Real-time Sentiment Analysis of Hindi Tweets

Aanusha Ghosh
M.A. in Linguistics
The EFL University
Tarnaka
Hyderabad - 500007
aanusha.ghosh@gmail.com

Indranil Dutta
Department of Computational Linguistics
The EFLUniversity
Tarnaka
Hyderabad - 500007
indranil@efluniversity.ac.in

Abstract

This paper explores real-time sentiment analysis (SA) of Twitter posts in Hindi, adopting a resource-based approach and classifying sentiment on a three-way scale of negative, positive and neutral. Furthermore, the efficiency of different approaches such as part-of-speech (POS) tagging and stopword removal are compared, and ways of improving the Hindi Senti-Wordnet (Aditya Joshi and Bhattacharya, 2010) are proposed.

Little work in the area of SA and opinion mining has been done on Indian Languages. With the growth of online content in these languages, especially in social media, sentiment analysis is a burgeoning field for the application of natural language processing methods. While there has been some work done regarding SA in Hindi (Mittal et al., 2013), it has been conducted on a static database. The goal of this paper is to conduct SA on a *dynamic* corpus of tweets.

1 Introduction

Much of the existing work on sentiment analysis and opinion mining has been carried out in resource-rich languages like English. While data in Indian languages still remain relatively sparse, web content in languages like Hindi have rapidly increased over the past few years due to the introduction of UTF-8 unicode standards and also greater penetration of the internet in the subcontinent. Also, with the rise in stature of social media platforms like Twitter and Facebook, the volume of user-generated content has swelled greatly, creating a new field for the application of opinion mining techniques . While SA of tweets has been carried out in English (Pak and Paroubek, 2010), there are no such analyses of similar data in Hindi. In this paper, we perform real-time SA on live streams of Hindi tweets.

Microblogging sites like Twitter are a popular platform for the sharing of opinions of millions of users on a diverse range of topics. Therefore, such websites are rich sources of data for sentiment analysis and opinion mining. Twitter users frequently use hashtags (keywords with a hash in front of them that act as metadata tags), which can then be used to group similar posts containing the same tag together. Used by a sufficiently large group of people, hashtags can become trends that attract more and more individuals to participate in a particular discussion.

The primary goal of this paper is to analyze the sentiment of individual tweets in Hindi and to capture the sentiment associated with a given hashtag. Since Twitter does not support hashtags in Devanagari, we gather tweets for a specific hashtag (which uses the Latin script) and calculate average sentiment over all the tweets containing the tag. We focus on analysing the context in which hashtags appear in order to determine the sentiment they are associated with. Sentiment analysis performed this way over a body of tweets streamed in real-time gives a snapshot of the mood of Twitter users regarding a specific topic at a given time.

The rest of the paper is organised as follows: In Section 2, related works regarding SA in English in general and those on Twitter corpora, as well as existing SA works in Hindi are presented. Section 3 describes the Twitter corpora and textual resources that have been generated for Hindi and also the methods that are employed for conducting SA on the Hindi tweets. In Section 4 we present the results from the statistical analyses. In section 5 we discuss the effects of POS tagging, the limitations of SentiWordnet as well as the challenges faced in assigning polarities to tweets and outline the various areas which require further research in the future.

2 Related work

One of the earliest works in the field of sentiment analysis was that of Turney (2002) which classified automobile and movie reviews by using unsupervised learning techniques. In the same year, Pang et al. (2002) undertook the task of sentiment classification using supervised machine learning methods, drawing contrast between three approaches — Naïve Bayes, maximum entropy classification, and support vector machines. Baccianella et al. (2010) presented, for the first time, a lexical resource in English called SentiWordNet that was designed for aiding opinion mining and sentiment classification, and followed it up later with updated versions, the latest being SentiWordNet 3.0.

Opinion mining and sentiment analysis of Twitter in English has already been explored by many people, including Pak and Paroubek (2010), who created training datasets by filtering tweets by 'happy' and 'sad' emoticons, and training a Naïve Bayes classifier on different features such as the presence of n-grams and the POS-tag distribution information. Kouloumpis et al. (2011) used existing hashtags in the tweets to train datasets on possible polarity values.

