International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 10 | Oct -2016 www.irjet.net p-ISSN: 2395-0072

Sentiment Analysis of Political News articles and the effect of negation scope

Aruna Gunda, Varsha Teratipally

Department of Computer Science and Engineering, University College of Engineering, Osmania University, Hyderabad, India

---

Abstract - *With the explosion of Web and increased activity in blogging, tagging and commenting, there has been an eruption of interest in people to mine these vast resources for opinions. Sentiment Analysis or Opinion Mining is the computational treatment of opinions, sentiments and subjectivity of text. In this paper, we present our work on analyzing public sentiment towards two major political parties namely Congress and BJP using online news articles(formal text) and user comments on these articles(informal text). While informal texts express the opinion directly, formal texts express the opinion in a subtle way. In order to extract the sentiment from both informal and formal texts in an effective way, we have used standard lexical resource- SentiWordNet for classifying the text as positive or negative or neutral. However, without effective calculation of negation and its scope, sentiment calculation will not be accurate. Handling negation and determining the scope of negation is a challenging task in sentiment analysis but has still received little attention. We present an effective method to calculate negation scope and draw a comparison with widely used negation scope determination algorithms.*

Key Words: **Sentiment Analysis, politics, news articles, negation, SentiWordNet**

1.INTRODUCTION

"What other people think" has always served as an important piece of information for most of us during decision making process. Before the internet, polls and surveys provided that information. With the advent of the internet, and especially blogging where people express their opinions freely, there has been an eruption of activity in sentiment analysis and opinion mining. Sentiment analysis attempts to identify the opinion in a text span. Analysing publicly available data to infer public opinion is faster and less expensive than traditional polls and can actually provide a more real-time view of the current political climate. The 2014 general elections in India witnessed extensive use of social media for campaigning and expressing thoughts. Informal political discourse has become a part of the micro-blogging community. This opens up exciting avenues for sentiment analysis in political domains with studies revealing similar results to opinion polls. This is in the case of news blogs. Many newspapers give an impression of objectivity so that journalists will often refrain from using clearly positive or negative vocabulary. They may resort to other means to express their opinion, such as embedding statements in a more complex discourse or argument structure, they may omit some facts and highlight others, they may quote other persons who say what they feel, etc. This is the main difference between customer reviews and political news articles- whereas customer reviews explicitly state the opinion, news articles express the opinion in a subtle and indirect way. Hence, sentiment analysis of news articles is comparatively a tough job. Apart from these challenges, political articles use a different jargon and thus machine learning based sentiment analysis does not work very well. For this reason, we have opted for the standard lexical resource- SentiWordNet and some domain specific customized lexicons. Though SentiWordNet helps in classifying the text, it does not handle negation. Handling negation is as important as classifying the text as positive or negative because a sentence with all positive adjectives can express a negative opinion using a single negation word. Equally important is determining the scope of negation which is a challenging task in sentiment analysis. Till date, very less work has been done on news articles and negation scope determination. Our aim is to find an effective way to extract sentiment from news articles and news blogs i.e., both formal and informal texts by considering negation and its scope as well, compare the results obtained to identify the winning party in 2014 general elections and demonstrate the effect of our negation scope determination algorithm.

2. LITERATURE REVIEW

2.1 What is Sentiment Analysis?

Sentiment Analysis is a Natural Language Processing and Information Extraction task that aims to obtain writer's feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of a document. In recent years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind Sentiment Analysis today. The Web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task. The analysis of sentiments may be document based where the sentiment in the entire document is summarized

as positive, negative or objective. It can be sentence based where individual sentences, bearing sentiments, in the text are classified. Sentiment Analysis can be phrase based where the phrases in a sentence are classified according to polarity.

