SENTIMENT ANALYSIS FROM URDU LANGUAGE-BASED TEXT USING DEEP LEARNING TECHNIQUES

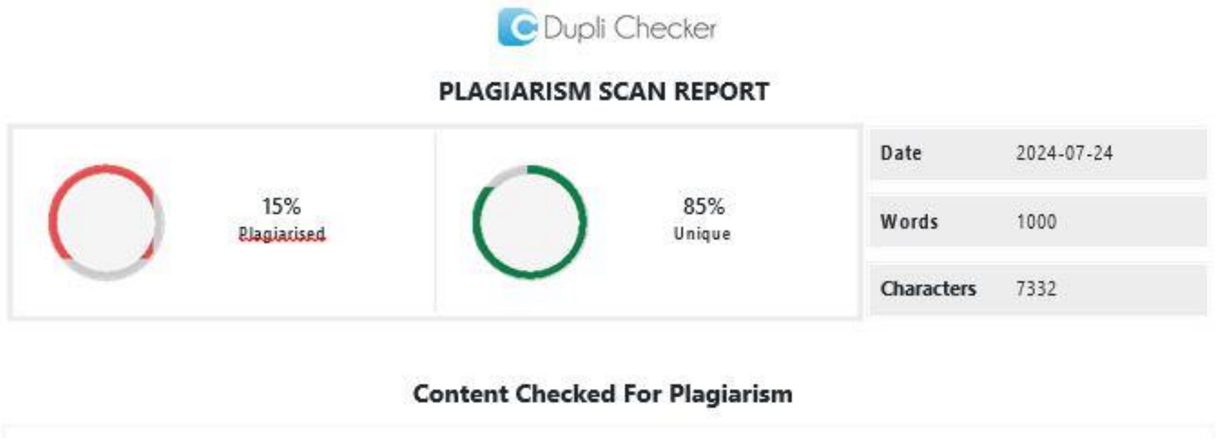
Submitted By:

Muhammad Afzal

L1F23PHDC0008

Hafiz M Hamza

L1S21MSCS0018



PLAGIARISM SCAN REPORT

15% Plagiarised

85% Unique

Date	2024-07-24
Words	1000
Characters	7332

Content Checked For Plagiarism

Sentiment Analysis from Urdu Language-based Text using Deep Learning Techniques

I. ABSTRACT

The Internet has seen substantial growth of domestic language data in recent times. It enables people to express their opinion by incapacitating the language walls. Urdu is a language used by170.2 million people for communication. Sentiment analysis is used to get sapience of people opinion. In recent times, experimenters' interest in Urdu sentiment analysis has grown. Operation of deep literacy styles for Urdu sentiment analysis has been least explored. There's a lot of ground to cover in terms of textbook processing in Urdu since it's a morphologically rich language. In this paper, we propose a frame for Urdu Text Sentiment Analysis(UTSA) by exploring deep literacy ways in combination with colorful word vector representations. The performance of deep literacy styles similar as Long Short-Term Memory(LSTM), 1D- Convolutional Neural Networks(CNN) IS estimated for sentiment analysis. piled layers are applied in successional model LSTM, In CNN, we used single complication subcaste. part of pre-trained and unsupervised tone- trained embedding models is delved on sentiment bracket task. The results attained show LSTM works good on this data and achieved the delicacy of 83 percent.

Keywords— UTSA, CNN, LSTM, Embeddings Deep Learning

II. INTRODUCTION

With the invention of internet many fields have gain a lot of attention in the modern days. In social media applications like Facebook, Twitter etc. People post a lot of comments. According to statistics there are 4.66 billion active internet users till October 2020. Many studies have been conducted in English to study the sentiments of people.

Sentiment analysis is a term used to extract subjective information by applying Natural Language Processing technique. It is used to classify tax as positive, neutral or negative regarding a product service topic event etc.