Existing work on sentiment analysis in languages other than English primarily deals with different techniques of assigning sentiment values to text. One method is cross linguistic SA by Machine Translation—translating the original text into English, and using the corresponding synsets in the English WordNet to assign polarity values. This works fairly well for languages that have scarce resources and a dependable machine translator. A second way of doing it is to create a special WordNet built solely for the purpose of sentiment analysis, where each word is assigned probabilities of being positive, negative or neutral. This method is a good alternative for languages that already have their own WordNet in place. According to Balamurali et al. (2011), WordNet synset-based features perform better than word-based features for SA.

Aditya Joshi and Bhattacharya (2010) devised a three-step model for SA that incorporated in-language sentiment analysis, machine translation and the development of a lexical resource called Hindi SentiWordnet (H-SWN) which assigned words probabilities for each sentiment and was based on Baccianella et al. (2010)'s English SentiWordNet, from which it borrowed the polarity scores of corresponding English words. Mittal et al. (2013) later used the H-SWN for classifying a corpus of Hindi *reviews* which incorporated negation handling and discourse analysis, as well as a method for augmenting the original H-SWN.

Sentiment analysis in Hindi is still a fledgling field, and no work has been done on SA of Hindi social network data. Our work on the analysis of Hindi tweets presents the first attempt to do so. We perform SA on a dynamic corpus composed of tweets that are streamed in realtime.

3 Materials and methods

In this section we present the resources we require to perform SA. The section on preprocessing outlines the tasks that need to be completed before SA can be performed on the data. Preparing the POS tagger is discussed next, followed by the steps for conducting SA on the data.

We build a corpus of Hindi tweets with the help of the Twitter Application Programming Interface (API) [1], which enables the user to query the Twitter fire-hose for either a specific keyword or for tweets of a specific language and obtain a live stream of tweets. During preprocessing, we use a lightweight stemmer for Hindi implemented in Python by Luis Gomes [2], which makes use of the suffix stripping algorithm outlined in Ramanathan and Rao (2003). Removal of stopwords was carried out with the help of Goutham Tholpadi's list of stopwords in Hindi [3].

Sentiment scores are computed using Hindi SentiWordnet (H-SWN) (Arora et al., 2012) – which is a lexicon of adjectives, adverbs and nouns, each of which are assigned polarity scores. H-SWN is compiled with the help of the Hindi WordNet (IIT Bombay) and English-Hindi WordNet Linking (Karthikeyan, 2010). Each line in H-SWN contains the relevant POS-tag, Synset ID, positive and negative probabilities along with a list of words as shown below:

a 28106 0.25 0.375 अच्छा,बढ़िया

We also use a tagged Hindi corpus by IIT Kharagpur to train Python's Natural Language Toolkit (NLTK) unigram POS tagger.

[1] https://dev.twitter.com/

[2] http://research.variancia.com/hindi_stemmer

[3] https://sites.google.com/site/gtholpadi/histpwords.txt

3.1 Preprocessing

A JavaScript Object Notation (JSON) parser is used to render the tweets into an organised, human-readable format. Every tweet is composed of numerous fields that contain values. For example, the body of the tweet is generally found in the field labelled 'text'. Exceptions occur when a tweet is actually a re-tweet, in which case the 'text' field contains a truncated version of the tweet, and the original text is stored in another field labelled 'retweeted_status'. There are also fields which store information on the number of times a tweet has been retweeted, the user who tweeted it, the number of followers the user has, the language of the tweet, the geographic location of the user and so on. Tweets are streamed by querying the API and setting the language as Hindi. Only data from the 'text' and 'retweeted_status' fields are used in this paper.

Once the tweets have been streamed to a file, we clean the raw data. Regular expressions are used to remove URL links (e.g., http://example.com), Twitter handles (e.g., @username) and special characters (such as %, * or $) and leading or trailing whitespaces. The file is also checked for repetitions, which are identified and removed after the cleaning process is over. Additionally, we make a list of all the hashtags that appear in each tweet, as well as the number of times each individual hashtag appears in the entire corpus.