Sentiment Analysis identifies the phrases in a text that bears some sentiment. The author may speak about some objective facts or subjective opinions. It is necessary to distinguish between the two. Sentiment Analysis finds the subject towards whom the sentiment is directed. A text may contain many entities but it is necessary to find the entity towards which the sentiment is directed. It identifies the polarity and degree of the sentiment. Sentiments are classified as objective (facts), positive (denotes a state of happiness, bliss or satisfaction on part of the writer) or negative (denotes a state of sorrow, dejection or disappointment on part of the writer). The sentiments can further be given a score based on their degree of positivity, negativity or objectivity.

2. 2. Challenges in Political Sentiment Analysis

Sentiment Analysis approaches aim to extract positive and negative sentiment bearing words from a text and classify the text as positive, negative or else objective if it cannot find any sentiment bearing words. In this respect, it can be thought of as a text categorization task. In text classification there are many classes corresponding to different topics whereas in Sentiment Analysis we have only 3 broad classes. Thus it seems Sentiment Analysis is easier than text classification which is not quite the case. The general challenges are summarized below.

Domain Dependency: There are many words whose polarity changes from domain to domain. For example, sentiment conveyed in 'The story is unpredictable' is positive whereas the sentiment conveyed in 'He is unpredictable' is negative

Thwarted Expectations: Sometimes, the author deliberately sets up context only to refute it at the end. For example, the election campaign seemed great with actors, businessmen and social workers supporting the party. However, it can't hold up. In spite of the presence of words that are positive in orientation the overall sentiment is negative because of the crucial last sentence, whereas in traditional text classification this would have been classified as positive as term frequency is more important there than term presence

Negation: Handling negation is a challenging task in Sentiment Analysis. Negation can be expressed in subtle ways even without the explicit use of any negative word. A method often followed in handling negation explicitly in sentences like "I do not like the movie", is to reverse the polarity of all the words appearing after the negation operator (like not). But this does not work for "I do not like the acting but I like the direction". So we need to consider the scope of negation as well, which extends only till but here. So we can change polarity of all words appearing after a negation word till another negation word appears. But still there can be problems. For example, in the sentence "Not only did I like the acting, but also the direction", the polarity is not reversed after "not" due to the presence of "only". So this type of combinations of "not" with other words like "only" should also be handled while designing the negation scope determination algorithm.

2. 3. Feature Selection in Sentiment Analysis

Feature engineering is an extremely basic and essential task for Sentiment Analysis. Converting a piece of text to a feature vector is the basic step in any data driven approach to Sentiment Analysis. Some commonly used features used in Sentiment Analysis are listed below.

Term Presence vs. Term Frequency: Term frequency has always been considered essential in traditional Information Retrieval and Text Classification tasks. But Pang-Lee et al. (2002) found that term presence is more important to Sentiment analysis than term frequency. That is, binary-valued feature vectors in which the entries merely indicate whether a term occurs (value 1) or not (value 0). This is not counter-intuitive as the presence of even a single string sentiment bearing words can reverse the polarity of the entire sentence. It has also been seen that the occurrence of rare words contain more information than frequently occurring words, a phenomenon called Hapax Legomena.

Term position: Words appearing in certain positions in the text carry more sentiment or weightage than words appearing elsewhere. This is similar to IR where words appearing in topic Titles, Subtitles or Abstracts etc are given more weightage than those appearing in the body. Generally words appearing in the first few sentences and last few sentences in a text are given more weightage than those appearing elsewhere.

Parts of Speech: Parts of Speech information is most commonly exploited in all Natural Language Processing tasks. One of the most important reasons is that they provide a crude form of word sense disambiguation. In our experiment, we have given at most importance to Parts of speech in calculating the sentiment.

