Stance Detection

Abhishek Kumar (140101003)
Rajan Garg (140101057)
Ankit Kumar (140101010)
Abhinav Prince (140101001)

Introduction

- We define stance detection to mean automatically determining from text whether the author is in favor of a given target, against the target or whether neither inference is likely.
- Stance and Sentiment are two different aspects. A positive sentiment tweet might have negative stance and vice-versa.
  - Eg. AAP party being cleaned in Delhi using AAP jadu!

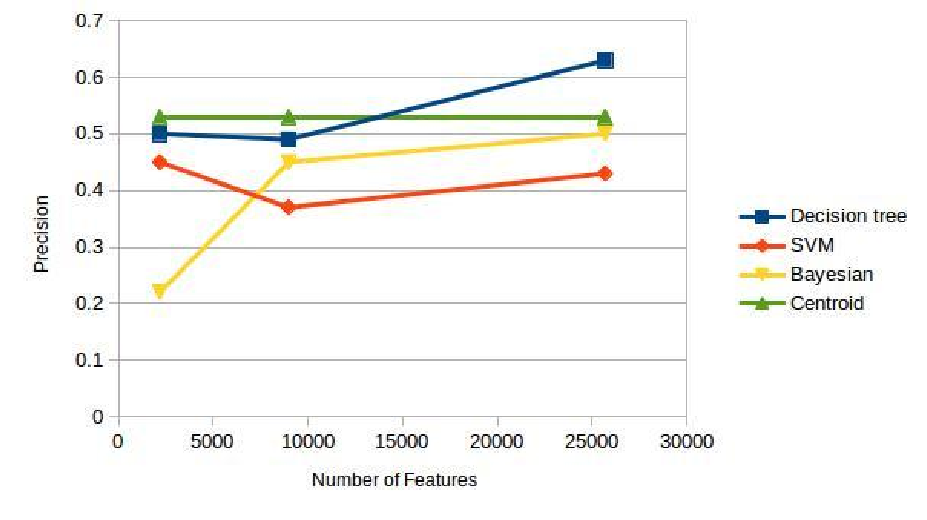
Approaches tried:

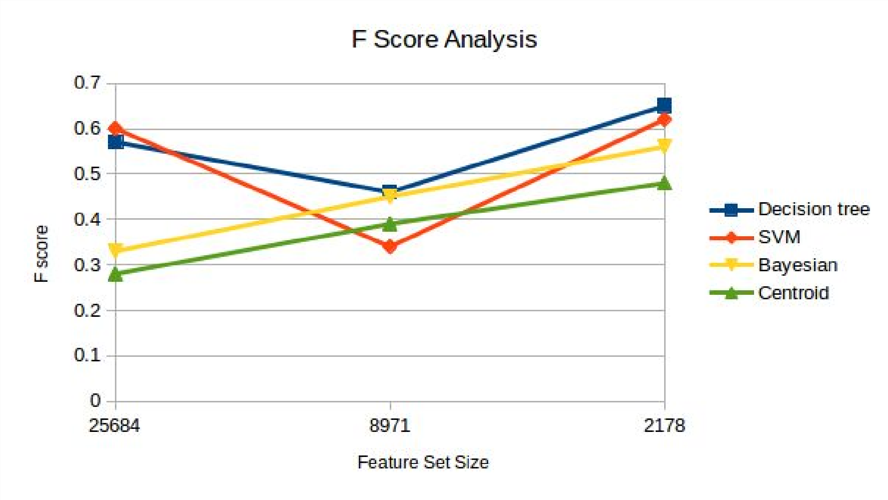
- Different ML models like SVM, Decision Trees, Bayesian Classification and Centroid Clustering method.
- Feature selection using variance based, chi 2, mutual information with decision trees applied on the reduced feature set.
- Better normalization of text.
- Based on semantic dependencies in words.

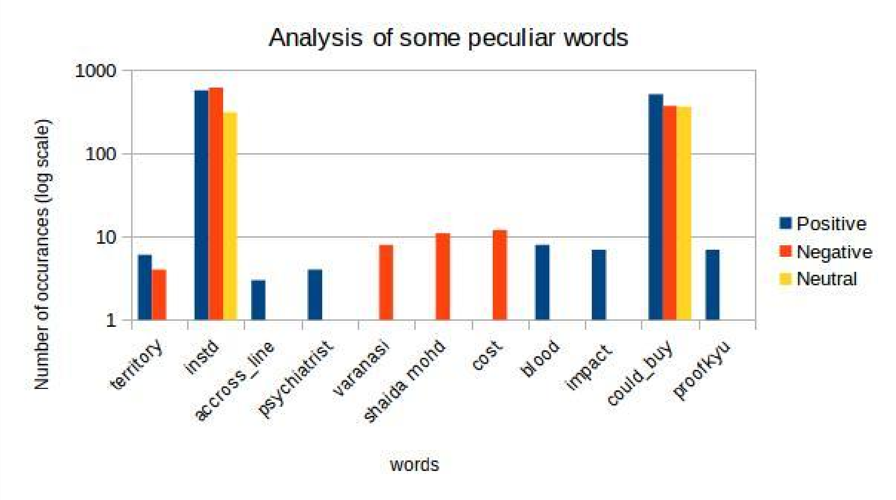
Definitions used to measure performance of methods:

