Sentiment classification of online political discussions: a comparison of a word-based and dependency-based method

Hugo Lewi Hammer
Oslo and Akershus University College
Department of Computer Science
hugo.hammer@hioa.no

Per Erik Solberg
Språkbanken
The National Library of Norway
p.e.solberg@ifikk.uio.no

Lilja Øvrelid
Department of Informatics
University of Oslo
liljao@ifi.uio.no

Abstract

Online political discussions have received a lot of attention over the past years. In this paper we compare two sentiment lexicon approaches to classify the sentiment of sentences from political discussions. The first approach is based on applying the number of words between the target and the sentiment words to weight the sentence sentiment score. The second approach is based on using the shortest paths between target and sentiment words in a dependency graph and linguistically motivated syntactic patterns expressed as dependency paths. The methods are tested on a corpus of sentences from online Norwegian political discussions. The results show that the method based on dependency graphs performs significantly better than the word-based approach.

1 Introduction

Over the past years online political discussions have received a lot of attention. E.g. the Obama 2012 election team initiated an extensive use of text analytics and machine learning techniques towards online material to guide advertising campaigns, identifying key voters, and improve fundraising (Issenberg, 2012). There has also been a lot of concern about the alarming growth in hate and racism against minorities like Muslims, Jews and Gypsies in online discussions (Goodwin et al., 2013; Bartlett et al., 2013). Sentiment analysis (SA) is the discipline of automatically determining sentiment in text material and may be one important tool in understanding the diversity of opinions on the Internet.

In this paper we focus on classifying the sentiment towards religious/political topics, say the Quran, in Norwegian political discussion. We use a lexicon-based approach where we classify the sentiment of a sentence based on the polarity of sentiment words in relation to a set of target words in the sentence. We expect that statistically the importance of a sentiment word towards the target word is related to the number of words between the sentiment and target word as suggested by Ding et al. (2008). Information about the syntactic environment of certain words or phrases has in previous work also been shown to be useful for the task of sentiment classification (Wilson et al., 2009; Jiang et al., 2011). In this work we therefore compare the results obtained using a token-based distance measure with a novel syntax-based distance measure obtained using dependency graphs and further augmented with linguistically motivated syntactic patterns expressed as dependency paths. In order to evaluate the proposed methods, we furthermore present a freely available corpus of Norwegian political discussion related to religion and immigration, which has been manually annotated for the sentiment expressed towards a set of target words, as well as a manually translated sentiment lexicon.

2 Previous work

Sentiment classification aims to classify a document or sentence as either positive or negative and sometimes also neutral. There are mainly two approaches, one based on machine learning and one based on using a list of words with given sentiment scores (lexicon-based approach). For machine learning any existing method can be used, e.g. naïve Bayes and support vector machine, (Joachims, 1999; Shawe-Taylor and Cristianini, 2000). One simple lexicon-based approach is to count the number of words with positive and negative sentiment in the document as suggested by Hu and Liu (2004). One may classify the opinion of larger documents like movie or product reviews or smaller documents like tweets, comments

Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 90–96,
Baltimore, Maryland, USA. June 27, 2014.

or sentences. See Liu (2012), chapters three to five and references therein for the description of several opinion classification methods.

SA has mostly been used to analyze opinions in comments and reviews about commercial products, but there are also examples of SA towards political tweets and discussions, see e.g. Tumasjan et al. (2010); Chen et al. (2010). SA of political discussions is known to be a difficult task since citations, irony and sarcasm is very common (Liu, 2012).

3 Proposed SA methods

In this section we present two methods to classify sentences as either positive, neutral or negative towards a target word. Both methods follow the same general algorithm presented below which is inspired by Ding et al. (2008) and is based on a list of sentiment words each associated with a sentiment score representing the polarity and strength of the sentiment word (sentiment lexicon). Both target words, sentiment words and sentiment shifters can in general appear several times in a sentence. Sentiment shifters are words that potentially shift the sentiment of a sentence from positive to negative or negative to positive. E.g. "not happy" have the opposite polarity than just "happy". Let $tw_i, i \in \{1, 2, \ldots, I\}$ represent appearance number i of the target word in the sentence. Note that we only consider one target word at the time. E.g. if a sentence contains two target words, e.g. Quran and Islam, the sentence is first classified with respect to Quran and then with respect to Islam. Further let $sw_j, j \in \{1, 2, \ldots, J\}$ be appearance number j of a sentiment word in the sentence. Finally let $\mathbf{ss} = (ss_1, ss_2, \ldots, ss_K)$ represent the sentiment shifters in the sentence. We compute a sentiment score, S, for the sentence as follows

