Classification of Urdu News Articles Using Machine Learning: A Comparative Analysis

Muhammad Muiz Farhan
26100085@lums.edu.pk
LUMS

Muhammad Ibrahim
26100151@lums.edu.pk
LUMS

Abdul Samad
26100314@lums.edu.pk
LUMS

Haider Abbas Virk
26100032@lums.edu.pk
LUMS

Hussain Sulaiman Zia
26100011@lums.edu.pk
LUMS

ABSTRACT

This project focuses on the classification of Urdu news articles into five categories: Entertainment, World, Sports, Business, and Science-Technology. We employed web scraping to collect a labeled dataset from various online news sources. Three machine learning models— Multinomial Naive Bayes, Logistic Regression, and Neural Networks—were implemented for classification. All models achieved an accuracy exceeding 96%, with the Neural Network achieving the highest accuracy of 97.04%. This report details the methodology, findings, limitations, and conclusions of the study, emphasizing the reasons for model performance variations and potential areas for improvement.

KEYWORDS

Urdu text classification, machine learning, Naive Bayes, Logistic Regression, Neural Networks, web scraping

ACM Reference Format:
Muhammad Muiz Farhan, Muhammad Ibrahim, Abdul Samad, Haider Abbas Virk, and Hussain Sulaiman Zia. 2024. Classification of Urdu News Articles Using Machine Learning: A Comparative Analysis. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Text classification in low-resource languages like Urdu is challenging due to limited annotated datasets and the linguistic intricacies of the script. This project addresses these challenges by collecting a diverse dataset of Urdu news articles and applying machine learning models to classify them into five categories. The primary objective was to identify the most effective model and understand the reasons for performance differences.

The project involves web scraping, preprocessing, and implementing three machine learning models: Multinomial Naive Bayes, Logistic Regression, and Neural Networks. A detailed comparison of these models' performance is provided to highlight their respective strengths and weaknesses.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 METHODOLOGY

2.1 Data Collection

Urdu news articles were web scraped from three online news platforms:

- Jang
- Express
- Geo

The NewsScraper class used is designed to scrape news articles from a website across various categories like entertainment, business, sports, science-technology, and world. It fetches multiple pages of articles, extracts the title, link, and content of each article, and stores them in a Pandas DataFrame. The scraping process handles page requests, parses HTML to locate article details, and increments an ID for each scraped article. It handles errors gracefully and provides a summary of successful scraping for each page.

We ran it for each of the three websites, concatenating all 3 dataFrames at the end. A total of 1121 articles were categorized into the following five classes:

- **Entertainment:** 229 articles
- **World:** 229 articles
- **Sports:** 228 articles
- **Business:** 222 articles
- **Science-Technology:** 213 articles

2.2 Data Cleaning and Preprocessing

The dataset was first cleaned to ensure its quality and consistency. Missing values and duplicate rows were removed, and the text data was standardized. Specifically, the content was preprocessed by applying tokenization, stop-word removal, and stemming. The clean_text function was used to remove non-Urdu characters, retaining only the Urdu text in the articles.

Following this, the articles were vectorized using a custom BagOfWords class. The fit method of the class creates a vocabulary by identifying unique words across all documents in our Train set, while the vectorize method converts each article into a numeric vector based on the word frequency in the vocabulary. This transformation results in a structured representation of the text data, ready for further analysis.

Additionally, the cleaned articles were saved to a CSV file, removing the need for repeated cleaning for every model.

2.3 Models Implemented

Three models were employed for classification:

(1) **Naive Bayes:** A probabilistic approach that assumes feature independence.
(2) **Logistic Regression:** A linear model for multi-class classification optimized using gradient descent.
(3) **Neural Networks:** A feedforward model with one hidden layer to capture non-linear patterns.

2.4 Model 1 : Multinomial Naive Bayes

2.4.1 Model Architecture

For the classification of Urdu news articles into five categories, we used the Multinomial Naive Bayes (MNB) algorithm. The architecture of the model relies on a probabilistic approach where each word in a document is treated as a feature, and the likelihood of its occurrence is computed for each class. The steps of the process include:

- **Input:** The feature matrix, obtained using the Bag-of-Words (BoW) model, where each document is represented by the frequency of words.
- **Model:** The model computes the conditional probability of each word occurring in each class, using Laplace smoothing to avoid zero probabilities for unseen words.
- **Output:** The model outputs the predicted class based on the highest posterior probability, calculated using Bayes' Theorem.

Naive Bayes was chosen for its efficiency and suitability for text classification, as it assumes feature independence, aligning well with the Bag-of-Words representation. Its computational simplicity makes it a strong baseline for comparing more complex models.

2.4.2 Model Training

During the training phase, the following steps were carried out:

- The **prior probabilities** for each class were estimated based on their frequency in the training data.
- The **conditional probabilities** of each word given each class were calculated using maximum likelihood estimation with Laplace smoothing, ensuring that unseen words had a non-zero probability.