It can be applied at document level, sentence level or aspect level. There are three kinds of approaches being used:

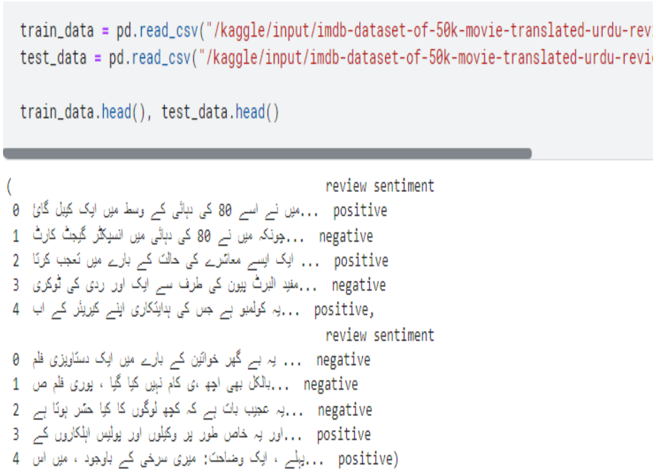
- Lexicon-based approach,
- Machine learning based approach and
- Deep learning-based approach.

In the study we have utilized publicly available IDBM datasheet which contain 50000 reviews and we have applied different machine learning and deep learning approaches to check the accuracy of our models.

III. METHODOLOGY

A. Dataset

we have taken publicly available dataset on Kaggle named as IMDB Dataset of 50K Movie Reviews. This dataset contains two files train and test csv files, having two columns named as reviews and sentiment. We have now 50,000 records available in our dataset



```
train_data = pd.read_csv("/kaggle/input/imdb-dataset-of-50k-movie-translated-urdu-rev
test_data = pd.read_csv("/kaggle/input/imdb-dataset-of-50k-movie-translated-urdu-revi

train_data.head(), test_data.head()
```

```
(                                              review sentiment
 0  میں نے اسے 80 کی دہائی کے وسط میں ایک کیبل گائ...  positive
 1  چونکہ میں نے 80 کی دہائی میں انسپکٹر گیجٹ کارٹ...  negative
 2  ایک ایسے معاشرے کی حالت کے بارے میں تعجب کرتا ...  positive
 3  مفید البرٹ پیون کی طرف سے ایک اور ردی کی ٹوکری...  negative
 4  یہ کولمبو ہے جس کی ہدایتکاری اپنے کیریئر کے اب...  positive,
                                               review sentiment
 0  یہ بے گھر خواتین کے بارے میں ایک دستاویزی فلم ...  negative
 1  بالکل بھی اچھ ہی کام نہیں کیا گیا ، پوری فلم ص...  negative
 2  یہ عجیب بات ہے کہ کچھ لوگوں کا کیا حشر ہوتا ہے...  negative
 3  اور یہ خاص طور پر وکیلوں اور پولیس اہلکاروں کے...  positive
 4  پہلے ، ایک وضاحت: میری سرخی کے باوجود ، میں اس...  positive)
```

fig 1.1: Dataset Preview

B. *Distribution of data*

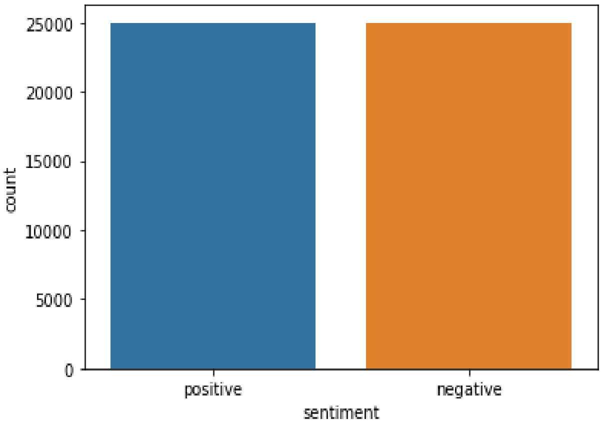


Fig 1.2

We can see that there are only two classes in our dataset:

- Positive means the review holds positive sentiment.
- Negative means the review holds negative sentiment.