Stopwords, such as postpositions and function words, are removed from each tweet. H-SWN lists only the root forms of words, while most of the occurrences of words in the tweets use inflected forms. Therefore, each word in a given tweet is stemmed, and both the stemmed and unstemmed forms of the words are searched in H-SWN.

3.2 Training the Tagger

The training corpus for the POS tagger [4] consists of more than 4400 sentences, each tagged for parts of speech, gender and number. We retain the tags that reflect only the parts of speech. The unigram tagger works on the assumption that the POS tag of a word is independent of the context it appears in. We use NLTK's unigram tagger and train it on this dataset. The tagger shows an accuracy of 77.1% when trained on 90% of the data and tested on the remaining 10%.

Since H-SWN contains only the tags for adjectives, nouns and adverbs, we bin the tags of the training corpus in order to make it compatible with H-SWN. This is done by taking all subcategories of a given tag and equating them to the corresponding coarse POS tag in H-SWN (eg: 'AMN' (manner) and 'ALC' (location), subcategories of the tag 'A' for adverbs are both converted to 'r', which is the tag for adverbs in H-SWN).

3.3 Sentiment Analyis

For the sentiment analysis in this paper, we assume that the presence of a word with predetermined polarity probabilities affects the polarity of the tweet it appears in. In this approach, grammar or word order is ignored, and priority given to the frequency of these positive and negative words. We build a model for the ternary classification task of categorising sentiment as positive, negative and neutral. Figure 1 illustrates the pipeline for conducting SA.

Sentiment scores of each tweet are computed on the cleaned corpus with the help of H-SWN as follows:

Polarity score of each word = positive + negative scores of all entries of that word in H-SWN

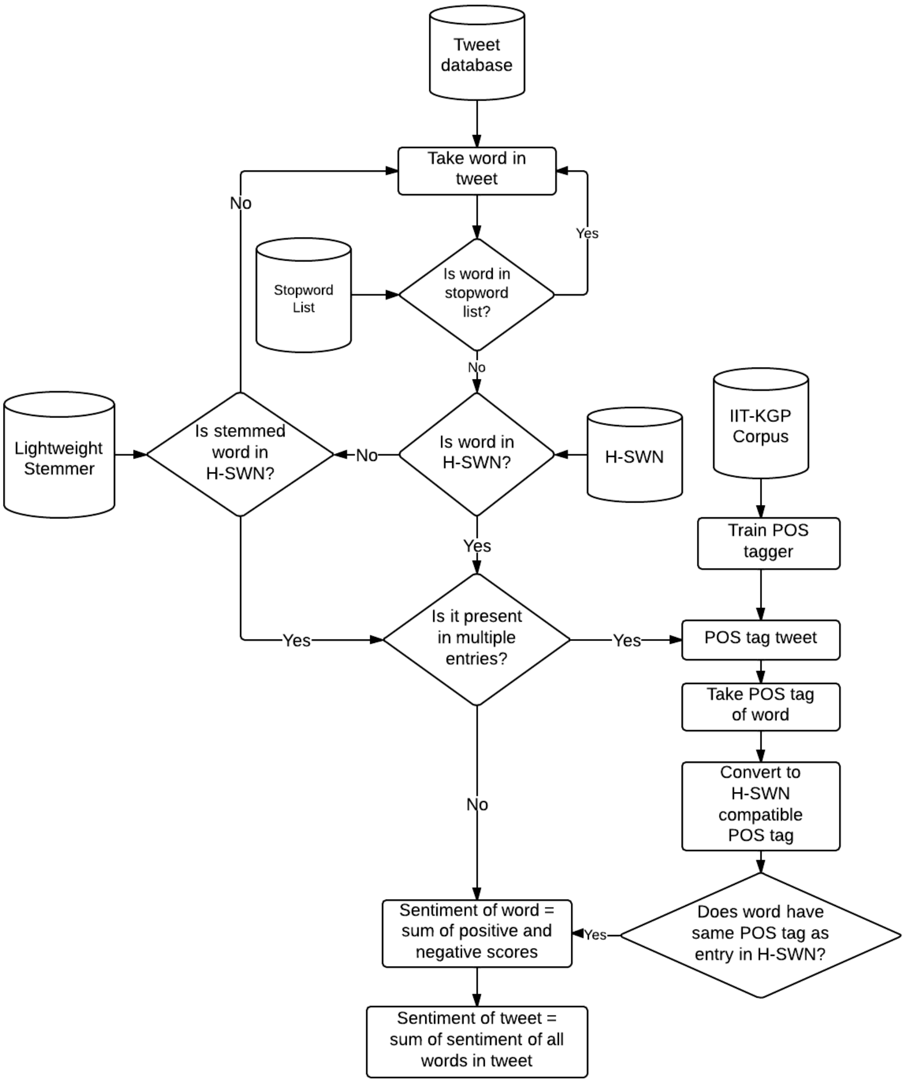
Sentiment score of tweet = sum of polarity scores of each word in that tweet