2.4.3 Loss Function and Optimization

For MNB, the objective function is the log-likelihood of the data given the class labels. The model was optimized by directly maximizing the likelihood through parameter estimation without the need for a separate optimizer as in gradient-based methods.

2.4.4 Evaluation

The performance of the model was evaluated using accuracy, macro-average F1 score, and a confusion matrix. The test results showed:

- **Test Accuracy:** 96.44%
- **Macro-Average F1-Score:** 96.54%

2.4.5 Results

The confusion matrix showed that the model performed exceptionally well for most categories, especially *sports*, where all predictions were correct. However, some misclassifications occurred between *business* and *science-technology*, and between *world* and *entertainment*. This could be attributed to vocabulary overlap between these categories.

The accuracy and F1 scores indicate the model's ability to generalize well across all classes. While the model is computationally efficient and performs well, future improvements may involve exploring advanced feature representations like TF-IDF or word embeddings to address the issue of overlapping vocabulary between certain categories.

Additionally, a confusion matrix was generated to analyze the classifier's performance across different categories:

$$\begin{bmatrix} 71 & 0 & 1 & 0 & 1 \\ 0 & 70 & 1 & 0 & 2 \\ 1 & 0 & 66 & 0 & 1 \\ 0 & 0 & 0 & 56 & 0 \\ 0 & 3 & 2 & 0 & 62 \end{bmatrix}$$

The confusion matrix shows strong overall performance with high correct classification rates (diagonal values) and minimal misclassifications, indicating balanced accuracy across all classes.

2.5 Model 2 : Logistic Regression

2.5.1 Model Architecture

To classify Urdu news articles into five categories, we implemented a One-vs-All Logistic Regression model. The architecture and process involve:

- **Input Representation:** Features are derived from the Bag of Words model.
- **Classifier:** Five binary logistic regression models are trained, each distinguishing one class from the others.
- **Loss Function:** Each model minimizes the `CrossEntropyLoss`, suitable for binary classification, by iteratively updating weights and biases using gradient descent.

Logistic Regression was selected for its simplicity and ability to handle multiclass problems effectively. It works well for tasks requiring linear decision boundaries and scales efficiently with large datasets, providing a reliable and interpretable baseline.

2.5.2 Optimization

Each classifier uses gradient descent with:

- Learning rate: **0.1**.
- Regularization term (*L2* penalty): **0.01**.
- Number of iterations: **200 epochs**.

This configuration ensures convergence with minimal overfitting.

2.5.3 Multiclass Prediction

For each test sample, we compare the probabilities generated by the five classifiers. The class with the highest probability is selected

as the final prediction.

2.5.4 Evaluation

The model's performance was evaluated on the test set using multiple metrics:

- **Overall Test Accuracy:** 96.44%
- **Macro Average F1-Score:** 96.49%

2.5.5 Results

Training loss curves for each binary classifier showed steady convergence within 200 epochs. The multiclass accuracy and F1-scores indicate that the model generalizes well to unseen data, with no significant imbalance in performance across classes.
The confusion matrix provided detailed insights into the model's predictions:

$$\begin{bmatrix} 67 & 0 & 1 & 1 & 0 \\ 0 & 61 & 1 & 0 & 1 \\ 0 & 0 & 66 & 0 & 1 \\ 2 & 0 & 1 & 63 & 0 \\ 0 & 0 & 4 & 0 & 68 \end{bmatrix}$$

The confusion matrix demonstrates effective classification, with most predictions aligning correctly with the true classes (diagonal values) and only a few misclassifications, reflecting consistent performance across all categories.

2.6 Model 3 : Neural Network

2.6.1 Neural Network Architecture

To classify Urdu news articles into five categories, we implemented a feedforward neural network using PyTorch. The architecture consists of:

- **Input Layer:** Dimensions equal to the feature size from our Bag of Words model.
- **Hidden Layers:**
  (1) A fully connected layer with 128 neurons and **sigmoid activation**.
  (2) A fully connected layer with 64 neurons and **LeakyReLU activation**.
- **Output Layer:** A fully connected layer with five neurons corresponding to the classes, using the **sigmoid function** for multi-class classification.

Neural Networks were employed to capture non-linear patterns that simpler models might miss. With a hidden layer, they offer flexibility and higher predictive power, making them ideal for learning complex relationships in text data.

2.6.2 Loss Function and Optimzation

We used the `CrossEntropyLoss` function, which is suitable for multi-class classification problems. It calculates the difference between predicted and true probabilities for the five output classes.

The **Adam optimizer** with a learning rate of 0.001 was used for training the model. This optimizer helps in faster convergence and adjusts learning rates dynamically.