$$S = \frac{1}{I} \sum_{i=1}^{I} \sum_{j=1}^{J} \mathbf{imp}(tw_i, sw_j) \mathbf{shift}(sw_j, \mathbf{ss}) \quad (1)$$

where the function **imp** computes the importance of the sentiment word sw_j on the target word appearance tw_i. This will be computed in different ways as described below. Further, the function $\mathbf{shift}(sw_j, \mathbf{ss})$ computes whether the sentiment of sw_j should be shifted based on all the sentiment shifters in the sentence. It returns -1 (sentiment shift) if some of the sentiment shifters are within d_p words in front or d_n words behind sw_j, respectively. Else the function, returns 1 (no sentiment shift). We classify the sentiment towards the target word to be positive, neutral or negative if $S >= t_p$, $t_p > S > t_n$ and $S <= t_n$, respectively. The parameters d_p, d_n, t_p and t_n is tuned using a training set, as described in section 5 below.

3.1 Word distance method

For the word distance method we use the following **imp** function

$$\mathbf{imp}(tw_i, sw_j) = \frac{\mathbf{sentsc}(sw_j)}{\mathbf{worddist}(tw_i, sw_j)} \quad (2)$$

where $\mathbf{sentsc}(sw_j)$ is the sentiment score of sw_j from the sentiment lexicon and $\mathbf{worddist}(tw_i, sw_j)$ is the number of words between tw_i and sw_j in the sentence plus one.

3.2 Parse tree method

When determining the sentiment expressed towards a specific target word, the syntactic environment of this word and how it relates to sentiment-bearing words in the context may clearly be of importance. In the following we present a modification of the scoring function described above to also take into account the syntactic environment of the target words. The function is defined over dependency graphs, i.e. connected, acyclic graphs expressing bilexical relations.

Dependency distance One way of expressing the syntactic environment of a target word with respect to a sentiment word is to determine its distance in the dependency graph. We therefore define a distance function $\mathbf{depdist}(tw_i, sw_j)$ which returns the number of nodes in the shortest dependency path from the target word to the sentiment word in the dependency graph. The shortest path is determined using Dijkstra's shortest path algorithm (Dijkstra, 1959).

Dependency paths A second way of determining the importance of a sentiment word towards a target based on syntactically parsed texts, is to establish a list of grammatical dependency paths between words, and test whether such paths exist between the targets and sentiment words (Jiang et al., 2011). The assumption would be that two words most likely are semantically related to each other if there is a meaningful grammatical relation

between them. Furthermore, it is reasonable to expect than some paths are stronger indicators of the overall sentiment of the sentence than others. To test this method, we have manually created a list of 42 grammatical dependency paths, divided into four groups, and given them a score from $0-1$. The higher the score is, the better indicator of sentiment the path is assumed to be. In the following paragraphs, we will briefly present the groups of paths and the maximum score we have assigned in each group. The paths are represented in the following format: postag-target:postag-sentiment word__DEPREL_up/dn(__DEPREL_up/dn etc.). *Up* and *dn* indicate the direction of the traversed arc in the graph.

A first group consists of paths from subject targets to sentiment predicates. Such paths can e.g. go from a subject to a verbal predicate, subst:verb__SUBJ_up, or from a subject to an adjectival or nominal predicate in the context of a copular verb, subst:adj/subst__SUBJ_up__SPRED_dn. Paths in this group can get the maximum score, 1. The combination of a subject and a predicate will result in a proposition, a statement which is evaluated as true or false. We expect that a proposition typically will represent the opinion of the speaker, although e.g. irony and certain kinds of embedding can shift the truth evaluation in some cases. Secondly, if the predicate represents an event brought about by an intentional agent, the subject will typically represent that agent. If the predicate has a positive or negative sentiment, we expect that this sentiment is directed towards this intentional agent.

A second group we have considered, contains paths from subject targets to sentiment words embedded within the predicate, such as from the subject to the nominal direct object of a verb, subst:subst__SUBJ_up__DOBJ_dn. Paths from subjects into different kinds of adverbials are also a part of this group. We consider paths from subjects to objects to be good indicators of sentiment and assign them the highest score, 1 . The reasoning is much the same as for subject predicate paths: The statement is a proposition and the subject will often be the agent of the event. Also, the object and the verb are presumably closely semantically connected, as the former is an argument of the latter. Paths into adverbials get lower values, as adverbials often are less semantically connected to the predicate than objects.

