Method

Materials

The data/ corpus we use was captured using twitter api for dates 09-Apr-2017 to 20-Apr-2017 with filters for language type as “hindi” and geo location of India. Total tweets collected during this period were 169922 (including re tweets). In these 12 days some of the major events that occurred in India were “Sonu Nigams comment on azan”, “Gaurakhsaks of UP”, “Mahaveer Jayanti”, “Gurunank Jayanti”, “Hanuman Jayanti”, “CM of UP”, “Problems with EVM”, “Bail for Asaram”, “Romeo squad” , “Indian Soldiers”, “Kulbhushan Jadhav” etc.

The annotation for this data (classifying into positive (1), negative (-1), neutral (0)) is done manually for 2000 tweets. This hand annotated data is used to train our random forest model. Apart from this, we use word net to label the tweets as positive, negative neutral depending on the words in our tweets.

Procedure

We started by collecting twitter data including user handle, tweet, hashtag, date and time. The tweet data had new line chars, extra punctuations and garbage data. This data was then pre-processed to remove new line characters between the tweets. There were certain tweets which were partially in hinglish. These were converted to hindi using transliteration [https://github.com/koshalt/Transliteration]. Eg “Kya hai”, this was converted to “क्या ह”. Then we removed the stopwords using a list of stopwords in hindi which we obtained from IIT-Bombays website.

Once this was done, we moved on to removal of high frequency words not contributing to any information using Tf-IDF. We have written our own TF-IDF as we wanted it for hindi language.

So after having a clean image of the tweets with only rare and meaningful words, we used this data and applied LDA on the same. We have used genism library to achieve the same. This provides us with a graph of top 10 topics with their top 30 words. These 30 words change with the change in relevance parameter (slider). [Screenshots] From these keywords, we generate a baseline topic keywords and their sentiments by looking up the wordnet annotated tweets. And depending on the votes for neutral, positive or negative we decide the opinion of the people on twitter for this topic/ keyword.

For our actual approach, we use random forest. For the training set, we provide random forest with the 2000 (1700 for development and 300 for train set) hand annotated tweets which we have classified and by tuning its parameters and we generate our model. This model is then executed on our train set to measure the accuracy. From LDA topics/ key words, we extract tweets containing that particular word and annotate it using the model developed above. Once, the annotation is done, we do our final step of giving the sentiment of the particular topic/ key word.

Evaluation:

We hand annotated 400 tweets from by selecting tweets belonging to various topics given by the LDA, this forms as a baseline to evaluate our other methods for tweet classification.

We used the Hindi SentiWordnet [] to score each tweet based on the presence of unigrams. For a geven tweet we add the positive and negative score of each word and assign the tweet with the polarity of the maximum value. This classification gave us an accuracy of 62.37%, the system performs poorly because the difference in spelling in the tweets and a lot of tweets do not contain adjectives in the wordnet or contain adjectives with neutral polarity. On using stemming on the tweets the accuracy of this classifier increases by 2.3% i.e. 64.67%. This classifier was then applied to the complete tweet dataset.

The second classifier that we built was using random forests by using 3000 tweets to train and 1000 tweets as test data, this improved the accuracy of our classifier to 74.69%; using stemming further improved our accuracy by 1.73%.

s