- Let M_{ij} be number of terms belonging to class i {-1, 0, 1} which are categorised as class j.
- $Precision_i = M_{ii}/(M_{-1i} + M_{0i} + M_{1i})$
- $Recall_i = M_{ii} / (M_{i-1} + M_{i0} + M_{i1})$
- F_{1i} score = $2 / (Precision_i^{-1} + Recall_i^{-1})$

All plots show average scores.


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Precision
0
5000
10000
15000
20000
25000
30000
Number of Features
Decision tree
SVM
Bayesian
Centroid


F Score Analysis
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
F score
25684
8971
2178
Feature Set Size
Decision tree
SVM
Bayesian
Centroid


Analysis of some peculiar words
1000
100
10
1
Number of occurances (log scale)
territory
instd
accross_line
psychiatrist
varanasi
shaida mohd
cost
blood
impact
could_buy
proofkyu
words
Positive
Negative
Neutral

Expected reasons for wrong classifications:

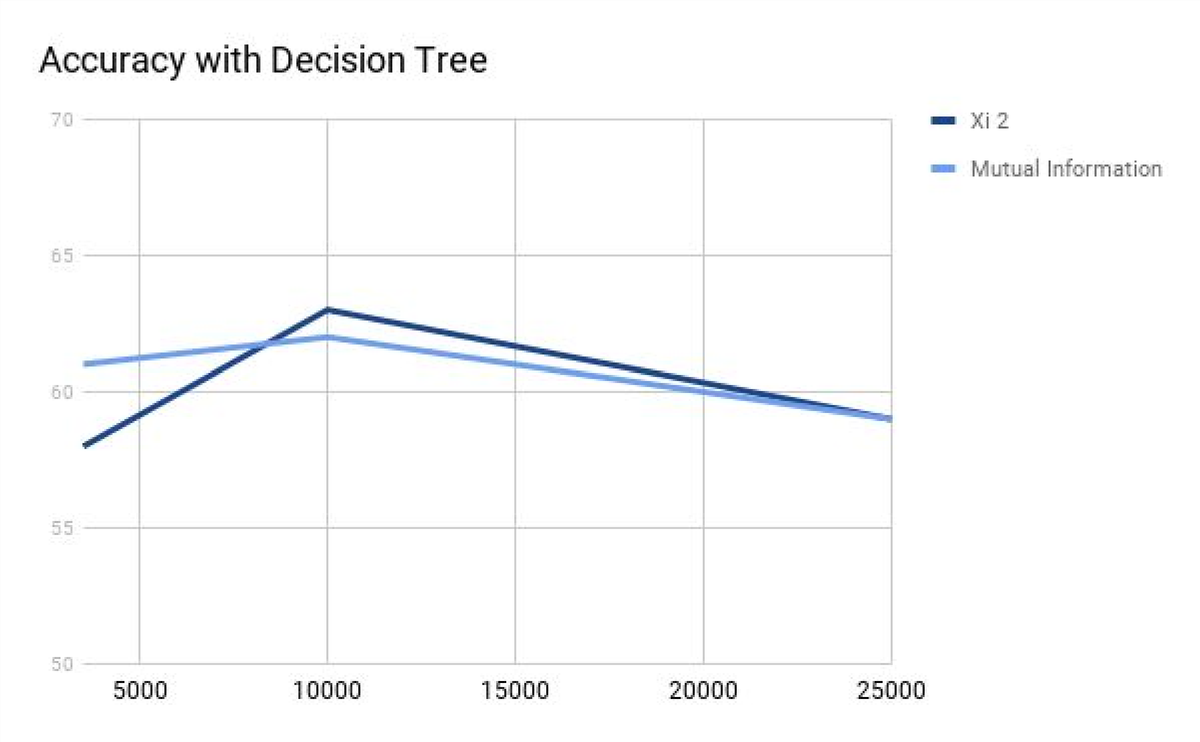
- Since the tweets had a significant fraction of words in Hindi, written in English. These words do not follow any consistent spelling. Later, we improve on this by using phonetics based lemmatization.
- Many English abbreviations were not consistent like “Government” was shortened to “gov” about 53 times in the whole corpus whereas it was shortened to “govt” 64 times. To solve this problem to a considerable level we have enriched our dictionary with a list of 75 twitter acronyms taken from “Business Insider” website, which are present in our corpus.
- Also, there were a lot of noisy features, which were removed as shown later.

