Transfer of Polarity Score for Sentiment Classification in Hindi

Vartika Rai **Sakshee Vijay** **Dipti Misra Sharma**
MT-NLP Lab,Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad -India
{vartika.rai,sakshee.vijay}@research.iiit.ac.in
dipti@iiit.ac.in

Abstract

Sentiment analysis in a resource scarce language is a tedious task. We propose a novel method for transfer learning from a target language to English. Our system doesn't rely on labeled data for the target language but instead links itself onto already existing and extensively labeled word-level lexical resource in English (ESWN[1]) and a semantic parser. Our proposed system transparently needs no target language sentiment corpus, and exploits complex linguistic structure of the target language for sentiment prediction. This cross lingual approach gives net accuracy as 83.6%, an improvement of 5.4% over the baseline system.

1 Introduction

In late 2000's, Hindi had least share in terms of online presence. English and European Languages had major share on web and social platforms.But after 2010, its presence has witnessed a sharp growth in web texts, social media platforms, online personal assertive tools, etc. There are over 200 million Hindi speakers in north India alone. With more and more people indulging themselves into using Hindi as their communication language, this huge amount of user generated corpus has created a strong need to exploit Sentiment Analysis of online web texts . Opinion Mining of these texts can open a big door to not only this language's and its speakers' properties but also the culture and practices of that language.

Sentiment Analysis is a natural language processing task that tries to identify nature of opinion in a piece of text. It can be with respect to a sentence, document or even aspects in sentences.
Key methods to extract/predict sentiment can be classified into three types.

- Using Machine Learning - Predicting data by applying supervised or semi supervised approach on features from the text .
- N-Gram Modeling/Bilingual Mappings : Using N-gram models along with training data for sentiment prediction.
- Using Subjective Lexicon : A Resource of words or group of words (phrases) with a polarity score assigned to each word. Score in this case points towards the properties bore by that word for categorization into positive, negative or neutral.

As Hindi is resource scarce language in terms of standard and labeled datasets, dependence on datasets which have low recall and coverage for classification tasks result in low precision and accuracy.
To solve this problem, a combination of above approaches have resulted in what we call Transfer Learning or Cross Lingual based approach,which is the task of predicting sentiments by testing in text of language L_target (in this case, Hindi) by using a classifier trained/labels on the corpus of another language L_source(English). This paper adopts the above approach of transfer learning between L_target and L_source to predict sentence level sentiment labels in Hindi text. We propose methods which are combination of

[1] http://sentiwordnet.isti.cnr.it/

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 373–382

above mentioned key methods. Method 1 uses Machine Learning techniques and classifiers such as Naive Bayes and SVM in predicting Sentence level score along with lexical resource to label data. Method 2 is a complete unsupervised approach which exploits highly accurate ESWN to label chunk level scores in sentence and hence calculate sentence level score using these chunk scores. Method 1 gives overall accuracy of 78.8 % and method 2 results in accuracy of 83.6 %.

2 Challenges

- **Weak Lexical Resources:** Sufficient resources like labeled data, Sentiment Tagged words, tools and annotated data for Hindi language are not available, and those which are available are not as good in coverage and accuracy standard as per its English counterpart. And Annotated corpora and tagger for Hindi language is not as good compared to English language, which makes the sentiment analysis task time consuming.

- **Free Word Order:** Word order plays important role in polarity detection. Hindi is a free word order language means there is no specific arrangement of words in Hindi language i.e. subject(S), object(O) and verb(V) comes in any order whereas English is fixed word order language i.e. subject-verb-object(SVO). Word order has a significant role in determining polarity of word and hence of sentences, documents which it is a building unit of.Even the slightest variations and changes in the word order affect the polarity label.

- **Multiple senses:** Words in Hindi language having same semantic meaning may occur in multiple contexts, making it tough to distinguish between senses and hence pick one of them.

- **Morphological Variations:** Hindi language is morphologically rich which means that lots of information is incorporated already in the words as compared to the English language.

- **Co-reference resolution:** Analysing multiple expressions that refer to the same thing. For example :
  - “गीता शाम को निकली और वह सब्ज़ियां खरीदने गयी” .
  - Transliteration : geeta shaam ko nikalee aur vah sabziyaan khareedane gayee
  - English : Geeta got out in the evening and she went to buy vegetables. “वह” also refers to गीता. This analysis is important while performing fine grained level sentiment analysis.