3. IMPLEMENTATION

This section introduces our framework for sentiment analysis and explains how it is handles negation identification, determines the scope of negation and calculation of sentiment at sentence level. The framework presented in Fig.1. consists of detection, extraction and classification components interacting at various levels. The framework uses SentiWordNet for sentiment classification. Its main components are briefly described in Sections 3.1 through 3.4

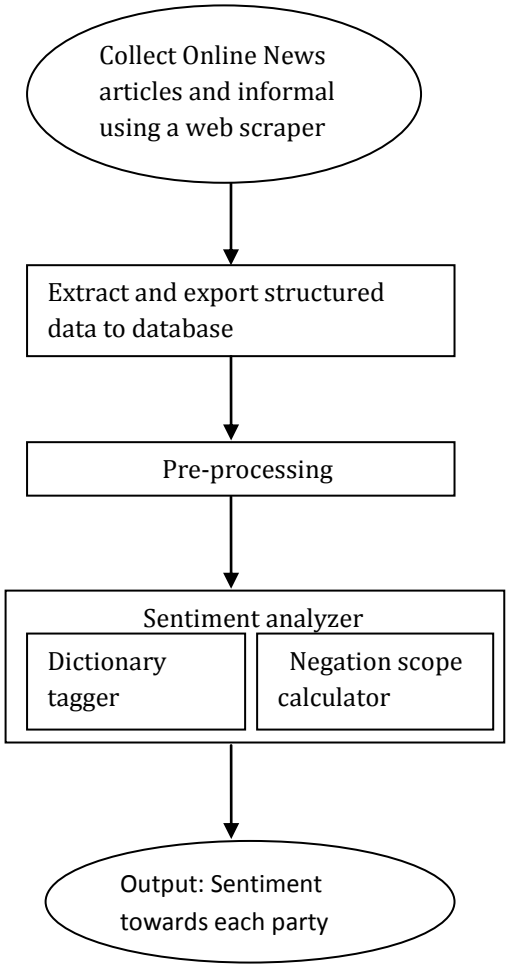

Fig -1: Sentiment analyser

3.1. Data Sets- Online news articles and their comments

We have extracted online news articles which are most debated and have many comments from three popular News websites: Times of India, The Hindu and NDTV. We focused on the two most likely winning parties in 2014 elections: Congress and BJP. The news articles are searched for either the abbreviations of the parties or the full names and their prominent leaders.

Web scraping is closely related to web indexing, which indexes information on the web using a boot or web crawler and is a universal technique adopted by most search engines. In contrast, web scraping focuses more on the transformation of unstructured data on the web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet. Web scraping is also related to web automation, which simulates human browsing using computer software.

We use a Web-scraping tool "easy web extract" to download articles from news websites. The unstructured data from these websites in HTML format is transformed into structured data by traversing the DOM structure. The resulting structured data is then exported to a database (MS Access in our experiment). We have collected a huge corpus of around 1000 articles and comments using this technique.

3.2. Pre-processing

In the pre-processing stage, the data is cleaned up to hold only what is essential for the analysis. Various steps like tokenization, pos tagging, stop word removal and lemmatization are done here using the python library NLTK.

Tokenization: Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing steps. Text is represented in Python using lists, such as: ['Slamming', 'Narendra', 'Modi', 'demand', 'debate', 'Article', '370', 'PDP', 'Monday', 'said', 'that', 'BJP', 'prime', 'ministerial', 'candidate', 'lacks', 'constitutional', 'knowledge', 'remarks', 'create', 'fissures', 'Jammu', 'Kashmir']

POS tagging: Part-of-speech tagging, also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph.
Example: [('Sheila',NN),('slams',VB),('Modi',NN),('over',IN), ('water',NN),('scarcity',NN)]

Stop word removal: Stop words are words which are filtered out prior to, or after, processing of natural language data (text). There is not one definite list of stop words which all tools use and such a filter is not always used. Some tools specifically avoid removing them to support phrase search. In our experiment, we have removed stop words after POS tagging to support phrase search.

3.3. Sentiment Analyzer

After pre-processing, the resulting list of tokens is passed to Sentiment Analyzer which has two components- Dictionary tagger and Negation scope calculator.