Also, the class is very balanced. So, it will be easy for us to build any model.

C. *Preprocessing*

For this purpose, we used Urdu Hack library in python which provide extensive features to work Urdu language. First of all, our dataset contain train and test csv files and we merge them to make a single file. Then we have applied labeled encoding to label our sentiment columns.

Then we applied the following preprocessing to review columns to clean our data. After that we removed the stop words and lemmatized the reviews Now, we are ready to apply different deep learning approaches.

IV. Model Architecture:

A. *Machne Learning Models:*

1) *TF - IDF Vectorization*

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

We split the dataset into training, test sets. Then we trained following models and check their accuracy

- Support Vector Machine Classifier
- Logistic Regression Classifier
- Decision Tree Classifier
- Xgboost Classifier
- Random Forest Classifier

B. *Deeep Leaning Models*

For deep learning approaches, we have used Spacy library to tokenize our data, then we prepared word to vector Model

1) *Word to Vector Model*

Word2Vec consists of models for generating word embedding. These models are shallow two-layer neural networks having one input layer, one hidden layer and one output layer.

After that we implemented embedding layers then we train our two models

- 1D-CNN
- LSTM

V. Results

A. *Machine Learning Models*

Finally, in order to further evaluate the performance of these models, the accuracy, precision, recall, and F1-score metrics are calculated and compared in. These performance metrics are briefly introduced as follows

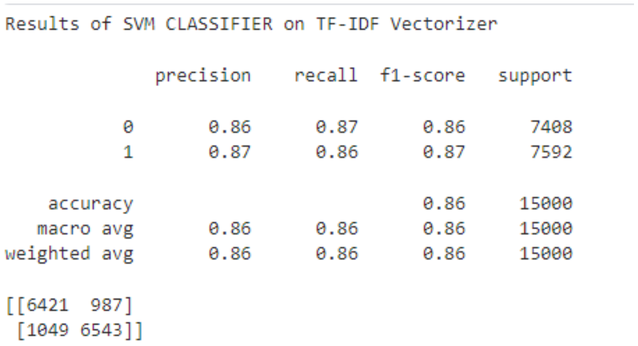
$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

$$\textbf{Precision} = \frac{\text{TP}}{\text{TP + FP}}$$

$$\textbf{Recall} = \frac{\text{TP}}{\text{TP + FN}}$$

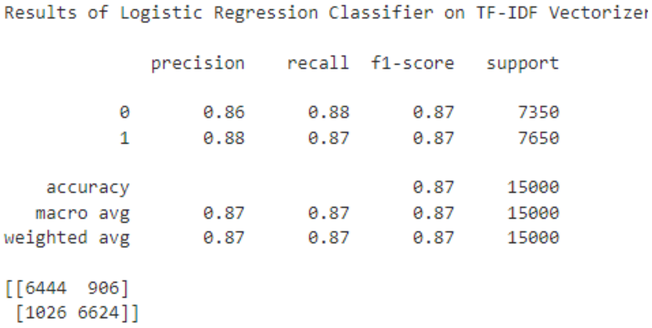
$$\textbf{\textit{F - measure}} = \frac{\text{2*Recall*Precision}}{\text{Recall + Precision}}$$

Now we will provide the result of different algorithms that we have mentioned above.