3 Literature Survey

A lot of work has been done until now in the field of sentiment analysis for Hindi language with purpose to classify text and create lexical resource. Existing multilingual and cross lingual sentiment analysis approaches involve extension of existing resources through translation, synset, concept linking to bridge the gap. Recent methods based on learning common vector spaces for multiple languages have also shown promise in some topics.

In terms of creating lexical resources and extensions, most notable contributions are from Amitava Das and Bandopadhya [1],in which they developed sentiwordnet for Bengali language by Word level lexical-transfer technique on English SentiWordNet using an English-Bengali Dictionary. They also devised four approaches to predict polarity of a word in [2]

A Fallback strategy was proposed by Joshi et al. in [3] for Hindi language to create lexical resource Hindi SentiWordNet (HSWN) based on its English format. H-SWN (Hindi-SentiWordNet) by using two lexical resources (English SentiWordNet and English-Hindi WordNet Linking) with using methods namely: In-language Sentiment Analysis, Machine Translation and Resource Based Sentiment Analysis. Bakliwal et al.[4] created resource using a graph based method .They depicted how the synonym and antonym relations can be used to generate the subjectivity lexicon by using the simple graph traversal approach with 79% accuracy on classification of reviews.

A Graph based method was proposed by Piyush Arora et al. [5] to build a subjective lexicon for Hindi language, using WordNet as a graph traversal resource. Small seed list of opinion words was initially built and by using WordNet and synonyms and antonyms of the opinion words were determined and added to the seed list. An efficient approach was developed by Namita mittal et al. [6] based on negation and discourse relation to identifying the sentiments from Hindi content by improving Hindi SentiWordNet (HSWN) by adding more entries. They also created the rules for handling negation and discourse and 80% accuracy was achieved by their proposed algorithm for classification of reviews.

Various alterations to features in training set with supervised approaches have been used in [7] [8] [9] [10] [11] [12] [13]

A simple technique to perform sentiment classification based on an unsupervised linguistic approach using SentiWordNet to calculate overall sentiment score of each sentence is expressed in [14].

In terms of approaches which involves cross lingual methodology,which means training in source language L_source and testing on target language L_target, following notable works have been published. Using an english dataset, two Hindi language training datasets are produced with different features by [15]. Balamurali (2012) used WordNet senses as features for supervised sentiment classification. They use the linked WordNets of two languages to connect the languages. [16]. Deep learning framework is used in [17] to learn feature representations for cross lingual approach.

But all of these approaches require at least some amount of labeled data and complete inhouse resources with training data, heuristics in that language and in case of cross lingual approach, involves dependency on resource of source language.

4 Experimental Setup

We conducted two experiments, one with dependency on Hindi lexical resource and supervised in nature and another unsupervised in nature with its dependence on lexical resource in English. For supervised approach, classifiers such as Naive Bayes and SVM are used, and for unsupervised approach, Google Translate is used translate chunks into L_target English and then we interlink chunk level sentiments to L_source for further processing.

4.1 Datasets

We have used data from following resources

4.1.1 Data Used

- Hindi News Sentences [18]

- English SentiWordnet(ESWN) as a lexical resource reference. [19]
  Entries in this resource is modified as per need of our task. Hence, for every word, it matches the POS tag, most common or most frequent used sense and then returns score as a tuple, in which first entry is the positive score of word and second entry is the negative score.Since we already know that second entry is the negative score, we do not necessarily put '-' (minus sign) infront of it, to indicate its negative nature.

  Examples:

  - ESWN_Score(good) : (0.75,0.0)
    It contains more than 8 senses, but it returns most commonly used sense.
  - ESWN_Score(evil) :
    (0.375,0.5) if it occurs as Noun and (0.0,0.875) if it occurs as adjective.

