though there is often more. Furthermore, all files use proper grammar, punctuation, spelling and capitalisation. While a lot of potential material on the Web probably wouldn't meet such requirements, NLP analysis of such cases would likely be fruitless.

Around 90% of the corpus is comprised of reviews from the amazon.com. Most of these relate to either books or movies, though not exclusively. The remainder contains a mixture of reviews and opinion pieces on different topics. Length of test files ranges from a single sentence to multiple paragraphs, though on average it's around 2-3 sentences. Most of the test files have some property that makes them difficult to classify, to better differentiate the classification techniques.

4.2 Results

	1	2	3	4	5	6	7	8	9	10	11
Baseline	46	50	48	50	48	49	50	50	50	49	49
POS	36	49	58	57	57	63	61	62	61	60	60
Negation	55	55	55	58	62	63	63	65	63	64	64
POS+Neg.	54	66	69	71	72	72	72	73	72	72	73

Table 2: Neighbourhood proximity classifier accuracy using varying range

	Accuracy	Precision	Recall	F-score
Baseline (8)	50.0	50.0	44.0	46.8
POS (8)	62.0	61.5	64.0	62.7
Negation (8)	65.0	54.1	66.0	59.5
POS+Neg. (8)	73.0	76.7	66.0	70.9
Dependency	61.0	50.8	66.0	57.4
Dep. + POS	65.0	53.6	74.0	62.2
Dep. + Neg	68.0	55.1	86.0	67.2
Dep.+POS+Neg.	83.0	81.1	86.0	83.5
Dep.+POS+Neg+Coref.	87.0	83.6	92.0	87.6

Table 3: Classifier accuracy, precision, recall and F-score

Baseline (8)	0
POS (8)	0
Negation (8)	0
POS+Neg. (8)	0
Dependency	4
Dep. + POS	4
Dep. + Neg	4
Dep.+POS+Neg.	4
Dep.+POS+Neg+Coref.	4

Table 4: Number of files given a sentiment score of zero

4.3 Analysis

Table 2 shows how the various neighbourhood proximity based approaches performed on the corpus. It first bears mentioning that all classifiers will assign a positive sentiment to a file by default, i.e. if the calculated sentiment score is zero. This is the cause of the surprisingly high accuracy seen for small neighbourhood ranges.

Another general result across the neighbourhood proximity classifiers is that they all give near-best performance using a range of around 8 (this is 8 words to each side, or 16 words total). The average sentence length in the English language has been found to be around 15-20 words, and so this optimal range corresponds closely with the most common sentence length.

The baseline classifier ended up performing no better than random on this data set, and while it is based on a simple principle, this is still a surprising result. The improvements given by the NLP assisted neighbourhood proximity approaches (and POS in particular) show that the proximity principle still has merits.

Applying either POS tagging or the simple negation principles here both give greater than 10% improvements. That the POS method outperformed the baseline so significantly shows that single word sentiment scoring was letting down the baseline, and indicates that trying to classify the sentiment of words without at least some idea of its usage is futile.

The combination of negation and POS tagging outperformed either individually, as would be hoped, and by a significant margin. This is further validation of the proximity principle, and that the improvements shown by POS tagging and negation were not one-off cases.

Using dependency analysis alone gave an accuracy of 61%. As the issues of POS, negation and coreferencing are not dealt with here, this improvement is not insignificant. Adding POS and negation individually gave improvements of 4% and 7% respectively. The improvements are not as significant as those seen when using the neighbourhood proximity approach, but in both cases the results are better using dependency analysis instead of proximity.

It should be noted that all these dependency based approaches show recall being higher than accuracy. Taken on its own might take this to indicate that dependency analysis is failing in a lot of the cases, assigning a sentiment score of 0, and defaulting to a positive classification. However we see from Table 4 that only four files are given a sentiment score of 0. Likewise the similar recall score with much higher accuracy seen when dependency, negation and POS tagging techniques are combined provides further evidence against such a theory.

The accuracy of 83% seen in that case is a large improvement over that of any of the simpler classifiers. The combination of the improved focus provided by dependency analysis, the improved single word sentiment provided by POS tagging, and the improved handling of the special (though widespread) case of negation comes together to perform better than what would be expected from the earlier results. When using proximity, the combination of POS tagging and negation gave an improvement of 8-11% over either individually, while with dependency sees a 15-18% improvement. Additionally, POS tagging and negation give 4% and 7% improvements respectively when used with dependency analysis, yet when used together they give a 22% improvement, double the sum of the improvements.

The final classifier was the result of adding coreference resolution. This only resulted in a 4% accuracy increase, however it does indicate there is scope for improvement using NLP techniques alone, specifically more refined coreference resolution. While including coreference resolution in this case gave better performance, its primitive nature makes it unsuitable for more general application (larger text size analysis where there are more potential subjects for sentiment would expose the simplicity of the algorithm used), and should be treated more as a proof of concept than anything else.