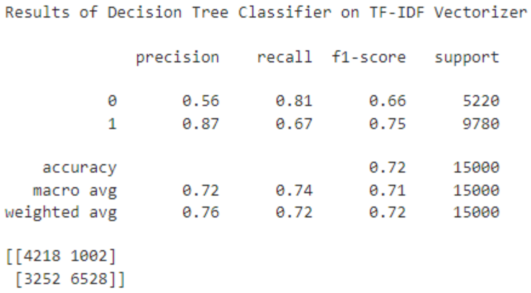
```
Results of SVM CLASSIFIER on TF-IDF Vectorizer

              precision    recall  f1-score   support

           0       0.86      0.87      0.86      7408
           1       0.87      0.86      0.87      7592

    accuracy                           0.86     15000
   macro avg       0.86      0.86      0.86     15000
weighted avg       0.86      0.86      0.86     15000

[[6421  987]
 [1049 6543]]
```

Fig 5.1 SVM



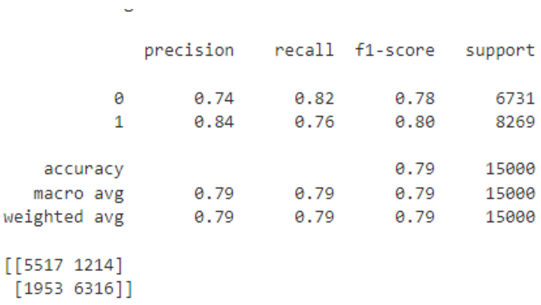
```
Results of Logistic Regression Classifier on TF-IDF Vectorizer

              precision    recall  f1-score   support

           0       0.86      0.88      0.87      7350
           1       0.88      0.87      0.87      7650

    accuracy                           0.87     15000
   macro avg       0.87      0.87      0.87     15000
weighted avg       0.87      0.87      0.87     15000

[[6444  906]
 [1026 6624]]
```

Fig 5.2 Logistic Regression



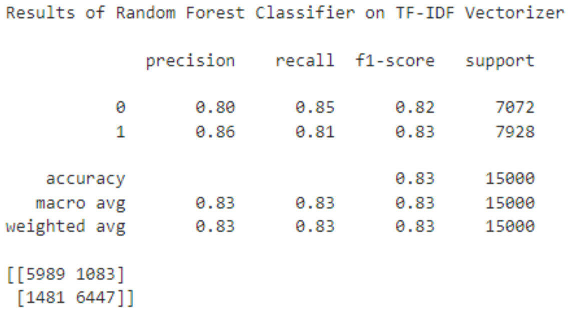
```
Results of Decision Tree Classifier on TF-IDF Vectorizer

              precision    recall  f1-score   support

           0       0.56      0.81      0.66      5220
           1       0.87      0.67      0.75      9780

    accuracy                           0.72     15000
   macro avg       0.72      0.74      0.71     15000
weighted avg       0.76      0.72      0.72     15000

[[4218 1002]
 [3252 6528]]
```

Fig 5.3 Decision Tree Classifier



```
              precision    recall  f1-score   support

           0       0.74      0.82      0.78      6731
           1       0.84      0.76      0.80      8269

    accuracy                           0.79     15000
   macro avg       0.79      0.79      0.79     15000
weighted avg       0.79      0.79      0.79     15000

[[5517 1214]
 [1953 6316]]
```

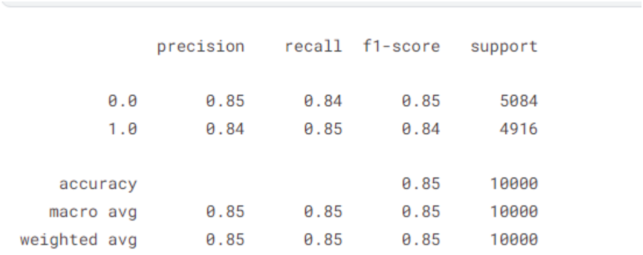
Fig 5.4 Xgboost Classifier



```
Results of Random Forest Classifier on TF-IDF Vectorizer

              precision    recall  f1-score   support

           0       0.80      0.85      0.82      7072
           1       0.86      0.81      0.83      7928

    accuracy                           0.83     15000
   macro avg       0.83      0.83      0.83     15000
weighted avg       0.83      0.83      0.83     15000

[[5989 1083]
 [1481 6447]]
```