For words that have multiple senses, and hence multiple entries in H-SWN, we take the sum of the scores of all the possible senses. For example, if a word has four possible senses, three of which are positive and one of which is negative, the overall score taken will be equivalent to three positive and four negative words.

For the POS tagging approach, however, instead of taking into account all the entries of a single word, we only take those entries which have the same tag as the one in the tweet. If the tagger fails to tag a word, we fall back on the original method of taking all possible senses of the word in question.

Since Twitter hashtags are always in Latin characters, the hashtags are not assigned any polarity score. Once every tweet in the corpus has been assigned a sentiment score, we assign sentiment scores for each individual hashtag. To do so, we take all the tweets in which a given hashtag appears and take the average of all their sentiment scores in order to compute the related sentiment of that hashtag, as below:

[4] http://nltr.org/snltr-software/



Tweet database
Take word in tweet
Stopword List
Is word in stopword list?
Yes
No
Lightweight Stemmer
Is stemmed word in H-SWN?
No
Is word in H-SWN?
H-SWN
IIT-KGP Corpus
Train POS tagger
Yes
Is it present in multiple entries?
Yes
Yes
POS tag tweet
Take POS tag of word
Convert to H-SWN compatible POS tag
No
Sentiment of word = sum of positive and negative scores
Yes
Does word have same POS tag as entry in H-SWN?
Sentiment of tweet = sum of sentiment of all words in tweet

Figure 1: Pipeline for conducting SA

Related sentiment score of hashtag = (sum of all sentiment scores of tweets given hashtag) / number of tweets

3.4 Training a Näive Bayes classifier

Since there are a lot of false neutrals in our output, we classify these into positives and negatives with the help of a Näive Bayes classifier. A feature set of all the tweets already classified as negative or positive is compiled. This set consists of the 150 most common words that appear in the corpus, and a Boolean variable for each that marks its presence or absence in each tweet. The feature set also contains the polarity of the tweets. The classifier is trained on 80% of the data and tested on the remaining 20%, and has an average accuracy of 72%. The feature extractor function is used to build feature sets for the neutral tweets and the trained classifier classifies them as either positive or negative.

4 Results

Sentiment analysis is carried out on our database, which consists of 10 sets of tweets collected over half-hour intervals, each containing over 600 tweets. For each set, density plots are generated for two sets of sentiment scores, one obtained with POS tagging, and the other without. All the density plots indicate a marked increase in the volume of objective outcomes in the case of POS tagging. Paired t-tests of these sets of sentiment scores were conducted, keeping the level of sig-