- Data extracted from websites such as
  www.patrika.com/gadgets/
  www.amarujala.com/
  aajtak.intoday.in/

4.1.2 Resource Contribution

2000 sentences with political domain as its background have been selected from above sources and have been manually labeled into three classes, Positive (P), Negative(N),& Neutral/Statement (S) on the basis of annotation guidelines.
Example:

लोगों को यह एक अच्छा व्यवसाय नजर आने लगा है।	**P**
भारत के लिए यह एक बुरा दिन साबित हुआ	**N**
नोटिस पर सुनवाई सोमवार को होगी	**S**

After one round of labeling sentence as positive,negative or neutral, the data was distributed into couple of more annotators who also labeled the data according to their understanding. Hence, each sentence was labeled by 3 annotators. The sentence with at least 2 similar label out of 3 were considered and one more round of annotations were conducted for them. At the end, the label having >75% inter annotator agreement were incorporated. This data also acts as our evaluation model on experiments mentioned in this paper.

4.2 Supervised approach using Hindi lexicon

This experiments uses HSWN to label sentence level polarities. After receiving labeled sentence level polarities, classifier is run on this data to predict unseen sentences into one of the two classes, positive and negative.

4.2.1 Preprocessing

The first task is to run several iterations of processing on data, in which each given sentence is checked for noise and entries other than Hindi words.Spelling mistakes are corrected so that parser produces as accurate parse trees as possible.
In example mentioned below , the original word in corpus with its translated English equivalent is mentioned in bracket, and then , the same word after spelling correction and its correct English form is mentioned.

1. अभीन्न (Abinn) -> अभिन्न (Integral)
2. हॉलमार्क (Halmark) -> हॉलमार्क (Hallmark)

The final task is appending the missing end marker of sentence “|”.

4.2.2 Feature Vector from Parser

Each hindi sentence is run through Shallow Parser [2] which produces a output which contains complete description of the word, its POS tag, its root form, morphological analysis and representation in WX notation. At a bigger level, chunks are also assigned heads and they have these properties too. Hence, whole sentence is now rich with linguistic features of all its words.
An example of feature set for a single word:
Hindi Word : असुविधा (English counterpart : Inconvenience)
असुविधा NN <fs af=’ असुविधा ,n,f,sg,3,d,0,0’ name=’asuviXA’>

4.2.3 Enhancing Feature Set

For each word in our parsed data, we incorporate not only its linguistic properties but also its polarity. We use lexical resource HSWN(Hindi SentiwordNet) [3] to retrieve polarities of words.

Algorithm 1

```
For Each word in Sentence
Search The word in HSWN :
if Present then
    append that particular polarity as a feature into existing feature set
else
    locate the English translated version in ESWN and append that polarity.
end if
```

While translating, sense in preserved by taking into account the POS tag. And in case of multiple senses present in the lexical corpus, we take into account the most commonly and frequent used sense. Hence, now we have a feature set which has both linguistic and polar properties.
Example :
अच्छा JJ <fs af=’अच्छा,adj,m,sg,,d,,’ name='अच्छा' posn=’110’,score = ‘0.75,0.0’>

4.2.4 Preparing Training Data

We convert our feature set into a metric of :
1) TF features
2) TF-IDF features.
The weight of a term that occurs in a documents is simply proportional to the term frequency, while a term’s inverse document frequency (idf) is inverse function of the number of documents in which it occurs.And,

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \qquad (1)$$

Python module Scikit Feature Extraction Module was used to perform this. Feature size obtained through this is roughly 20,000 which is pretty large to process with good results. Hence, we apply dimensionality reduction techniques such as Principal Component Analysis(PCA) [20] to reduce the feature size by 1/4th and the feature set now has a size of 5000.

[2] http://ltrc.iiit.ac.in/analyzer/hindi/

4.2.5 Assigning score to sentence

The net score of sentence is weighted summation of polarity scores of all of its words. These weights are designed through special heuristics which is mentioned below.

A. Negative Dominance
It states that given a sentence, if the sum of negative polarity of words is greater than 50% of sum of positive polarity, we classify the sentence as negative.

B. Chunk Rule
Given a chunk (format in which the sentences occur in dataset), if a chunk contains a NEG tag, it reverses the polarity of all the words and hence the sentence till then.

C. Inflected Case
The equations Given a word with inflected form, it is not necessary to have its polarity similar to that of root word and has different polarity assigned to it in HSWN. But in case, the inflected word in unavailable, we tend to derive its polarity from its root word, root word being detected from the feature vector produced by shallow parser (the second word depicting the lemma(root) of the word).