Dictionary tagger tags each token of every sentence with tags like positive, negative, negation. SentiWordNet values are taken here and according to the score given by SentiWordNet, we tag either positive or negative or neutral. Also, separate lexicons are maintained for domain specific words and negation words. Accordingly, tokens are tagged in the following manner: [[('minister', ['neutral', 0.0, 'NN']), ('of', ['neutral', 0.0, 'IN']), ('state', ['neutral', 0.0, NN']), ('for', ['neutral', 0.0, 'IN']), ('water', ['neutral', 0.0, 'NN']), ('resources', ['neutral', 0.0, 'NNS']), ('and', ['neutral',

0.0,'CC']), ('bjp', ['neutral', 0.0, 'VB']), ('leader', ['neutral', 0.0, 'NN']), ('nanu', ['neutral', 0.0, 'NN']), ('vanani', ['neutral', 0.0, 'NN']), ('has', ['positive', 1.0, 'VBZ']), ('attacked', ['neutral', 0.0, 'VBN']), ('for', ['neutral', 0.0, 'IN']), ('giving', ['neutral', 0.0, 'NN']), ('a', ['neutral', 0.0, 'DT']), ('ticket', ['neutral', 0.0, 'NN']), ('to', ['neutral', 0.0, 'TO']), ('social',['neutral', 0.0, 'JJ']), ('activist', ['neutral', 0.0, 'NN']), ('medha', ['neutral', 0.0, 'NN']), ('patkar', ['neutral', 0.0, 'NN']), ('to', ['neutral', 0.0, 'TO']), ('contest', ['neutral', 0.0, 'VB']), ('forthcoming', ['neutral', 0.0, 'JJ']), ('loksabha', ['neutral', 0.0, 'NN']), ('polls', ['neutral', 0.0, 'NNS'])]].

However, these SentiWordNet sentiment scores may not be sufficient, as we aim to process negation as well. Our negation processing approach relies on occurrences of (English) negation keywords like: "no", "not", "-n't", "never", "less", "without", "barely", "hardly", "rarely", "no longer", "no more", "no way", "nowhere", "by no means", "at no time", and "not (...) anymore". Each negation keyword is assumed to have a scope of influence. This scope can be determined in many ways, as further detailed in Section 3.4. Having determined the scope of negation keywords using any of the considered methods, the sentiment scores associated with the words in the negation keywords' scope can be inverted. To this end, we introduce per-word sentiment modifiers, which are initialized at a value of 1, indicating that the sentiment score retrieved from the sentiment lexicon is considered to be the true sentiment score associated with that word in the considered context. In case a word is negated, the sentiment modifier may be multiplied with an inversion factor. Initially, we assume this factor to be equal to −1. However, as we hypothesize that negated sentiment may not necessarily be as strong as its non-negated counterpart (compare, e.g., "not bad" and "good"), our framework also supports Modified Inversion Strength (MIS), where the inversion factor is -1. Finally, when all word scores have been determined while accounting for negation, sentences can be classified as either positive or negative. For this, we have implemented a sentence scoring function. If the sum of word-level sentiment scores in a sentence produces a number smaller than 0, the sentence is classified as negative, if the score is 0 then the sentence is classified as neutral, else the sentence is classified as a positive sentence.

The input to Sentiment Analyser is a paragraph of sentences and the output is its sentiment classification. For each identified negation keyword, the sentiment modifier of the words within the scope of this keyword is multiplied with the sentiment inversion factor. This sentiment modifier initially equals 1, indicating that no inversion is applied. When all negation keywords have been processed, the sentence is scored by summing the (modified) sentiment scores of all words in the sentence. The resulting sentiment score is then used to classify the sentence.

3.4. Negation Scope Determination

Negation identification and scope determination is very important in calculating the polarity of a sentence.
The words in a sentence, their meanings, alternative words, polarity of each word and intensity associated with each word are basic elements used by sentiment analyzer for sentiment identification. The polarity of sentence is usually based on the meaning of words. However, negation words change the meaning of the words and polarity of the sentence. Moreover, grammatical structure of the sentence determines the scope of this negation word. In order to accommodate these rules, our dictionary tagged sentences are forwarded to negation scope calculator where a parse tree is generated based on the grammatical structure of the sentence using Stanford NLP parser. The scope of Negation will be identified using this parse tree, which indicates how negation word is interacting with other words in the sentence.