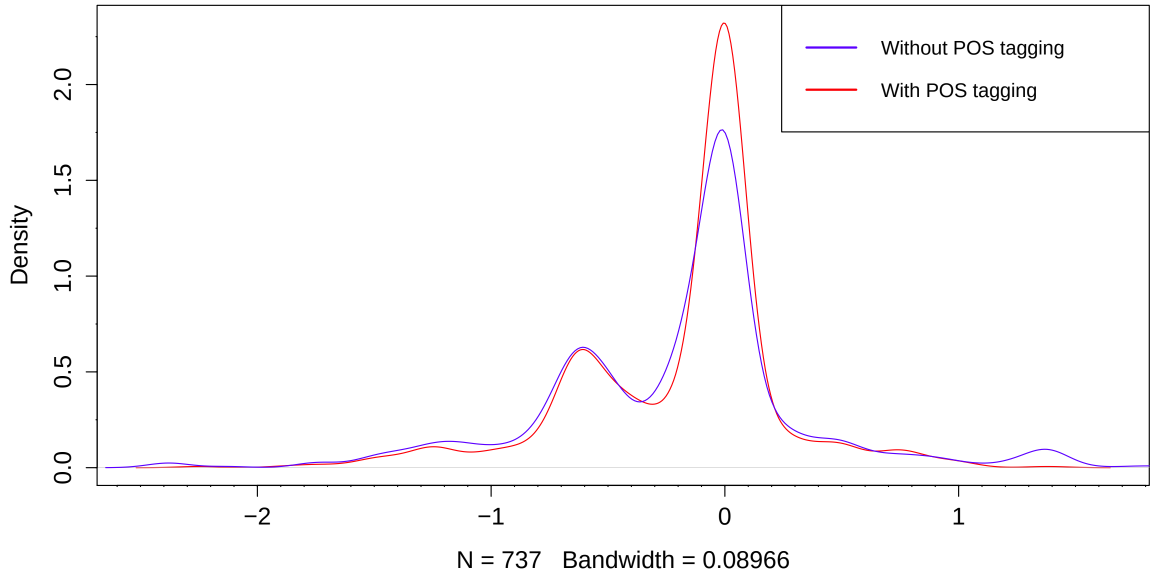


Figure 2: Density plots of sentiment scores with and without POS tagging

nificance at 95% ($\alpha = 0.05$). The results suggested that there is a significant difference between the outcomes of the POS tagging approach and the one without, since in 8 out of 10 cases the p-values were lower than α.

Table 1: Paired t-test Results

Set	Sample Size	t-values	p-values
1	737	0.612	0.5407
2	760	1.859	0.0635
3	630	3.645	0.0003
4	617	3.360	0.0008
5	656	3.115	0.0019
6	724	2.294	0.0220
7	622	3.470	0.0006
8	726	2.831	0.0048
9	731	2.230	0.0260
10	708	2.283	0.0227

Closer inspection of individual tweets and corresponding pairs of assigned sentiments using the two methods shows that many of the positive outcomes in the non-POS-tagged method became objective outcomes when POS tagging was enforced. A representative density plot is shown here (Figure 2). Note how the positive peak in the non-POS-tagged sentiment score vanished in the POS-tagged score and how it is marked with a corresponding rise in the number of neutral outcomes.

Thus, we conclude that POS tagging does not improve accuracy, and reduces the degree of polarisation of the sentiment scores.

Using a Näive Bayes classifier to classify tweets which have falsely been labelled as neutrals works well, as shown in the example below, which was correctly identified as negative by the classifier:

7 वर्षीय बच्ची के साथ स्कूल के भीतर बने स्टाफ क्वॉर्टर में रेप किया गया। #BangaloreRape

However, the accuracy of the classifier depends largely upon the accuracy of the training dataset, which in turn depends on the quality of H-SWN.

The variation of the sentiment attached to a hashtag over time can be mapped in order to get an idea of the trend of people's views on that topic. Figure 3 shows one such example. The hashtag #whybapujitargeted appeared in all of the 10 datasets, and the fluctuation of the sentiment values assigned to it over time has been plotted. The mapping of sentiment associated with a hashtag gives valuable insight into the varying moods of the populace, and the trending topics at a given time. Future work regarding hashtags include using regression analysis to establish relationships between different hashtags by tracking the changes in their associated sentiments.