The paths in our third group go from targets to sentiment words within the predicate. These include paths from nominal direct object target to verbal predicates, subst:verb__DOBJ_up, and from various kinds of adverbials to verbal predicates, etc. We assume that predicate-internal paths are less good indicators of sentiment than the above groups, as such paths do not constitute a proposition. Also, arguments within the predicate usually do not represent intentional agents. Such paths will get the score $1/3$.

Our fourth and final group of dependency paths contains paths internal to the nominal phrase, such as from target nouns to attributive adjectives, subst:adj__ATR_dn, and from target complements of attributive prepositions to target nouns, subst:subst__PUTFYLL_up__ATR_up. A positively or negatively qualified noun will probably often represent the sentiment of the speaker. At the same time, a nominal phrase of this kind can be used in many different contexts where the holder of the sentiment is not the speaker. We assign $2/3$ as the maximum score. Table 1 summarizes the groups of dependency paths.

Path group	Number	Score range
Subj. to pred.	9	1
Subj. to pred.-internal	13	$1/3-1$
Pred.-internal	6	$1/3$
NP-internal	14	$1/3-2/3$

Table 1: Grouping of dependency paths with the number of paths and score range for each group.

Modified scoring function Let $\mathcal{D}$ denote the set of all salient dependency paths. The function $\mathbf{gram}(tw_i, sw_j)$ returns the dependency path, and if $\mathbf{gram}(tw_i, sw_j) \in \mathcal{D}$, then the function $W_{\mathbf{dep}}(tw_i, sw_j) \in [0,1]$, returns the salience score of the path. Further let $\mathbf{depdist}(tw_i, sw_j)$ return the dependency distance, as described above. The **imp** function is computed as follows. If $\mathbf{gram}(tw_i, sw_j) \in \mathcal{D}$ we use

$$\begin{aligned}\mathbf{imp}(tw_i, sw_j) = \\ \alpha \cdot \mathbf{sentsc}(sw_j) W_{\mathbf{dep}}(tw_i, sw_j) \\ +(1-\alpha) \cdot \frac{\mathbf{sentsc}(sw_j)}{\mathbf{depdist}(tw_i, sw_j)}\end{aligned} \quad (3)$$

where $\alpha \in [0,1]$ is a parameter that weights the score from the salient dependency path and the

tree distance and can be tuned using a training set. If $\mathbf{gram}(tw_i, sw_j) \notin \mathcal{D}$ we simply use

$$\mathbf{imp}(tw_i, sw_j) = \frac{\mathbf{sentsc}(sw_j)}{\mathbf{depdist}(tw_i, sw_j)} \quad (4)$$

Note that when $\alpha = 0$, (3) reduces to (4).

4 Linguistic resources

4.1 Sentiment corpus

We did not find any suitable annotated text material related to political discussions in Norwegian and therefore created our own. We manually selected 46 debate articles from the Norwegian online newspapers *NRK Ytring, Dagbladet, Aftenposten, VG and Bergens Tidene*. To each debate article there were attached a discussion thread where readers could express their opinions and feelings towards the content of the debate article. All the text from the debate articles and the subsequent discussions were collected using text scraping (Hammer et al., 2013). The debate articles were related to religion and immigration and we wanted to classify the sentiment towards all forms of the following target words: *islam, muslim, quran, allah, muhammed, imam and mosque*. These represent topics that typically create a lot of active discussions and disagreements.

We automatically divided the material into sentences and all sentences containing at least one target word and one sentiment word were kept for further analysis. If a sentence contained more than one target word, e.g. both Islam and Quran, the sentence was repeated one time for each target word in the final text material. We could then classify the sentiment towards each of the target words in the sentence consecutively. To assure that we do not underestimate the uncertainty in the statistical analysis, we see each repetition of the sentence as the same sentence with respect to the sentence random effect in the regression model in Section 5.1

Each sentence was manually annotated as to whether the commenter was positive, negative or neutral towards the target word in the sentence. Each sentence was evaluated individually. The sentences were annotated based on real-world knowledge, e.g. a sentence like "Muhammed is like Hitler" would be annotated as a negative sentiment towards Muhammed. Further, if a commenter presented a negative fact about the target word, the sentence would be denoted as negative.