Existing algorithms like Fixed Window Length (Scope is a fixed window length of words following a negation word), Rest of the Sentence (Scope is the rest of the sentence following the negation word) do not address the grammatical structure of the sentence and hence do not yield accurate results. Few experiments have been done considering the grammatical structure and dependency tree. However, these algorithms do not perform very well as they consider only siblings of the negation word or children of the negation word in the dependency tree. In this experiment, we have implemented an improvised algorithm based on dependency tree that has shown better results compared to the existing algorithms.

In our proposed method, 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).

For every sentence, if there are any negation words, then a parse tree is generated and tree is traversed to determine the scope of the negation. To get the parse tree of the sentences, we use Stanford parser. Stanford parser output is given to the NLTK parse function to get the Tree object of NLTK, so that traversing through the parse tree is made possible. After getting the Parse tree object, we find the polarity with the following algorithm:

For each node, check if any of the grandchildren of the current sub-tree is a negation word. If so, reverse the polarities of all the leaves of this sub-tree and call the function recursively. If not, calculate the polarity. In this process, sentences are tagged with a reverse flag- True or False. For calculating the polarity of the sentences, we check if the reverse flag is true or false. Depending on the reverse flag, polarities are added or reversed before final calculation.

Consider the below example:

"People favour anything but Congress this elections says Medha Patkar"

Here but is acting as a Negation word meaning except and thus reversing the sentiment expressed by the word 'favour'. The parse tree for this sentence is as shown in Fig- 2 below.

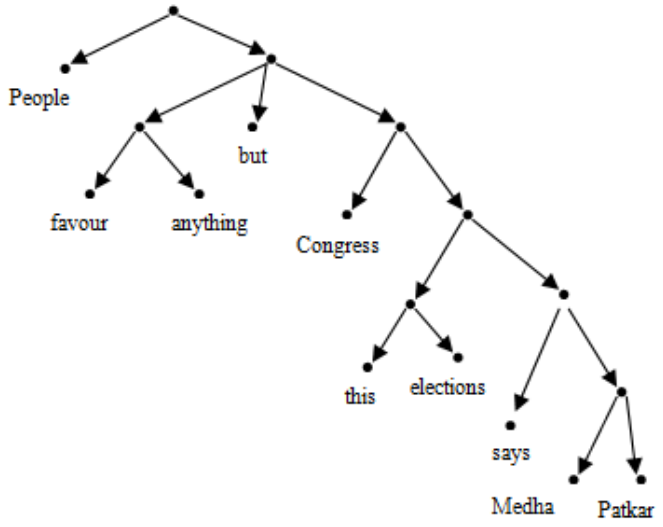

Fig -2: Parse tree for example sentence

A dependency algorithm that considers only siblings or children in the parse tree would not work well in sentences like this which are very common in political news articles or comments. As our algorithm applies negation to all the words for which the parent of negation word is an 'ancestor', the above sentence's polarity will be accurately calculated as negative by reversing the polarity of favour(+).

4. EFFECT OF OUR NEGATION SCOPE ALGORITHM

As stated in the previous sections, we have applied straightforward negation scope algorithms like Fixed Window Length (FWL), Rest of the Sentence (ROS) and existing dependency algorithm (DAE) as well on our data to draw a comparison of the effectiveness of these algorithms with that of our proposed algorithm. Below is the table of resultant accuracy in percentage for each of these algorithms.

