FINAL TERM PROJECT

ROMAN URDU SENTIMENT ANALYSIS

NUMAN AHMED, AOUN JABBAR, WAZIR KHAN

1. Abstract

Sentiment analysis is a computational process to identify positive or negative or neutral sentiments expressed in a piece of text. In this paper, we present a sentiment analysis system for Roman Urdu. Considering the lack of publicly available benchmark datasets, we use a dataset which provides a Roman Urdu dataset which consists of 14646 sentiments annotated against positive, negative and neutral classes. To provide benchmark baseline performance over the presented dataset, we adapt diverse machine learning (Support Vector Machine, Logistic Regression, Naive Bayes, KNN), deep learning (recurrent neural network). From results of experiments, it can be concluded that LR performs better than NB and KNN and SVM in terms of accuracy.

2. Introduction

As indicated by SmartInsights study, People manages to publish 3.3 million posts on Facebook, 4.5 million tweets on Twitter inside a moment. These convincing insights of being dependent via web-based media stages are hoisting further with the speed of light. Considering the broad utilization of online media sites, extracting and breaking down client reviews identified with a certain occasion, issue, item, administration, association or big name, a committed assignment known as Sentiment Analysis has become a promising zone of Natural Language Processing (NLP).

Quite possibly the most enticing purposes behind broadly utilizing feeling examination is that it to a great extent helps the organizations to grasp shopper needs and plan basic changes in advertising and business methodologies to upgrade client experience.

In order to support the development of NLP applications for Roman Urdu, considering the deficiency of publicly available Roman Urdu dataset, the paper in hand presents the publicly available benchmark Roman Urdu sentiment dataset. A very limited sentiment analysis work exists for Roman Urdu which can be classified into lexicon based, machine learning, and deep learning based approaches. Lexicon based approaches have low applicability over unseen data, and machine learning based approaches predominantly use bag-of-words based feature representation approaches which face the problem of data sparsity. Whereas, deep learning approaches utilize less effective feature representation approaches like one-hot encoding or randomly initialized neural word embeddings asthere does not exist any pre-trained neural word embeddings for Roman Urdu.

3. Roman Urdu Sentiment Analysis

Roman Urdu is just a symbolic representation of linguistically rich but morphologically extremely complex language Nastaleeq Urdu just using English alphabets. Roman Urdu is extensively being used over diversified social media platforms mostly by Asian Users for communication and to express their opinions regarding certain brand, product or service. Despite not knowing English, user still manage to communicate using English like language. For instance, English word "Notorious" will always be written as in Nastaleeq Urdu and in Roman Urdu, it is written as "badnam". However due to the dynamic nature of Roman Urdu and not having any pre-defined linguistic and grammatical rules, there are several variations people use to write badnam such as "badnam", "badnaam", "bdnam", badnammm", "baadnam", "baddnam".

Data Set Information:

Tagged for Sentiment (Positive, Negative, Neutral)

Attribute Information:

Each record comprises of two string datatype values. One for Comment/Review and the second for sentiment.

4. Materials And Methods

This section briefly describes the Sentiments of Roman Urdu machine learning and deep learning approaches used for analysis. It discusses the specificities of developed benchmark dataset for the task of Roman Urdu sentiment analysis.

Naïve Bayes: Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = P(B|A)\,P(A) / P(B)$$

Logistic Regression: Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more

complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

KNN: The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. KNN algorithm is one of the simplest classification algorithm and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

SVM: A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text. SVM can be used for classification (distinguishing between several groups or classes) and regression (obtaining a mathematical model to predict something). They can be applied to both linear and non-linear problems. Until 2006 they were the best general purpose algorithm for machine learning. In 2006 Hinton came up with deep learning and neural nets. He improved the current state of the art by at least 30%, which is a huge advancement.

RNN: A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. ... Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. To perform SA we use **LSTM,** LSTM networks are a type of RNN that uses special units in addition to standard units. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time.