4.2.6 Classifiers

Naive Bayes and SVM are run onto this to classify sentence as Positive or Negative. Accuracy is measured through manually labeled corpus mentioned in contribution.

Naive Bayes

A Naive Bayes Classifier is based on Bayes' theorem and is particularly used when the input dimensions are high. Naïve Bayes classification is a text classification approach that assigns the class c to a given document d as :

$$c* = argmax_c P(c/d) \qquad (2)$$

Where P(c|d) is the probability of instance d being in class c [21].

SVM

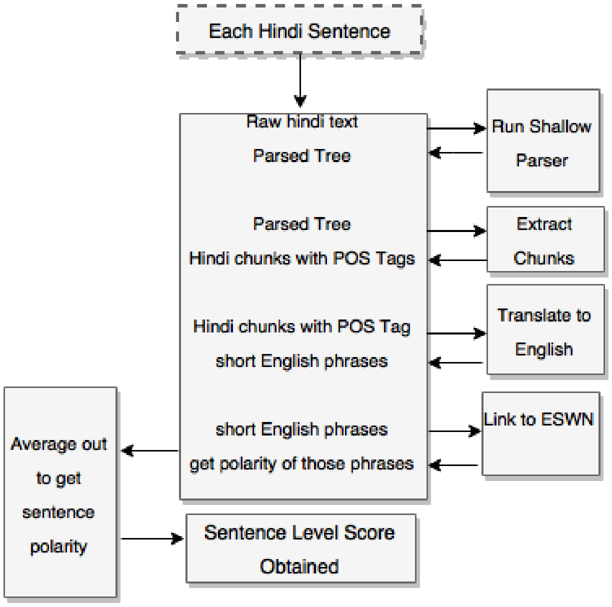
Support Vector Machine Classifier constructs N-dimensional hyper-plane represented by vector $\vec{w}$ which separates data into two categories.SVM takes the input data and for each input it predicts the class. SVM can be seen as a constrained optimization problem, in which class

$$c_{\rm j}1, -1 \qquad (3)$$

corresponds to either positive or negative class that belongs to document $\rm d_j$, the solution can be written as :
$\vec{w} = \sum_j \alpha_{\rm j} c_{{\rm j}{\rm d_j}}, \alpha_{\rm j} >= 0$
Where $\vec{w}$ is a vector, $c_{\rm j}$ is a class and $d_{\rm j}$ is a document [22].

4.3 Unsupervised Approach with Transfer Learning from English to Hindi

This approach works on the basis that each sentiment bearing word polarizes the words near it and hence the polarity around that word is similar to that of word. So, for each sentence, we extract chunk based polarity, with assumption that each polarity bearing word/words assign polarity to the whole chunk, and instead of using lexical resource in Hindi, we depend on already existing and quite accurate lexical resource in English, English SentiWordNet, to extract polarity scores for those Hindi to English translated chunks and then we transfer each chunk level score back to its original Hindi chunk, hence labeling every Hindi chunk with polarity score one by one. This approach rules out the dependence on Hindi labeled data and other resources which isn't that rich compared to ESWN.



Parsed Tree Example :

1	((	NP <fs af=शाम,n,f,sg,3,d,0में,0 head="शाम">
1.1	शाम	NN <fs af=शाम,n,f,sg,3,d,0,0 name="शाम">
1.2	को	PSP <fs af=को,psp„ „ „'>
	))	
2	((	NP <fs af=मौसम,n,m,sg,3,d,0,0' head="मौसम">
2.1	मौसम	NN <fs af=मौसम,n,m,sg,3,d,0,0' name="मौसम">
	))	
3	((	JJP <fs af=अच्छा,adj,m,sg„d„ head="अच्छा">
3.1	बहुत	INTF <fs af=बहुत,n,m,sg,3,d,0,0' poslcat="NM">
3.2	अच्छा	JJ <fs af=अच्छा,adj,m,sg„d„' name="अच्छा">
	))	
4	((	VM <fs af=हो,v,any,any,any„0,0' name="हो">
4.1	हो	VM <fs af=हो,v,any,any,any„0,0' name="हो">
4.2	गया	VAUX <fs af=जा,v,m,sg,any„या१,yA1' poslcat="NM">
	))	

Table 1: Shallow Parser Output

4.3.1 Sentence Parsing:

Given a sentence , Shallow parser is run on it, which gives complete analysis of a sentence in terms of Part of Speech , Morphology, Chunking etc. By using these properties, we will be able to predict overall sentiment score.