Text source	***DAP***	***DAE***	***ROS***	***FWL***
Hindu formal BJP	*82.75*	*81.60*	*73.27*	*81.03*
Hindu formal Cong.	*80.76*	*79.56*	*77.69*	*77.69*
Hindu informal BJP	*73.52*	*70.80*	*70.58*	*67.64*
Hindu informal Cong	*87.5*	*84.33*	*79.17*	*83.33*
NDTV formal BJP	*87.14*	*86.66*	*81.43*	*80.71*
NDTV formal Cong.	*90.85*	*87.75*	*80.48*	*82.31*
NDTV informal BJP	*83.33*	*79.80*	*72.22*	*72.22*
NDTV informal Cong	*84.88*	*80.00*	*81.39*	*75.58*
TOI formal BJP	*80.48*	*78.17*	*75.61*	*78.05*
TOI formal Cong.	*82.35*	*79.58*	*70.58*	*70.58*
TOI informal BJP	*80.76*	*76.25*	*78.84*	*75*
TOI informal Cong.	*76.74*	*73.33*	*72.09*	*72.09*
Average(Accuracy)	***82.58***	*79.82*	*75.89*	*76.35*

Table -1: Comparison of various Negation scope algorithms
DAP- Our proposed Dependency Analysis algorithm

5. CONCLUSION

In order to assess the effects of various methods of determining negation scope as well as our proposed method for accounting for negation strength, we have implemented the framework presented in Section 3. The performance of our sentiment analyzer with an improvised dependency analysis algorithm for determining negation scope was evaluated on a collection of around 1,000 positive and negative news articles and comments extracted from news web sites. We calculated the accuracies of these algorithms by comparing with the manual annotation and our results are reported in Table -1 above in the section 4. A comparison between the widely used existing algorithms and our proposed algorithm has clearly shown our algorithm performs better for both formal and informal texts. Overall, the worst performing approach is ROS. FWL considering words around a keyword to be the scope do not appear to perform better than the existing dependency algorithm or our proposed dependency algorithm in terms of overall accuracy.

Finally, our improvised Dependency Analysis algorithm performs better than the existing dependency algorithms as well, especially in case of informal texts. The framework is not designed by keeping any specific lexical resource in mind; therefore, by improving the precision of resources the results can easily be improved. Our experiment results are clearly favorable to BJP and that is proved true in 2014 elections.

ACKNOWLEDGEMENT

We express our deepest sense of gratitude to our guide Dr. S. SAMEEN FATIMA, Professor, University College of Engineering, Osmania University, Hyderabad, who has driven our passion to explore Natural Language Processing. We have received great guidance, encouragement and support from her and have learned a lot because of her willingness to share knowledge and experience.

REFERENCES

[1] John. C. Platt. "Sequential Minimal Optmization: A fast algorithm for training support vector machines", Microsoft research technical report, Apr 21, 1998. [Online]Available: http://research.microsoft.com/pubs/69644/tr-98-14.pdf

[2] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," presented at the Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, 2005.

[3] M. Hu and B. Liu, "Mining and summarizing customer reviews," presented at the proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA, 2004.

[4] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques," presented at the proceedings of the Third IEEE International Conference on Data Mining, 2003.

[5] N. Jakob and I. Gurevych, "Extracting opinion targets in a single- and cross-domain setting with conditional random fields," presented at the Proceedings of the 2010 Conference on empirical Methods in Natural Language Processing, Cambridge, Massachusetts, 2010.

[6] J. S. Kessler and N. Nicolov, "Targeting sentiment expressions through supervised ranking of linguistic configurations," in Proceedings of the Third International AAAI Conference on Weblogs and Social Media,, San Jose, California, USA, 2009, pp. 90-97.

[7] W. Jin, H. H. Ho, and R. K. Srihari, "OpinionMiner: a novel machine learning system for web opinion mining and extraction," presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009.

[8] V. Stoyanov and C. Cardie, "Topic identification for fine-grained opinion analysis," presented at the Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, Manchester, United Kingdom, 2008.

[9] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," Comput. Linguist., vol. 37, pp. 9-27, 2011.