Procedure

Downloading tweets and preprocessing

We started by collecting twitter data including user handles, hashtags, date and time. This data was then pre-processed to remove newline characters between the tweets. There were tweets which contained hinglish words, these tweets were then converted to hindi using transliteration[4] eg. “Kya hua” is converted to “क्या हुआ”(What happened?).

We then removed stopwords using a list of stopwords published by IIT-Bombay[6] and University of Neuchatel[17]. Once this was done, we moved on to removal of high frequency words not contributing to any information using TF-IDF. We tried to use the Scikit learn tf-idf algorithm but it does not work well for hindi language as it splits the words. Hence we wrote our own tf-idf algorithm and thereby retaining only 80% of the words in the corpus.

LDA for topic modeling

Now the clean tweets with only rare and meaningful words are used as input to the Latent Dirichlet Algorithm(LDA). We have used Gensim multicore library with input as number of topics, tweet corpus and fine tuned hyperparameters. This provides us with a graph of top 10 topics with their top 30 words. These 30 words change with the change in hyperparameters. From these keywords, we generate a topic keywords.

Manually Annotate Tweets

We manually annotated 2000 tweets giving each tweet one of the 3 classes: negative(-1), neutral(0), positive(1). This data is given as input to the Random Forest Algorithm for sentiment Analysis.

Sentiment Analysis using Hindi Wordnet(Baseline model)

For each manually annotated tweet , we calculate its sentiment using the Hindi WordNet. Hindi WordNet provides 3 scores for each word specifying how positive, negative or neutral the word is. We used POS tagging [13] for annotating the tweets and mapped this tag to the tag provided by Hindi WordNet. We then take the sum of the sentiment score for each word and each class in the tweet and compare the value for the classes. The class with highest value will give us the sentiment of the tweet. We calculate the accuracy of this model using the actual labels.

Sentiment Analysis using Random Forest Classifier

For our actual approach, we have used random forest multiclass classifier. Random forest tries to build multiple decision trees with different observations and different initial features . It will repeat this process multiple times and then make a final prediction on each observation by taking the mean of all the predictions of various decision trees. We use manually annotated tweets corpus for training the model. The corpus is split into 3:1 ratio(with random shuffle) for training and testing purposes which is approximately 1500 tweets for training and 500 tweets for testing.

For LDA topics/keywords, we extract tweets corresponding to each LDA topic/keyword and use our trained model to classify these tweets into positive, negative or neutral sentiment class. We calculate the accuracy of this model using the actual labels and compare with baseline.