Roman Urdu Sentimental Analysis

Executive summary:

Social media text usually comprises short length messages, which typically contain a high percentage of abbreviations, typos, phonetic substitutions, and other informal ways of writing. The inconsistent manner of text representation poses challenges in performing Natural Language Processing and other forms of analysis on the available data. Therefore, to overcome these issues, the text requires to be normalized for effective processing and analysis, we will be doing a type of sentimental analysis named Fine-grained Sentiment Analysis. In this work, we will perform text analysis on how the Roman Urdu language sentences. We have used different machine learning techniques such as clustering along with FastText, Naive Bayes and SVM in order to explain the process of the data deeply.

Before doing preprocessing, we cleaned the dataset for better quality. We dropped the null values, cleaned for missing column names and fixed incorrect values of sentiments (i.e changing negative into negative). We also did exploratory analysis on our original dataset to see the distribution of the words count. We found that neutral responses have 50% higher word count as compared with positive and negative responses. Thus , it is inferred that our data set is unbalanced.

For the preprocessing part, we removed punctuation marks and blank spaces, and changed all the text into lowercase. Not only removing stopWord as part of normalizing the dataset, we also looked more inside stopWord to find the most common stopword in our dataset. We also tried to remove suffixes through stemming for classification methods. However this did not work well for clustering as many words have been written similarly for a different meaning like “mein” means Me while some sentences have “Ma” used instead of “Mein” and “Ma” also means mother. So replacing sentences like these gave less meaning.

We did clustering with Kmeans clustering on all the sentiments as a whole initially from those results. We saw that most of the clusters were neutral sentiments, and only 2 were positive words segmented out. We separated each sentiment and performed clusters separately on positive, negative and neutral datasets to go further in-depth. From those clusters, we analyzed that the neutral dataset clustering had the most accurate results with most neutral words, which was followed by positive cluster results. The cluster results on negative sentiments were not that meaningful as compared to other results. To determine the value of k we used the knee-elbow method; from that method, we saw that choosing K from the knee-elbow method plot didn't give good clusters as there was not a similar point for the SSE curve starting to bend to make the elbow point. Hence we also choose K based on data domain knowledge and try different values for K. The worse results than sentiments were when performing clustering on Negative sentiments; hence we did try K as 5 and 6. A From the analysis of K-mean clustering, we determined that the best clusters turned out to K of 6

we got words in clusters that were primarily neutral while K with 5 had better segmentation of negative clusters; hence we used K=5. From the analysis, we can perceive that the Neutral sentiment segmentation had the most accurate results with a K of 5, although the elbow plot didn't have a bend in the plot at K = 5. The results can be dependent on the fact that we had more data for Neutral values to examine results with less bias, we should have more values for Positive and Negative sentences.

We used FastText in our project as well, although FastText could be used for generating vectors from words and doing analysis with classification using the FastText vector. However, we just used FastText to analyze words and their similarity using training the model and building a vocabulary upon it for our project scope.

We used MultiNominalNB and SVM classifiers for the classification by dividing our dataset into 80-20 ratio for training and testing along with tuning parameters. To compare the accuracy of two different classifiers, the approach we used different approaches, one removing stopWord using tfidf vectorizer and second one without removing the stopword along tfidf vectorizer. First classifier is MultiNominalNB and it gives around 60% accuracy without tuning parameters. In order to optimize alpha parameters for MultiNominalNB, a pipeline was used along with 5-fold validation. As a result, alpha = 0.55 is selected, which gives accuracy around 64%. Likewise, SVM classifier without a parameter gave 64% accuracy. In order to optimize, gmma = 5 and C=100 parameters were used along with 5 fold validation.

While doing classification process for the both approaches, multinomial naive bayes classifier gave 64% accuracy for both dataset with and without stopWord. Similarly, the SVM classifier gave 66% accuracy using dataset with removing stopword and 67% accuracy using dataset without removing stopWord. We found that both the approaches gave the almost same accuracy, finally, we went ahead creating a model using a dataset without having stopWord . We analyzed that due to an unbalanced dataset we are not getting proper accuracy even using both classifiers. We removed neutral from our dataset to balance it and did the classification technique on the new balance dataset. We observed that using a new balanced dataset we obtained accuracy 77% for both naive bayes and SVM using a tuning parameter.

Overall, we observed by applying stemming into our sentences, the sentences produced of words that didnt have meaning on its own concluding that stemming does not apply correctly on Roman-Urdu as it does on English language. However, we applied the stemming technique and tested it over clustering and our classification models and came to the conclusion that the accuracy drastically dropped from our models. So we did not use datasets with stemmed sentences. Furthermore, from the analysis between balanced(removing neutral response) and unbalanced dataset(before removing neutral response) with removing stopwords, we determined that balanced datasets gave overall high accuracy in comparison to unbalanced datasets. Lastly, from the analysis of two classifiers, we determined SVM is working better overall for a Roman urdu dataset. From Kmeans clustering, we have our findings that Neutral sentiment analysis has the most accurate data clusters, while negative clusters were the most segregated. If we could get a balanced data set for Neutral response too then the model could have improved a lot with all three responses(Neutral,Positive,Negative).

Contributions:

Data preprocessing was completed by all team members: Umaima Khurshid performed data stemming, Pinki Sharma did stop word removing and Rauf Nugmanov made TF-IDF transformation.
After that Umaima did extensive exploratory analysis followed by K-means clustering and explaining language specific characteristics. Rauf performed multinomial naive bayes classification and analyzed relationship between class sizes and outcome accuracy. Pinki built an SVM classifier and made a comparison of performance for each classifier. Pinki further did analysis on comparing the balance dataset(i.e removing neutral class response) and unbalanced dataset(i.e having all three classes) accuracy for both classifiers. She also tried model accuracy with and without removing stopwords.