	Negative	**Neutral**	**Positive**
Training	174 (46%)	162 (42%)	46 (12%)
Test	102 (33%)	182 (59%)	24 (8%)

Table 2: Manual annotation of training and test set.

In order to assess inter-annotator agreement, a random sample of 65 sentences from the original text material was annotated by a second annotator. These sentences were not included in either the training or test set. For these sentences, the two annotators agreed on 58, which is an 89% agreement, with a 95% confidence interval equal to (79%, 95%) assuming that each sentence is independent. Since the sentences are drawn randomly from the population of all sentences this is a fair assumption.

Finally the material was divided into two parts where the first half of the debate articles with subsequent discussions make up the training set and the rest constitutes a held-out test set. In the manual development of the salient dependency paths, only the training set was used. After the division, the training and test set consisted of a total of 382 and 308 sentences, respectively. Table 4.1 summarizes the annotation found in the corpus.

4.2 Corpus postprocessing

The sentiment corpus was PoS-tagged and parsed using the Bohnet&Nivre-parser (Bohnet and Nivre, 2012). This parser is a transition-based dependency parser with joint tagger that implements global learning and a beam search for non-projective labeled dependency parsing. This latter parser has recently outperformed pipeline systems (such as the Malt and MST parsers) both in terms of tagging and parsing accuracy for typologically diverse languages such as Chinese, English, and German. It has been reported to obtain a labeled accuracy of 87.7 for Norwegian (Solberg et al., 2014). The parser is trained on the Norwegian Dependency Treebank (NDT). The NDT is a treebank created at the National Library of Norway in the period 2011-2013, manually annotated with part-of-speech tags, morphological features, syntactic functions and dependency graphs (Solberg et al., 2014; Solberg, 2013). It consists of approximately 600 000 tokens, equally distributed

between Norwegian Bokmål and Nynorsk, the two Norwegian written standards. Only the Bokmål subcorpus has been used here. Detailed annotation guidelines in English will be made available in April 2014 (Kinn et al., 2014).

4.3 Sentiment lexicon and sentiment shifters

Unfortunately, no sentiment lexicon existed for the Norwegian language and therefore we developed our own by manually translating the AFINN list (Nielsen, 2011). We also manually added 1590 words relevant to political discussions like 'deport', 'expel', 'extremist' and 'terrorist', ending up with a list of 4067 Norwegian sentiment words. Each word were given a score from -5 to 5 ranging from words with extremely negative sentiment (e.g. 'behead') to highly positive sentiment words (e.g. 'breathtaking').

Several Norwegian sentiment shifters were considered but only the basic shifter 'not' improved the sentiment classification and therefore only this word was used in the method.

5 Experiments

In this study we compare four different methods based on the general algorithm in (1).

- We use the **imp**-function presented in (2). We denote this method WD (word distance).
- For this method and the two below we use the **imp**-function in (3). Further we set $\alpha = 0$ which means that we do not use the salient dependency paths. We denote this method A0 ($\alpha = 0$).
- We set $\alpha = 1$ and for all dependency paths we set $W_{\mathbf{dep}} = 2/3$. We denote this method CW (constant weights).
- We set $\alpha = 1$ and for $W_{\mathbf{dep}}$ we use the weights presented in Table 1. We denote this method OD (optimal use of dependency paths)

For each method we used the training set to manually tune the parameters d_p, d_n, t_p and t_n of the method. The parameters were tuned to optimize the number of correct classifications.

5.1 Statistical analysis of classification performance

We compare the classification performance of a set of M different methods, denoted as $\Pi_1, \Pi_2, \ldots, \Pi_M$, using random effect logistic regression. Let the stochastic variable $Y_{tm} \in \{0, 1\}$ represents whether method Π_m, $m \in \{1, 2, \ldots, M\}$ classified the correct sentiment to sentence number $t \in \{1, 2, \ldots, T\}$, where T is the number of sentences in the test set. We let Y_{tm} be the dependent variable of the regression model. The different methods $\Pi_1, \Pi_2, \ldots, \Pi_M$ is included as a categorical independent variable in the regression model. We also assume that classification performance of the different methods depends on the sentence to be classified, thus the sentence number is included as a random effect. Fitting the model to the observed classification performance of the different methods we are able to see if the probability of classifying correctly significantly vary between the methods.

