SENTIMENT ANALYSIS IN HINDI

Aakashdeep Singh Shubham Srivastava
231110001 231110049
asingh23@iitk.ac.in shubhamsri23@iitk.ac.in
Indian Institute of Technology, Kanpur

April 23, 2024

Abstract

Sentiment analysis, a fundamental aspect of text mining, entails the computational processing of sentiments, opinions, and subjective expressions within textual data. This paper focuses specifically on sentiment analysis in Hindi, recognizing its significance due to Hindi being the primary language for a substantial portion of India's population. The project aims to improve Hindi SentiWordNet via various methodologies, techniques that all fall under Natural Language Processing.

1 Introduction

Sentiment Analysis (SA) is a focal point in text mining research, focusing on the computational analysis of sentiments, opinions, and subjective expressions within text. With the abundance of web-based content in Indian languages like Hindi, Marathi, Kannada, Tamil, etc., analyzing this data has become notably important for extracting valuable insights. Considering that Hindi is the primary language for a significant portion of India's population, performing SA in Hindi has emerged as a crucial endeavor, especially for businesses and government entities.

Sentiment Analysis performs subjectivity classification which deals with identifying the clause of the sentence and then classifies the opinionated text into positive or negative. Work on SA for English language is going on for

quite some time now. We also have efficient resources that make it an easier task to perform the same. However looking at Hindi, we also have quite some progress but the sheer difference in English and Hindi Wordnets is evident based on their size.

In this project, we aim to improve upon the existing Hindi SentiWordnets. Though resources are being available for Hindi language they are limited and not that efficient as compared to English language.

In this report, we start with discussing related work and then propose our methods for improving existing HindiSentiWordNet. Then we describe these methods in detail in the next section. Finally, we conclude with results and scope for future work.

2 Related Work

Sentiment analysis, the process of computationally identifying and categorizing opinions expressed in text, has garnered significant attention in recent years due to its wide range of applications in various domains such as social media monitoring, product reviews, and customer feedback analysis.

Very few research work has been done related to sentiment analysis in Hindi. The earliest of them was by Aditya Joshi et al [11]. They proposed that a fall-back strategy could be adopted for doing sentiment analysis for a new language. They suggested that we could first of all train a sentiment classifier on in-language labeled corpus and use it to classify a new document.

An important contribution to Hindi Polarity Classification was done by Bakliwal et al [9]. Their major contributiob was that they created a resource for Hindi by using Hindi WordNet to retrieve synonyms and antonyms of a given word in Hindi for which they knew the polarity and then assigned the similar polarity to synonyms and opposite polarity to antonyms.

An efficient approach was developed by Namita mittal et al. [12] based on negation and discourse relation for identifying the sentiments from Hindi content. The annotated corpus for Hindi language was developed and existing Hindi SentiWordNet (H-SWN) was improved by incorporating more opinion words into it.

3 Proposed Idea

In our project, we have analyzed Sentiment Analysis for Hindi Language. Through a on the State-of-the-art Techniques, We formulated several techniques for entiment Analysis.

- In Language Classification
- Resource Based Classification based on H-SWN
- Adding similar words to H-SWN using FastText Embeddings.
- Lexicon Creation via manual seeding
- Translating to English and using EnglishSentiWordNet and augmenting H-SWN using English-Hindi Dictionary mappings.

4 Methodology

The dataset used for our various methods uses are Hindi Sentiment WordNet (H-SWN) http://www.cfilt.iitb.ac.in/wordnet/webhwn/, English Sentiwordnet https://github.com/rmaestre/Sentiwordnet-BC, Hindi Shabdanjili, HindiStopwords, Positive-Negative Sentiment Example Sentences and FastText.

The methodologies involved are elaborated below:

4.1 In Language Classification

This approach is based on training the classifiers on the same language as the text. It relies heavily on availability of resources in the same language to analyze the sentiment.

The feature representation (Term frequency or TF-IDF) can be varied to see the effect on In-language classification on Hindi reviews. In this approach, we use a variety of classifiers to train and test the data.We know TF-IDF can be a better way of feature matrix generation as it reduces effect of very frequent words in document but do not contribute much to the relevance of text.