Algorithm 2

```
for Each Sentence S: do
    S_parsed = Shallow Parser(S)
end for
```

For example, given the sentence :
Hindi : शाम को मौसम बहुत अच्छा हो गया.
Transliteration : Shaam Ko Mausam Bohot Achha Ho Gaya
English : Weather got really good in the evening.
Shallow parser output is shown in Table 1(above)
As seen, the whole sentence can be represented as group of various chunks, with chunk heads. We extract these chunks along with their POS tags, and proceed to chunk processing step.

4.3.2 Chunk Processing Step:

In this step, we have chunks of sentences with their POS tags. The algorithm for this is :

Algorithm 3

```
for each sentence S do:
    for each chunk C in all chunks of S:
do
        Translate(Chunk_hindi)-(Chunk_english)
    end for
end for
```

In this step, each Hindi chunk of a sentence in translated to english,& since we are translating chunk with max of 5-6 words per chunk,expected translation error is pretty low as compared to translation of complete sentence, which will help us to effectively map sentiments,without using any hindi resource and with assumption that sentiment remains constant across these chunks.

4.3.3 Finding Sentiment:

Now, Each sentence S is a group of translated english chunks(E_Chunks). For each english chunk, we find its sentiment according to following algorithm:

Algorithm 4

```
for for each english chunk C in E_Chunks:
do
    for for each word w in chunk C: do
        if if word in ESWN and its POS tags
match then
            assign score to the word.
        else if Word is present but not in
root form then
            convert word to its root and go to
step 1
        else
```

```
            assign score as 0,0.
        end if
    end for
end for
```

This algorithm assigns each word with a specific polarity taking into account its context as well. POS tags of words are used to distinguish between word senses, For example:
In the given phrase, ‘ he has the will to live’, the word ‘will’ is having NN as its POS tag, while in the phrase, ‘I will go there tomorrow’, the word ’will’ has VB tag.

So, it will have different score with respect to its manner/sense of occurrence in the sentence. Therefore, the POS tags need to match with the one in lexical corpus to match the correct sense and therefore attach correct score to the word. For words with multiple senses and hence different scores, the most commonly used sense is used for reference.
After marking each word with polarity score, we can have two approaches to assign score to the chunk:

Algorithm 5

```
if if a chunk contains more than one polarity bearing word: then
    chunk_score = max(score of all polarity bearing words)
else
    chunk_score = score of polarity or opinion bearing word
end if
```

This step assigns a polarity score to the current translated English chunk. Now, we transfer this score back to its original untranslated chunk, and after retrieving polarity scores all chunks, we calculate sentence level polarity by averaging out the chunks’ score with total number of chunks.
Here, total number of chunks are those which actually contain any amount of polarity score and are not completely neutral.
So, an important point to note in this step is that not every chunk contributes to the overall polarity score of a sentence, while some chunks might have net polarity as (p:0,n:0), other might not have any polarity score due to their semantic and syntactic space, and their strict objective nature.

Example 1(positive label)

- Sentence: व्यापार में बेहतर काम उपभोक्ताओं के लिए लाभप्रद होता है ।

- Transliterated : vyaapaar mein behatar kaam upabhoktaon ke lie laabhaprad hota hai

- English : Better work in business is profitable for consumers.

- Chunked: (व्यापार में)_NP (बेहतर काम)_NP (उपभोक्ताओं के लिए)_NP (लाभप्रद)_JJP (होता है)_VGF

- English Chunks: (in buisness)_(better work)_(for the consumers)_(profitable)_(happens).