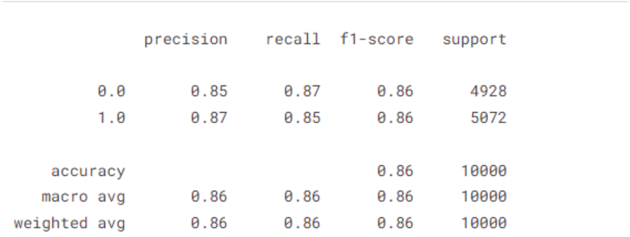
Fig 5.5 Random Forest

B. *Sentiment Analysis using Word to Vector and Deep Learning*



```
              precision    recall  f1-score   support

         0.0       0.85      0.84      0.85      5084
         1.0       0.84      0.85      0.84      4916

    accuracy                           0.85     10000
   macro avg       0.85      0.85      0.85     10000
weighted avg       0.85      0.85      0.85     10000
```

Fig 5.6 Confusion Matrix of 1D-CNN



```
              precision    recall  f1-score   support

         0.0       0.85      0.87      0.86      4928
         1.0       0.87      0.85      0.86      5072

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000
```

(Shabbir)
Fig 5.6 Confusion Matrix for LSTM

VI. Conclusions

We have preprocessed the text data using Urduhack library and applied multiple algorithms on this dataset and achieved maximum accuracy of 87 percent using Logistic Regression. we have achieved 88 percent accuracy using SVM without Urdu Hack but after preprocessing using Urduhack the accuracy falls down. So, we can conclude that the Urduhack text processing preprocess the text quite well but also affects the accuracy.

Then we build the Word to Vector embeddings and build deep learning models. We build 2 models, LSTM and Convolutional Network and compare their results and noticed that the LSTM works good on this data and achieved the accuracy of 83 percent. We also achieved the same accuracy with ConvNet but LSTM is more stable in results.

VII. Future Work

In coming days, we will also build Glove Embeddings for this data and train Deep Learning Models.