BERT: It is a Bidirectional Encoding Representations from Transformers. It's a bidirectional transformer pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia. It is simple & extremly powerful & open sourced by the team at HuggingFace.

5. Evaluation Measures

This section briefly discusses the evaluation measures used to compare the performance of proposed precisely extreme multi-channel hybrid methodology with

adapted machine and deep learning based methodologies. All utilized multi-class evaluation metrics are described below:

Accuracy: Accuracy is the proportion of correctly predicted samples to all types of predictions made by the model. Mathematically, it is defined as:

Accuracy (A) = (tp + tn) / (tp+ tn + fp + fn)

Precision: Precision measures how many samples that are predicted as positive by the model, actually belong to positive class. It can be defined in the following way:

Precision (P) = tp / (tp + fp)

Recall: Precision measures how many samples that are predicted as positive by the model, actually belong to positive class. It can be defined in the following way:

Recall (R) = tp / (tp + tn)

F1 Score: F1 Score is computed by taking the harmonic mean of precision, and recall. It is defined in the following way:

F1_Score (F) = (2 * P * R) / (P + R)

6. Results

	precision	recall	f1-score	support
0	0.52	0.44	0.48	711
1	0.64	0.78	0.71	1396
2	0.59	0.46	0.52	823
accuracy			0.61	2930
macro avg	0.59	0.56	0.57	2930
weighted avg	0.60	0.61	0.60	2930

0.6075085324232082

SVM MODEL

	precision	recall	f1-score	support
0	0.64	0.44	0.52	729
1	0.65	0.83	0.73	1392
2	0.66	0.53	0.59	809
accuracy			0.65	2930
macro avg	0.65	0.60	0.61	2930
weighted avg	0.65	0.65	0.64	2930

0.6501706484641638

LR MODEL

```
[570 132 239]]
              precision    recall  f1-score   support

           0       0.33      0.83      0.47       729
           1       0.67      0.30      0.41      1392
           2       0.60      0.37      0.46       809

    accuracy                           0.45      2930
   macro avg       0.53      0.50      0.45      2930
weighted avg       0.57      0.45      0.44      2930

0.4484641638225256
```

NAIVE BAYES MODEL

```
[[
              precision    recall  f1-score   support

           0       0.29      0.31      0.30       711
           1       0.52      0.64      0.57      1396
           2       0.48      0.26      0.33       823

    accuracy                           0.45      2930
   macro avg       0.43      0.40      0.40      2930
weighted avg       0.45      0.45      0.44      2930

0.45290102389078496
```

KNN MODEL

```
Test accuracy: 0.47337883710861206
```

RNN MODEL

7. Conclusion

This paper is about sentiment analysis for Roman Urdu text using machine learning models like Naïve Bayes, KNN, Logistic Regressions, SVM and deep learning RNN model. We have tried our best to perform well on these models to get better results and achieved 65% accuracy using LR model with a Roman Urdu dataset which contains 14646 comments with positive, negative and neutral classes. Also try to perform SA using BERT transformer but didn't achieved the goal very well. So at last the LR model gives us the best accuracy result as compare to the other models.

References

https://arxiv.org/ftp/arxiv/papers/2003/2003.05443.pdf

https://www.smartinsights.com/internet-marketing-statistics/happens-online-60-seconds/

https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

https://medium.com/analytics-vidhya/sentiment-analysis-on-roman-urdu-using-python-sklearn-and-nltk-c3a279ef7748

https://www.kaggle.com/ramanchandra/sentiment-analysis-with-bert-transformer

https://analyticsindiamag.com/step-by-step-guide-to-reviews-classification-using-svc-naive-bayes-random-forest/

https://ieeexplore.ieee.org/document/9223633

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

https://datascience.stackexchange.com/questions/9736/what-kinds-of-learning-problems-are-suitable-for-support-vector-machines

https://www.kaggle.com/owaisraza009/roman-urdu-dataset