5 Further Work

The results contained within this report already provide some direction for possible classifier extensions.

The most obvious are extending the dependency analysis to deal with sentiment expressed about the target before the target has occured in text, and the integration of an advanced coreference resolution system, to identify more passages that might refer to a target.

Less obvious but potentially beneficial directions for further research include the identification of phrases that express a sentiment that cannot be determined from the combination of words in it alone. SentiWordNet already provides the capability to give a sentiment for some simple phrases. Another is an improved method for handling sentiment negation, either determining the scope of negation or how to treat negated words (simply negating the sentiment is simplistic, e.g. "great" has high positive sentiment, whereas "not great" has a relatively low negative sentiment). Finally, both the Stanford parser and SentiWordNet are probabilistic tools. While this is taken into account to a degree with SentiWordNet (by averaging the sentiment of possible word meanings), the Stanford parser provides multiple parse trees with a corresponding probability. Some method of incorporating alternative parse trees and their probabilities could further improve performance.

One drawback of an NLP based approach is that it would likely perform very poorly when used on grammatically incorrect text. As much of the sentiment bearing content available on the Internet could fall into that category, methods to detect and possibly correct bad English would be necessary before use on a larger scale.

6 Conclusion

The three main techniques used to improve classification were POS tagging, dependency analysis through parse tree structure, and the treatment of negation in sentences. The combination of these techniques provided a classifier that was able to give 83% accuracy on the test corpus. This was a huge improvement on the 50% accuracy given by the neighbourhood proximity based classifier that was implemented as a baseline.

The results show consistent gains from using these NLP based techniques, as in every case classification was improved by applying the technique, no matter the combination used. Applying all the techniques gave a performance improvement that exceeded what would be expected from the results when any sub-combination was used.

Also promising was the 87% accuracy that was achieved when coreference resolution was added to the classifier. Given the scope for improvement in handling coreferences, adding a more advanced treatment of this NLP technique would likely give even better performance.

References

[1] Koppel, M., Shtrimberg, I. (2004) “Good News or Bad News? Let the Market Decide”. AAAI Spring Symposium on Exploring Attitude and Affect in Text. Palo Alto, p. 86-88. : AAAI Press.

[2] Whitelaw, Casey, Garg, Navendu & Argamon, Shlomo (2005) “Using Appraisal groups for sentiment analysis”. Proceedings of ACM SIGIR Conference on Information and Knowledge Management. Bremen, p. 625-631. CIKM

[3] Narayanan, Ramanthan, Liu, Bing & Choudhary, Alok (2009) “Sentiment Analysis of Conditional Sentences”. Proceedings of Conference on Empirical Methods in Natural Language Processing, Singapore.

[4] Turney, P., (2002) “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews”. Proceedings of the 40th Annual Meeting on Association For Computational Linguistics. New Jersey, p. 417-424.

[5] Jeonghee Yi, Nasukawa, Tetsuya, Bunescu, Razvan, Niblack, Wayne, (2003) "Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques," icdm, , Third IEEE International Conference on Data Mining (ICDM’03). Florida, pp.427

[6] Pang, B., Lee, L., and Vaithyanathan, S. (2002) “Thumbs up?: sentiment classification using machine learning techniques.” Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, New Jersey, p. 79-86

[7] Pang, B. and Lee, L. (2004). “A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts.” Proceedings of the 42nd Annual Meeting on Association For Computational Linguistics, Barcelona

[8] Nicolov, N.; Salvetti, F.; and Ivanova, S. (2008). “Sentiment Analysis: Does Coreference Matter?” AISB 2008 Convention Communication, Interaction and Social Intelligence.

[9] Kanayama, H., Nasukawa T. and Watanabe, H. (2004) "Deeper Sentiment Analysis Using Machine Translation Technology,", In Proceedings of the 20th International Conference on Computational Linguistics

[10] http://nlp.stanford.edu/software/tagger.shtml

[11] http://nlp.stanford.edu/software/lex-parser.shtml

[12] www.cis.upenn.edu/~treebank/

[13] Manning, Christopher D., Klein, Dan (2003) "Fast Exact Inference with a Factored Model for Natural Language Parsing." Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press, pp. 3-10.

[14] Manning, Christopher D., Klein, Dan (2003) "Accurate Unlexicalized Parsing." Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

[15] Esuli, Andrea & Sebastiani, Fabrizio (2006) "SENTIWORDNET: A publicly available lexical resource for opinion mining." In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06), Genova.

[16] http://wordnet.princeton.edu/

[17] http://alias-i.com/lingpipe/

[18] http://opennlp.sourceforge.net/