English Chunks	Polarity(pos,neg)
in business	(0.0,0.0)
better work	(0.875,0.0)
for the consumers	(0.0,0.0)
profitable	(0.25,0.0)
happens	(0.0,0.0)

Determining Average Polarity

Net average Polarity = (average positive polarity,average negative polarity)

- Average Positive Polarity : sum of all positive scores in chunks/ number of chunks having score >0

- Average Negative Polarity : sum of all negative scores in chunks/ number of chunks having score >0

Following the mentioned steps,in this case
Average polarity : (0.56,0.0)
Since |positive polarity| > |negative polarity|
Label Generated : Positive

Example 2(negative label)

- Sentence: सही एडिटिंग न होने के कारण दूसरे हिस्से में यह फिल्म कमजोर हो जाती है।

- Transliterated : sahee editing na hone ke kaaran doosare hisse mein yah philm kamajor ho jaatee hai.

- English : Due to lack of proper editing, this film becomes weak in the second part.

- Chunked: (सही एडिटिंग)_NP (न होने के कारण)_NP (दूसरे हिस्से में)_NP (यह फिल्म)_NP (कमजोर)_JJP (हो जाती है)_VGF

- English Chunks :
  (correct editing)_(Reasons for not being)_(In second part)_(this film)_(weak becomes)

English Chunks	Polarity(pos,neg)
correct editing	(0.625,0.0)
Reasons for not being	(0.0,0.675)
In second part	(0.0,0.0)
this film	(0.0,0.0)
weak becomes	(0.125,0.5)

Average polarity : (0.25, 0.4)
Since |negative polarity| > |positive polarity|
Label Generated : Negative

4.4 Improvisation over Previous Experiment

A different scenario is observed when a chunk contains Negation tag. In most of the cases, It is seen that it negates the chunk/word just previous to it. Therefore, presence if 'NEG' tag can alter the chunk level and hence sentence level polarity calculated through the previous experiment.So, we incorporate this factor too, while predicting sentiments.

- If a current chunk has 'NEG' tag:
  - *It nulls the polarity of previous chunk if it is positive*, and
  - *strengthens/adds up to the previous chunk score if its already negative.*

For example:

- Sentence: फिल्म की कहानी अच्छी नहीं है.

- Chunked: (फिल्म की)_NP (कहानी)_NP (अच्छी)_JJP (नहीं है)_VGF.

- English chunks:
  (film's)_(story)_(good)_(is not)

English Chunks	Polarity(pos,neg)
film's	(0.0,0.0)
story	(0.0,0.0)
good	(0.875,0.0)
is not	(0.0,0.625)

- Sentence polarity without negation handling: (0.44, 0.31)
  - Which is positive (wrong label)

- Sentence polarity after negation handling: (0.0, 0.675)
  - Which is negative (correct label)

4.5 Results

Results depict that with supervised approach, best case accuracy is 48.8% in case of Naive Bayes and 78.2 % in case of using SVM as our classifier,which is our baseline. In unsupervised transfer learning based approach, the accuracy is 83.6% which indicates the importance of lexical coverage and wideness if the experimental approach is corpus based.

Naive Bayes	**Result**
TF	40.8%
TF with heuristics	42.6 %
TF-IDF	44.6 %
TF-IDF with heuristics	48.8 %

Table 2: Naive Bayes

SVM	**Result**
TF	62.7%
TF with heuristics	64.4 %
TF-IDF	74.6 %
TF-IDF with heuristics	**78.2%**

Table 3: SVM Classifier

Experiment	**Result**
transfer learning sentence level	82.2 %
Transfer learning with negation handling	**83.6%**

Table 4: Transfer Learning

5 Conclusion

The experiments state one thing very clearly, that the performance of system is in accordance with the lexical resource at disposal, if any. When we used Hindi lexicon, the performance was not as good even though it was a supervised learning approach. The problem lies with the fact that Hindi SentiWordnet is limited in its coverage area and is not as extensive and rich in word level sentiment coverage as its English counterpart. For example,Basic word such as ‘नहीं ‘ isn’t present in the list. So, most of the sentiment bearing words couldn’t get sentiment labels, and the translation approach used to enhance the coverage depends on sense present in ESWN and acquired by parsed output. Although translator is not up to the mark for all time, as the sentence length shrink to 4-5 words, it performs decent enough to capture the underlying sentiment in that chunk.

Naive Bayes and SVM performance were hence, not very effective. When switched to lexical resource in L_target English, and unsupervised approach, the accuracy is increased because sentiment across language remains as preserved as possible because of minimal translation error and better coverage.

This approach is also important because the complete experiment depends on translation and ESWN and the problem of low coverage of lexical resource and no good training data in resource scarce language doesn’t comes to picture.