Tweet Collection

- ❖ Another sub-part of this project was to come up with an effective technique to find tweets which are related to a particular target (eg. Surgical Strike).
- ❖ Since we already have some data based on the target, we utilized the information available about the frequency of words in various tweets related to the surgical strike.
- ❖ Following algorithm was used to crawl the tweets:
  - ➢ Get the list of lemmatized features (available from first part of this project).
  - ➢ Remove common dictionary terms.
  - ➢ Select top 50 words based on frequency with manual filter.
  - ➢ Use twarc API to crawl tweets for every pair of words in the above bag of words i.e. ${}^{50}C_2$ tweets.

Feature Selection

- Due to our N-Gram approach, we are having a very huge feature set. Which make processing and trying new techniques quickly not possible as well as it introduces a lot of noise.
- We studied different feature selection methods and present here analysis for the two methods which performed best - Xi Square selection and Mutual Information based selection.



Number of Features

We observe that Xi square method does better. Values of mutual information of most of the features with the target were very close and thus Xi square test provided a better separation of effective features from noise.

Better normalization of twitter data

- Tokenize each tweet, remove hashtags and other symbols.
- If a word belongs to the dictionary, leave it as it is. Else:
  - Get the recommended word by Hunspell based on spelling.
  - Get the phonetic key based on the soundex algorithm and get two closest ones based on Levenshtein distance.
    - This helped with certain short forms like : `reputation` -> `reputtn` and `government` -> `govmt`, spell checkers like Hunspell map “govmt” to “format” and “gourmet”, whereas soundex algorithm just gives a distance of 1 between “government” and “govmt”.
    - This also helped with spelling mistakes like “terorst”.
- Pick a word from the intersection of both the sets, if intersection is empty select one from the lowest Lavenshtein distance word from second set.

Feature selection results with text normalization

We observe that now with the new normalization technique accuracy has increased. Also, initially mutual information between target and features was very less. Zero for almost all, this was making feature selection not noisy. Now, we see mutual information technique performs better than Xi square technique.

F Score of 0.66 and recall - 0.69 was observed for 10,000 features with Decision Tree with Mutual Information as feature selection method.

Semantic Dependency Relations

- We also explored dependency relationships between different words in a tweet and how what kind of dependency it was using Stanford parser. We found results were best when only following classes of dependency were taken into account:
  - Accomp : relationship between verb and its object, eg. “looks” -> “beautiful”
  - Advcl, aux, amod
- These relationships combined with relationship tagging method shown in next slide helped significantly improve scores of very tweets like:

Baloch leaders welcome SurgicalStrikes, say gives them 'hope'

Score calculation method

- Based on existing data, two bag of words are formed:
  - Positive Bag : P(tweet is positive | w) >= k
  - Negative Bag : P(tweet is negative | w) >=k
- From each of the dependency relations, we select relations which have one word in positive or negative bag and then positive relationships or negative relationships are counted:
  - Positive relationship : Sentiment of other word is Pos and W in Pos bag or Sentiment of other word is Neg and W is in Neg bag.
  - Negative relationship : Sentiment of accompanying word is Neg and W in Pos bag or sentiment of accompanying word is Pos and W in Neg bag.

Results observed with Dependency Relations

- Accuracy : 58%
- Recall : 0.74
- F Score : 0.64
- Though this method doesn’t increase accuracy, we see a sharp increase in Recall. This increase in Recall is because we noticed with this method “clearly positive” or negative tweets as shown in previous slide are now getting almost 100% correctly classified. Though, this method performs poor in other cases and thus accuracy is not very high.

Papers referred:

- Exploiting Dependency Relations for Sentence Level Sentiment Classification using SVM. [http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7226110]
- A LEXICON BASED ALGORITHM FOR NOISY TEXT NORMALIZATION AS PRE-PROCESSING FOR TWITTER ANALYSIS
- Detecting Stance in Tweets And Analyzing its Interaction with Sentiment
- Performing Stance Detection on Twitter Data using Computational Linguistics Techniques
- Phonetics based idea is taken from Lexical Normalisation of Twitter Data
- Stanford parser dependency manual