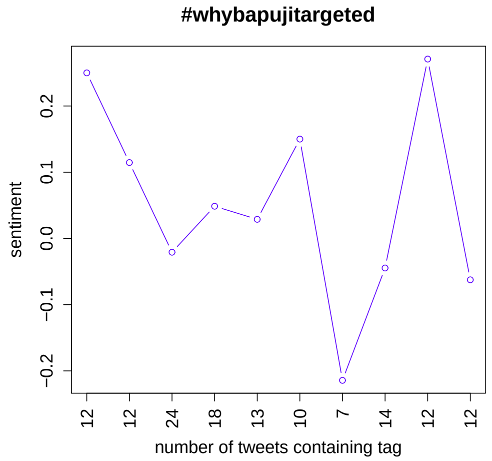


Figure 3: Variations of sentiments over time for a single hashtag

5 Conclusion and Discussion

Using the H-SWN to analyse the sentiments of tweets works moderately well with larger datasets, where the sentiments related to the hashtags take into account a greater sample of tweets, and errors in polarity assignment for individual tweets averages out in the long run. However, the chief drawback is its issues with accuracy. A lot of tweets that should have been either negative or positive get assigned neutral scores, which is due to three possible reasons:

- All the words in the tweet have equal probabilities of having negative and positive senses, that is, each word has an overall sentiment score of zero.
- The net value of the tweet cancels out to zero due to an equal number of positive and negative words.
- None of the words are assigned polarity scores because of the fact that they do not appear in H-SWN.

In some cases, a tweet might have a hashtag which is unrelated to the subject of the main tweet. In such cases, relating the sentiment score of the tweet to that of the hashtag is erroneous. For example, consider the following tweet:

कांग्रेस पार्टी के चरित्र में लोकतंत्र नहीं है,वह वशंवाद से चलते हैं,परिवादियों को शामिल करते हैं। #Modi

Here, although the hashtag is #Modi, the subject of the tweet is the Congress. Hence, either the sentiment of the tweet should not count towards the hashtag at all, or the hashtag should be assigned a polarity opposite of what the tweet is assigned. The program however, assigns#Modi the same sentiment score as that of the tweet.

Another aspect of the problem that the program does not address is the degree of polarisation of opinion. Opinions on topics are generally highly polarised on social media platforms. However, the program does not take polarisation into consideration when computing a sentiment value for a hashtag. As a result, highly polarised scores run a risk of being nullified by each other, resulting in a flattened neutral value.

A noticeable trend in many of the Hindi tweets was that they were bilingual, typed in a mix of Devanagari, English, and in some cases Romanized Hindi. In such cases, only the text in Devanagari was processed, leading to a substantial loss in information. The only way to deal with this issue is to perform multilingual sentiment analysis, and to convert the Romanized text into the Devanagari script, which in itself is a complex problem, given the fact that there is no fixed popular standard for Romanizing Hindi, and there are multiple equally popular variations of even very simple words (e.g., ‘main’, ‘mein’, and ‘me’ all refer to the same word in Hindi).

5.1 Hindi SentiWordnet

A close examination the H-SWN reveals that the number of objective words are significantly higher than the number of positive and negative words (6464 entries out of the 10349 entries in H-SWN are completely objective, that is, they have absolutely no negative or positive probabilities). One of the reasons for this might be the incomplete linking between English and Hindi Wordnet synsets, as well as an insufficient seedlist that was chosen to initially populate the H-SWN when it was created. Additionally, many stopwords like और, बाद, बहुत etc. have nonzero polarity scores. Stopword removal, while failing to reduce the number of neutral outcomes, significantly decreases the number of negative outcomes, as illustrated in Figure 4.

