Project Documentation

Nishi Singh
HOOMACL201900025
MA Computational Linguistics

Introduction

Sentiment Analysis is a natural language processing task which helps to identify and categorize opinions expressed in a piece of text as positive or negative. It helps to determine the reviewer's point of view on a particular topic. It uses computational linguistics.
Hindi is the fourth highest speaking language in the world. Nearly spoken by 425 million people as the first language and 120 million people as a second language. A large part of data present on the internet is in Hindi, including different websites, blogs and tweets. Most of the research work done till now is mainly focused on English language and very less attention is paid in the direction of sentiment analysis in Hindi. Increasing user-generated content in Hindi on the internet is the main motivation behind this project.

Problem Statement

The idea of the project is to develop a model that can classify Hindi newspaper articles(or in general any Hindi sentence) as positive or negative by analysing the sentence and based upon the previous knowledge of the dataset, we can identify the given statements.
This project will be useful when we are reviewing a thousands of newspaper articles(or sentences) and we want to know the overall view of the sentiments of all the sentences, without manually reading them all.

Challenges

1. Hindi is a resource scarce language which causes problems in collection and generation of dataset. Also, there are hardly any parsers available for the language.
2. Hindi is morphologically rich as compared to English, meaning there are no specific arrangements of words in Hindi i.e. subject,object and verbs come in a specific order in English (subject is always followed by a verb and then an object), because of this fact it becomes easier to determine word polarity in English as compared to Hindi.
3. Limited resources like Hindi SentiWordNet(H-SWN) are available only. It consists of a limited number of adjectives and adverbs.

Applications

1. Sentiment analysis has been used by e-commerce companies for customer satisfaction reviews. We can estimate what percentage of users are happy with a product, and accordingly work on better strategies to grow the business.
2. Sentiment analysis is used everyday in social media, surveys, feedback and to identify people spreading hate speech.
3. To identify the detractors and promoters.

Word Done

Dataset Used

HindiSentiWordsnext.txt (file attached) contains the dataset with more than 1000 articles classified as positive or negative, I have taken 250 labeled articles from the dataset of IIT-B which contains 125 positive and 125 negative news articles. In addition to that 750 articles are taken from a website Jagran.com out of which half are marked as positive and other half are marked negative so as to keep the dataset balanced and unbiased.

Data Preprocessing

The first step was to remove all the words that do not contribute to accuracy of classification or in general the sentiment of the statement.

1. Punctuations: They are symbols like [,/*/-!$.?;'] which are used while writing to separate sentences and their elements.
2. Numbers: Numbers do not contribute anything to accuracy and have no meaning in sentiment analysis.So they are removed in Preprocessing.
3. Stop words: The natural language processing words which have very little meanings such as articles, pronouns and prepositions are denoted as stop words.
4. One length words.

Approach

The approach to predict the sentiment of an article was to use the prior polarity of the terms present in it. In order to find the polarity, a lexical resource was required. In this method I used Hindi SentiWordNet(H-SWN) as the resource for developing majority based sentiment classifiers.
Each word present in the H-SWN has a positive and a negative sentiment score. Based upon the maximum of the scores, a polarity is assigned to each word in a sentence. The polarity which covers the maximum number of words in the sentence is predicted as the sentiment of that article.

Algorithm / Logic

Input:
1) A list of articles, R which contain positive articles and negative articles in string format. R = $\{p_1, p_2....p_{m1}, n_1, n_2....n_{m2}\}$, where m_1 = number of positive articles and m_2 = number of negative articles
2) Hindi SentiWordNet dictionary dict which contains a tuple for every word having its positive polarity score and negative polarity score.
Output: A list P containing the polarity of articles.

Make a list of polarity of articles P=[]
For each article r_i in R
    Apply preprocessing on r_i
    Make a list of votes v=[]

```
        Initialize two variables x1=0, x2=0
        For each word w in r_i
                if w exists in dict
                        pos score, neg score = dict[w]
                        if pos score > neg score
                                v.append(1)
                                x1+=pos score
                        else
                                v.append(0)
                                x2+=neg score
                        else
                                ignore the word
        ENDFOR
        x = number of ones in the list
        y = number of zeros in list
        if x > y
                sense = 1 (here 1 denotes positive)
        else if y > x
                sense = 0 (here 0 denotes negative)
        else
                if x1 > x2
                        sense = 1
                else
                        sense = 0
        P.append(sense)
ENDFOR
```

Result and Output

The following evaluation measures in order to compute the overall performance of the system.
T_p = True positive (predicted positive and is positive)
T_n = True negative(predicted negative and is negative)
F_p = False positive(predicted positive and is negative)
F_n = False negative(predicted negative and is positive)
1. Precision: Precision is defined as a portion of true positive predicted instances among all positive predicted instances.