4.2 Resource Based Classification based on H-SWN

In this classification approach, we employ Hindi SentiWordNet (H-SWN) as the primary resource to construct a sentiment classifier based on majority opinion. Each word contained in H-SWN is assigned both a positive and a negative sentiment score. By determining the maximum score among these, we assign a polarity to each word within a review. In the subsequent methods, we use Resource Based Classification for testing any augmentations made to H-SWN dataset.

4.3 Adding similar words to H-SWN using FastText Embeddings

FastText Embeddings capture semantic similarities between words, which can help expand the coverage and accuracy of sentiment analysis.

One way to leverage FastText embeddings to improve H-SWN is by enriching the dataset with similar words identified through FastText embeddings. We generate 2 versions of H-SWN based on this methodology, One containing 7 similar word embeddings and another containing the most similar word embedding.

4.4 Lexicon Creation via manual seeding

In this approach we are trying to create hindi language lexicons from manual seeding process where we have take 50 odd hindi words and have associated sentiment values to each word. It's either 0 for +ve and 1 for -ve and 2 for neutral. Then we are using pyiwn: A Python-based API to access Indian Language WordNets to access the hindi wordnets and assigning the same sentiments to its synonyms as well. If it's already present then we are adding on top of those words and then normalizing it so that sum of the values ranges between 0 and 1. This would be a common way to create lexicon for our specific interest fields and the particular domain on which we are working.

4.5 Translating to English and using EnglishSentiWordNet and augmenting H-SWN using English-Hindi Dictionary mappings.

Here, we are trying to leverage the large Sentiwordnet for English language for Hindi sentiment analysis. As we know, english sentiwornet has over 1.4 lakhs words in it. We have translated the entire set of positive and negative movie reviews sentences in English and try to predict it's sentiment using English Sentiword Net. The translation is done using a pretrained model readily available online (facebook/nllb-200-distilled-600M). Next, we also try translation based on online English to Hindi Dictionary Sabdanjali. Here, we have taken an online English to Hindi Dictionary and for each Hindi word we have extracted all the words in english that represents the data. Then we have taken the sentiment values of those English Words from English SentiWord net and then normalized the values for each Hindi Word. We have a created a sentinet for hindi words based on this.

5 Results

In this section, we present the findings and assessments derived from our investigation into sentiment analysis for Hindi. We meticulously crafted some methodologies and conducted relevant experiments, assessed performance metrics, and compared outcomes to provide a comprehensive understanding of the state-of-the-art sentiment analysis methods tailored to the Hindi language.

Results for In Language Classification

```
###### Unigram+Tf Accuracies
### LogisticRegressionClassifier Accuracy ::   0.7941919191919192 +/-  0.050148300628362896
### LogisticRegressionClassifier F-measure ::   0.7995080888952094
### SGDClassifierClassifier Accuracy ::   0.773989898989899 +/-  0.04544287729027214
### SGDClassifierClassifier F-measure ::   0.7638603768710903
### SVCClassifier Accuracy ::   0.7711616161616162 +/-  0.05588775335805396
### SVCClassifier F-measure ::   0.7826649186638197
### NearestNeighbourClassifier Accuracy ::   0.5733030303030302 +/-  0.03544015506975424
### NearestNeighbourClassifier F-measure ::   0.4343337876897035
### NeuralNetworkClassifier Accuracy ::   0.7650404040404041 +/-  0.05407238645785128
### NeuralNetworkClassifier F-measure ::   0.7604895850169603
### DecisionTreeClassifier Accuracy ::   0.710919191919192 +/-  0.038566800783063994
### DecisionTreeClassifier F-measure ::   0.7093608015049823
### MultinomialNB Accuracy ::   0.7781111111111112 +/-  0.04756593542050134
### MultinomialNB F-measure ::   0.7796333227907226
### EnsembleClassifier Accuracy ::   0.7862424242424242 +/-  0.05092171159900489
### EnsembleClassifier F-measure ::   0.7752458634409508
```