Although the Hindi SentiWordnet provides a starting point for conducting resource-based sentiment analysis, the accuracy of the outcome has definite scope for improvement. The following features could be implemented in order to address present issues in accuracy:

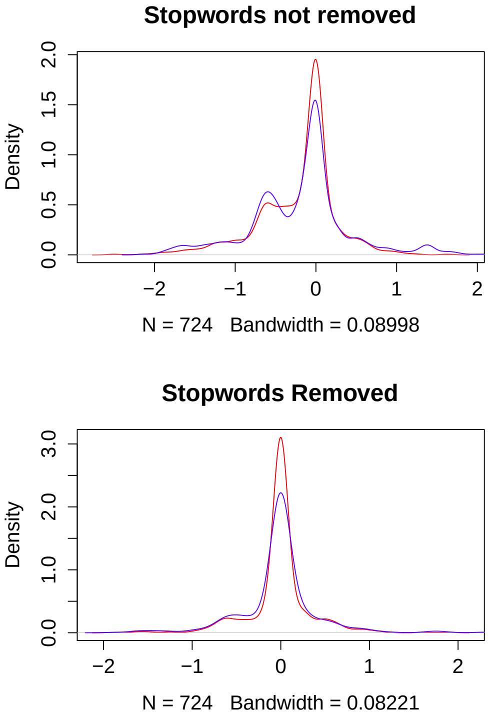


Figure 4: Effect of stopword removal

- Greater coverage - Using the improved H-SWN (Mittal et al., 2013), which adds 573 new words to the original H-SWN, results in more neutrals getting resolved into positives and negatives, as illustrated in Figure 5.The H-SWN currently has a size of 10349 entries. Complete linking of the H-SWN with the English Sentiwordnet 3.0, which has 206941 entries, would improve its scope and quality substantially.

- Inclusion of more polarised words - The majority of words in the H-SWN happen to be objective. Adding more words which have either positive or negative scores will reduce the bias towards neutrality that we observe in our study.

5.2 Negation Handling

Negation handling ensures that the presence of negations is taken into account while calculating the polarity of a tweet. However, negation handling in free word-order languages like Hindi is tricky. A possible solution to this issue has been addressed in Mittal et al. (2013), where the presence of negative words were used to reverse the polarities of words in the immediate vicinity (taking a window size of three of five words).

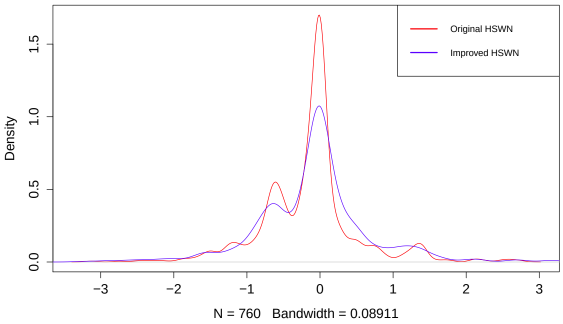


Figure 5: Comparison of original and improved H-SWN

5.3 Named Entity Recognition

Named entity recognition (NER) is a method of extracting names and named entities (geographical, geo-political, etc) from a body of text. NER is essential for determining the subject of a tweet. A named entity annotated corpora for Hindi is available from the Indian Language Technology Proliferation and Deployment Centre which could be used to tag named entities for every tweet and check for discrepancies between the text of the tweet and the hashtag.

References

Balamurali A.R Aditya Joshi and Pushpak Bhattacharya. 2010. A fall-back strategy for sentiment analysis in hindi: a case study. In *In International Conference On Natural Language Processing (ICON)*, Hyderabad, India.

Piyush Arora, Akshat Bakliwal, and Vasudeva Varma. 2012. Hindi subjective lexicon generation using wordnet graph traversal. In *Journal Proceedings of CICLing 2012*, New Delhi, India, March.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. 2011. Harnessing wordnet senses for supervised sentiment classification. In *Proceedings* *of the Conference on Empirical Methods in Natural Language Processing*, pages 1081–1091. Association for Computational Linguistics.

A Karthikeyan. 2010. A hindi english wordnet linkage.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press.

Namita Mittal, Basant Agarwal, Garvit Chouhan, Prateek Pareek, and Nitin Bania. 2013. Discourse based sentiment analysis for hindi reviews. In *PReMI*, pages 720–725.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ananthakrishnan Ramanathan and Durgesh D. Rao. 2003. A lightweight stemmer for hindi. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.