One important thing using translator is the error while translating chunks having co-reference to other part, or when the sentence structure is of a very casual conversation. For Example :

- अपने भविष्य की कोई खबर नहीं है उसे
- Transliterated : bhavishy kee koee khabar nahin hai use
- Translation Output : There is no news of his future
- Correct Translation : He has no idea of his Future.

This makes it difficult in capturing the essence of sentence and it becomes more of a general statement than a concern and hence looses the sentiment tag. While these errors were less in number as chunks were rarely greater than 3-4 words, its important to take these semantic points to get better idea of what’s in the sentiment property of every sentence.

6 Future Work

Future work involves extension of our contributed dataset to aspect level and increasing its size to make it more useful and effective for purpose of Sentiment Analysis in various domains.

In second approach, incorporating more semantic and sense information while translating and taking into account the contribution of nearby chunks in determining a particular chunk polarity can increase the accuracy. The relation between chunks can help semantic properties intact. Also, a sophisticated mathematical model can be developed to figure out sentence level polarity instead of averaging the chunk scores.

References

[1] Amitava Das and Sivaji Bandyopadhyay. Sentiwordnet for bangla. *Knowledge Sharing Event-4: Task*, 2, 2010.

[2] Amitava Das and Sivaji Bandyopadhyay. Sentiwordnet for indian languages. *Asian Federation for Natural Language Processing, China*, pages 56–63, 2010.

[3] Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON*, 2010.

[4] Akshat Bakliwal, Piyush Arora, and Vasudeva Varma. Hindi subjective lexicon: A lexical resource for hindi polarity classification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, 2012.

[5] Piyush Arora, Akshat Bakliwal, and Vasudeva Varma. Hindi subjective lexicon generation using wordnet graph traversal. *International Journal of Computational Linguistics and Applications*, 3(1):25–39, 2012.

[6] Namita Mittal, Basant Agarwal, Garvit Chouhan, Nitin Bania, and Prateek Pareek. Sentiment analysis of hindi review based on negation and discourse relation. In *proceedings of International Joint Conference on Natural Language Processing*, pages 45–50, 2013.

[7] Amandeep Kaur and Vishal Gupta. A survey on sentiment analysis and opinion mining techniques. *Journal of Emerging Technologies in Web Intelligence*, 5(4):367–371, 2013.

[8] Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 650–655. Springer, 2015.

[9] Subhabrata Mukherjee, Pushpak Bhattacharyya, et al. Sentiment analysis in twitter with lightweight discourse analysis. In *COLING*, pages 1847–1864, 2012.

[10] Christiane Fellbaum. *WordNet.* Wiley Online Library, 1998.

[11] Akshat Bakliwal, Piyush Arora, Ankit Patil, and V Verma. Towards enhanced opinion classification using nlp techniques. In *Proceedings of the 5th international joint conference on natural language processing (IJCNLP). Chiang Mai, Thailand*, pages 101–107. Citeseer, 2011.

[12] Richa Sharma, Shweta Nigam, and Rekha Jain. Polarity detection movie reviews in hindi language. *arXiv preprint arXiv:1409.3942*, 2014.

[13] Shriya Se, R Vinayakumar, M Anand Kumar, and KP Soman. Predicting the sentimental reviews in tamil movie using machine learning algorithms. *Indian Journal of Science and Technology*, 9(45), 2016.

[14] Monalisa Ghosh and Animesh Kar. Unsupervised linguistic approach for sentiment classification from online reviews using sentiwordnet 3.0. *Int J Eng Res Technol*, 2(9), 2013.

[15] Poonam R Gohad and Archana S Vaidya. Hindi opinion mining for opinion target extraction. *International Journal of Engineering Science*, 6733, 2016.

[16] AR Balamurali. Cross-lingual sentiment analysis for indian languages using linked wordnets. 2012.

[17] Sarthak Jain and Shashank Batra. Cross lingual sentiment analysis using modified brae. In *EMNLP*, pages 159–168, 2015.

[18] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *LREC*, pages 759–765, 2012.

[19] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[20] Ian Jolliffe. *Principal component analysis.* Wiley Online Library, 2002.

[21] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2):103–130, 1997.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.