$$\text{Precision} = \frac{Tp}{Tp+Fp}$$

2. Recall: Recall is calculated as a portion of true positive predicted instances against all actual positive instances.

$$\text{Recall} = \frac{Tp}{Tp+Fn}$$

3. Accuracy: Accuracy basically is the portion of true predicted instances against all predicted instances.

$$\text{Accuracy} = \frac{Tp+Tn}{Tp+Tn+Fp+Fn}$$

4. F-measure: F-measure is the combination of Precision and Recall and is calculated as:

$$F - measure = \frac{2*Precision*Recall}{Recall+Precision}$$

The accuracy for the algorithm described above is 53.66%,whereas the 52.7%
A few sample output of the project are:

Article Headline	Sentiment Predicted by model	Actual sentiment of the article
BJP सांसद छेदी पासवान ने कहा - प्रदेश में अधिकतर हत्याएं विपक्ष के लोग ही करा रहे हैं!	Negative	Negative
पटनाः PMCH के गार्डों ने पत्रकार को पीटा, DS ने कहा - जो लिमिट क्रॉस करेगा, उसका यही हश्र होगा	Negative	Negative
नीतीश कुमार ने लोहिया पथ चक्र के निर्माण कार्य का लिया जायजा, कहा - 2021 के अंत तक होगा काम पूरा	Positive	Positive
'बाबा का ढाबा' के मालिक ने खोला नया रेस्टोरेंट, फेमस करने वाले गौरव वासन को लेकर कही ये बात	Positive	Positive
गांव में घर -घर चलती थी शराब की भट्टियां, बाराबंकी एसपी की प्यारी सी मुहिम ने बदल दिया नजारा	Positve	Positive
देश में बीते 24 घंटे में कोरोना के 18, 732 नए केस, संक्रमितों की संख्या हुई 101, 87, 850	Negative	Negative
ब्रिटेन से लौटे 2 शख्स निकले Covid - 19 पॉजिटिव, इंदौर में कोरोना वायरस के नए स्ट्रेन को लेकर मचा हड़कंप	Negative	Negative
जल्द दिल्ली पुलिस के जवानों को मिलेगी कोरोना वैक्सीन, मोबाइल पर भेजा जाएगा हर अपडेट	Positive	Positive
महाराष्ट्र: सरकार की रियल एस्टेट पॉलिसी का बीजेपी ने किया विरोध	Negative	Negative
भव्य आयोजन के साथ दुनिया के सामने लॉन्च हुआ अल बैन अली स्टेडियम	Neutral	Positive

How to Run

- The command used is: ResourceBasedSentimentClassification.py
- Make sure all the three files along with the python file are in the same folder.
- Install nltk using pip3 install nltk
- Install other packages like ‘re’, ‘sklearn’ and ‘pandas’ before running the program
- In case ‘punkt’ is not found download using

    import nltk
    nltk.download(‘punkt’)

Future Work and Areas of Improvement

- Since the accuracy of the model is not really good, we can explore a few more approaches that might give better results and handle edge test cases better like:
    - 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. Thus all training text, testing text are in Hindi language. 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.
    - Machine Translation Based Semantic Analysis: There is scarcity of resources in Hindi, that enforces us to take into consideration the machine translation based semantic analysis approach . The idea behind machine translation based semantic analysis is to model a classifier on standard English movie reviews. Then a translation module(here, Google Translate) is used to translate the reviews in Hindi to English. The model can then be used to classify the translated documents. Here, the result is reported only for TF-IDF representation of feature matrix run on Decision Tree classifier.
- In future, we can extend resource-based sentiment analysis to include Word Sense Disambiguation(WSD) so that a specific sense of word can be looked up in the H-SWN. Since Hindi SentiWordNet covers only a limited number of words at present, we can extend our work to cover more number of words by improving H-SWN. This will help us in achieving better accuracy. Further we can expand our approach to handle negation rules which is not supported by our present models.

References

- Arora, P. Sentiment analysis for hindi language. *MS by Research in Computer Science (2013).*
- Bakliwal, A., Arora, P., and Varma, V. Hindi subjective lexicon: A lexical resource for hindi polarity classification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC) (2012)*
- Bansal, N., Ahmed, U. Z., and Mukherjee, A. Sentiment analysis in hindi. *Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India (2013)*.

- Joshi, A., Balamurali, A., and Bhattacharyya, P. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON (2010).*
- Mittal, N., Agarwal, B., Chouhan, G., Pareek, P., and Bania, N. Discourse based sentiment analysis for hindi reviews. In *International Conference on Pattern Recognition and Machine Intelligence (2013), Springer, pp. 720–725.*
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12 (2011), 2825– 2830.*
- Sharma, R., Nigam, S., and Jain, R. Polarity detection movie reviews in hindi language. *arXiv preprint arXiv:1409.3942 (2014).*

Files attached

1. ResourceBasedSentimentClassification: This Module is used to do Resource based sentiment classification of hindi articles using HindiSentiWordnet as a resource.
2. pos_hindi.txt: This contains a dataset of positive hindi sentences, which are separated by $.
3. neg_hindi.txt: This contains a dataset of negative hindi sentences, which are separated by $.
4. HindiSentiWordnet.txt: This contains the dataset with more than 1000 words classified as positive or negative. In addition to that 750 articles are taken from a website Jagran.com out of which half are marked as positive and other half are marked negative so as to keep the dataset balanced and unbiased.