VIII. References

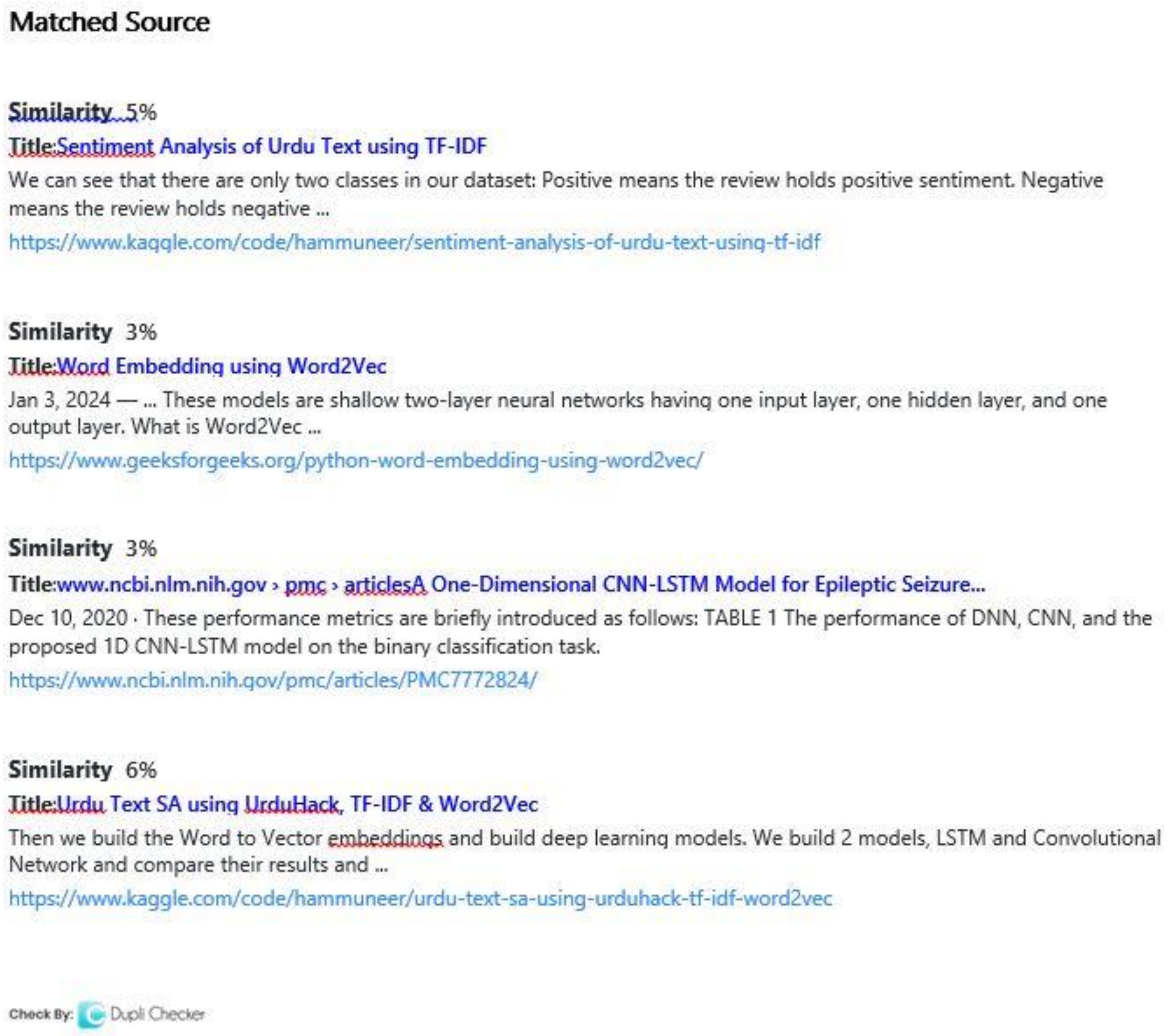
Ain, Q. T. (2017). Sentiment Analysis Using Deep Learning Techniques: A Review. *International Journal of Advanced Computer Science and Applications*.

Daud, A. (2016). Urdu language processing: a survey. *Springer Science+Business Media Dordrecht* .

Hafeez, B. (2023). Contextual Urdu Lemmatization using Recurrent Neural Network Models. *mathematics*.

Naqvi, U. (2021). UTSA: Urdu Text Sentiment Analysis Using Deep Learning Approaches. *IEEE Access*.

Shabbir, M. (n.d.). Sentiment Analysis From Urdu Language-based Text using Deep Learning Techniques. *2024*.



Matched Source

Similarity 5%

Title:Sentiment Analysis of Urdu Text using TF-IDF

We can see that there are only two classes in our dataset: Positive means the review holds positive sentiment. Negative means the review holds negative ...

https://www.kaggle.com/code/hammuneer/sentiment-analysis-of-urdu-text-using-tf-idf

Similarity 3%

Title:Word Embedding using Word2Vec

Jan 3, 2024 — ... These models are shallow two-layer neural networks having one input layer, one hidden layer, and one output layer. What is Word2Vec ...

https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/

Similarity 3%

Title:www.ncbi.nlm.nih.gov › pmc › articlesA One-Dimensional CNN-LSTM Model for Epileptic Seizure...

Dec 10, 2020 · These performance metrics are briefly introduced as follows: TABLE 1 The performance of DNN, CNN, and the proposed 1D CNN-LSTM model on the binary classification task.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7772824/

Similarity 6%

Title:Urdu Text SA using UrduHack, TF-IDF & Word2Vec

Then we build the Word to Vector embeddings and build deep learning models. We build 2 models, LSTM and Convolutional Network and compare their results and ...

https://www.kaggle.com/code/hammuneer/urdu-text-sa-using-urduhack-tf-idf-word2vec

Check By: Dupli Checker