Figure 1: Unigram-TF Results

```
###### Unigram+Tfidf Accuracies
### LogisticRegressionClassifier Accuracy ::   0.8443535353535353 +/-  0.008490672604295825
### LogisticRegressionClassifier F-measure ::   0.9151668716173349
### SGDClassifierClassifier Accuracy ::   0.9216666666666666 +/-  0.021469589448678005
### SGDClassifierClassifier F-measure ::   0.957943879743984
### SVCClassifier Accuracy ::   0.8864747474747474 +/-  0.02048605628698833
### SVCClassifier F-measure ::   0.9367730094870869
### NearestNeighbourClassifier Accuracy ::   0.8634040404040404 +/-  0.018995488646504938
### NearestNeighbourClassifier F-measure ::   0.9234301926800874
### NeuralNetworkClassifier Accuracy ::   0.9256464646464646 +/-  0.022260043572348925
### NeuralNetworkClassifier F-measure ::   0.959112062054175
### DecisionTreeClassifier Accuracy ::   0.9026060606060605 +/-  0.037803286369853986
### DecisionTreeClassifier F-measure ::   0.9439036408316029
### MultinomialNB Accuracy ::   0.8423535353535353 +/-  0.005050313127666536
### MultinomialNB F-measure ::   0.9141580443789994
### EnsembleClassifier Accuracy ::   0.9176161616161614 +/-  0.020688841446340017
### EnsembleClassifier F-measure ::   0.9538572668713396
```

Figure 2: Unigram-TFIDF Results

Results for H-SWN word augmentations

```
###### METHOD 2 :: SA using H-SWN(original)
Accuracy        ::  53.5140562248996
F-measure       ::  0.5270684371807968
```

Figure 3: Original H-SWN Results

```
(envname) [gauche@archlinux Method3]$ python3 Method3_test.py
###### METHOD 3 :: SA using H-SWN(modified) :: 7 words added
Accuracy        ::  49.39759036144578
F-measure       ::  0.2522255192878338
(envname) [gauche@archlinux Method3]$ python3 Method3_test.py
###### METHOD 3 :: SA using H-SWN(modified) :: Most similar word
Accuracy        ::  54.61847389558233
F-measure       ::  0.513978494623656
```

Figure 4: Augmented H-SWN Results

When we initially ran the sentiment analysis on Movie Reviews data using the existing Hindi Senti Wordnet then the results which we were getting in terms of Accuracy and F1 Score are as follows:-
Accuracy:: 53.5140562248996 F-measure:: 0.5270684371807968

In our next approach we were using the hindi sentiword net that we generated using Manual Seeding Approach to extracts the sentiments and the results are as follows:-
Accuracy:: 51.00401606425703 F-measure:: 0.672922252010724

In our next approach we translated the entire set of positive and negative movie review dataset in english language and leveraged English Sentiword Net to find its sentiments and the result are as follows
Accuracy:: 63.417085427135675 F-measure:: 0.6856649395509499

In our last approach we formed another Hindi Sentiment Dictionary using Sabdanjali online eng-hin dictionary and extracted the sentiments on the movie reviews in hindi language. The results are as follows:-
Accuracy:: 64.82412060301507 F-measure:: 0.7078464106844742

6 Future Works

Negation plays an important role specifically in written natural languages. Negation is mainly used to reverse the polarity of the statement. Also the popular and well-known problem associated with SA is Sarcasm. Though sarcasm is a very significant problem in SA it has been less explored in Indian languages,especially, Hindi. Along with negation handling and sarcasm, there are some other issues that are important and need to be significantly handled. Some of them are handling discourse connectors, conjunctions, contextual variances, and so on.

7 References

1. *Dhanashree S. Kulkarni and Sunil S. Rodd. 2021. Sentiment Analysis in Hindi—A Survey on the State-of-the-art Techniques. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 21, 1, Article 21 (January 2022), 46 pages. https://doi.org/10.1145/3469722*

2. *J. Wiebe. Learning subjective adjectives from corpora. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. AAAI Press, 2000*

3. *FastText. https://fasttext.cc/*