The statistical analysis is performed using the statistical program R (R Core Team, 2013) and the R package `lme4` (Bates et al., 2013).

	d_p	d_n	t_p	t_n	**Accuracy**	**p-val**
WD	2	0	0.7	0.0	47%	
A0	2	0	2.0	0.3	52%	0.023
CW	2	0	2.0	0.3	52%	0.024
OD	2	0	2.0	0.3	53%	0.016

Table 3: The second to the fifth column show the optimal values of the parameters of the model tuned using the training set. The sixth column show the number of correct classifications and the last column shows p-values testing whether the method performs better than WD.

5.2 Results

Table 3 shows the optimal parameter values of d_p, d_n, t_p and t_n tuned using the training set, and classification performance for the different methods on the test set using the parameter values tuned from the training set. The p-values are computed using the regression model presented in Section 5.1. We see that $d_n = 0$, meaning that the sentiment shifter 'not' only has a positive effect on the classification performance when it is in front of the sentiment word. We see that using dependency distances (method A0) the classification results are significantly improved compared to using word distances in the sentence (method WD) (p-value = 0.023). Also classification based on

salient dependency paths (method OD) performs significantly better than WD. We also see that OD performs better than A0 (162 correct compared to 161), but this improvement is not statistically significant.

6 Closing remarks

Classifying sentiment in political discussions is hard because of the frequent use of irony, sarcasm and citations. In this paper we have compared the use of word distance between target word and sentiment word against metrics incorporating syntactic information. Our results show that using dependency tree distances or salient dependency paths, improves the classification performance compared to using word distance.

Manually selecting salient dependency paths for the aim of sentiment analysis is a hard task. A natural further step of our analysis is to expand the training and test material and use machine learning to see if there exists dependency paths that improve results compared to using dependency distance.

References

Jamie Bartlett, Jonathan Birdwell, and Mark Littler. 2013. The rise of populism in Europe can be traced through online behaviour... Demos, `http://www.demos.co.uk/files/Demos_OSIPOP_Book-web_03.pdf?1320601634`. [Online; accessed 21-January-2014].

Douglas Bates, Martin Maechler, and Ben Bolker, 2013. *lme4: Linear mixed-effects models using S4 classes*. R package version 0.999999-2.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465. Association for Computational Linguistics.

Bi Chen, Leilei Zhu, Daniel Kifer, and Dongwon Lee. 2010. What Is an Opinion About? Exploring Political Standpoints Using Opinion Scoring Model. In *AAAI*.

E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A Holistic Lexicon-based Approach to Opinion Mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 231–240, New York, NY, USA. ACM.

Matthew Goodwin, Vidhya Ramalingam, and Rachel Briggs. 2013. The New Radical Right: Violent and Non-Violent Movements in Europe. Institute for Strategic Dialogue, `http://www.strategicdialogue.org/ISD%20Far%20Right%20Feb2012.pdf`. [Online; accessed 21-January-2014].

Hugo Hammer, Alfred Bratterud, and Siri Fagernes. 2013. Crawling Javascript websites using WebKit with application to analysis of hate speech in online discussions. In *Norwegian informatics conference*.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Joan B. Hooper and Sandra A. Thompson. 1973. On the Applicability of Root Transformations. *Linguistic Inquiry*, 4(4):465–497.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Sasha Issenberg. 2012. How President Obamas campaign used big data to rally individual voters. `http://www.technologyreview.com/featuredstory/509026/how-obamas-team-used-big-data-to-rally-voters/` [Online; accessed 21-March-2014].

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large-scale SVM Learning Practical. In *Advances in Kernel Methods*.

Kari Kinn, Pl Kristian Eriksen, and Per Erik Solberg. 2014. NDT Guidelines for Morphological and Syntactic Annotation. Technical report, National Library of Norway.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.

R Core Team, 2013. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

John Shawe-Taylor and Nello Cristianini. 2000. *Support Vector Machines*. Cambridge University Press.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *Proceedings of LREC 2014*. Accepted.

Per Erik Solberg. 2013. Building Gold-Standard Treebanks for Norwegian. In *Proceedings of NODALIDA 2013*, Linkping Electronic Conference Proceedings no. 85, pages 459–464, Linkping, Sweden. LiU Electronic Press.

Andranik Tumasjan, Timm O Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the fourth international aaai conference on weblogs and social media*, pages 178–185.

Theresa Wilson, Janyce Wiebe, and Paul Hoffman. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399 – 433.