2.6.3 Evaluation

We split the data into Train, Validation and Test sets in a ratio of 70:15:15
The model's performance was evaluated on the validation set using accuracy as the primary metric. Additionally, a classification report provided precision, recall, and F1-scores for each class. The test set results showed:

- **Test Accuracy:** 97.04%
- **F1-Score (macro average):** 97%

2.6.4 Results

The training and validation loss curves indicate that the model converged effectively within 30 epochs. Accuracy curves show consistent improvement, with minimal overfitting.
Additionally, a confusion matrix was generated to analyze the classifier's performance across different categories:

$$\begin{bmatrix} 34 & 0 & 0 & 0 & 0 \\ 2 & 31 & 0 & 0 & 0 \\ 0 & 0 & 34 & 1 & 0 \\ 0 & 0 & 0 & 35 & 0 \\ 0 & 1 & 1 & 0 & 30 \end{bmatrix}$$

The matrix indicates the model's strong ability to generalize across most categories, with minimal errors and high precision in its predictions.

2.7 Tools and Libraries

The following tools and libraries were utilized:

- **numpy:** For implementing Naive Bayes and Logistic Regression. Numpy was the only library used for these two models.
- **pytorch:** For building and training the Neural Network.
- **regex:** For preprocessing Urdu text.
- **pandas:** For dataset handling and management.
- **matplotlib and seaborn** : For plotting
- **sklearn**: For test train split only.
- **kagglehub**: For Urdu stopwords

3 FINDINGS

The hyperparameters of all the models (i.e., learning rate, number of iterations, layers of the Neural Network, and activation functions) were tuned repeatedly until the best accuracy and Macro F1 score for each model were achieved. Some key findings are summarized below:

3.1 Model Performance

The accuracies of the models are as follows:

- **Multinomial Naive Bayes:**
  - Accuracy = 96.44%
  - Macro-Average F1 Score = 96.54%
- **Logistic Regression:**
  - Accuracy = 96.44%

  - Macro-Average F1 Score = 96.49%
- **Neural Networks:**
  - Test Accuracy = 97.04%
  - Macro-Average F1 Score = 97%

3.2 Misclassification of "World" Articles

Articles with the label "world" were the most misclassified among all the models. This was largely due to its frequent overlap with the "sports" and "entertainment" categories. Many articles with "world" as the category often discussed sports events, celebrity news, or international entertainment. This confusion led to errors in categorizing these articles, highlighting a common issue faced by all models in distinguishing between closely related categories.

3.3 Neural Networks' Superior Performance

Among all the models, Neural Networks achieved the highest test accuracy (97.04%) and Macro F1 score. The reason for NN's superior performance lies in its ability to capture complex non-linear relationships in the data, which provides a more nuanced understanding of the features and their interactions. The neural network's architecture, especially with its layers and activation functions, enabled it to effectively learn and generalize from the dataset.

3.4 Model Comparison and General Performance

Despite Neural Networks outperforming the other models, all the models (MNB, Logistic Regression, and NN) performed exceptionally well, providing high accuracy and consistent performance across all labels. Each model showed its strengths in handling different types of data, with MNB and Logistic Regression providing reliable results in linear separability and simple feature relationships. In contrast, Neural Networks excelled at capturing more complex patterns in the data, making it the best performing model overall.

These results demonstrate that even simpler models such as MNB and Logistic Regression can achieve high accuracy and F1 scores, though the deeper non-linear models, like Neural Networks, tend to edge out in terms of overall performance when more complex patterns are present in the data. All models showed effectiveness in predicting labels, and each has its place in applications depending on the specific requirements of model interpretability, complexity, and performance.

4 LIMITATIONS AND CONCLUSION

4.1 Limitations

One of the main limitations of this study is the relatively small sample size. We only scraped around 1100 articles from three websites, which could potentially limit the generalizability of our findings. The small dataset size might lead to overfitting, especially as the models might have adapted too closely to the specific data patterns from these particular websites. This could cause a lack of robustness when applied to articles from other sources with different writing styles or content.

Additionally, our models were trained on Urdu articles, and the complexity of the Urdu language posed its own set of challenges. Multiple Urdu words can have different meanings depending on the context, making it difficult for the models to consistently predict the correct label. This issue is particularly prominent in languages like Urdu, where words may change their meaning based on syntactic or semantic contexts. The linguistic intricacies of Urdu, such as word variations and the use of homonyms, added further complexity to the classification task.

Another limitation is the stopwords library we used for preprocessing. While it was useful for some basic text cleaning, it was not exhaustive and may have missed important stopwords or failed to address certain variations in the language. This could have impacted the preprocessing of data and, in turn, the performance of the models, as the effectiveness of stopword removal is crucial for improving the signal-to-noise ratio in text data.

4.2 Conclusion

In this study, we successfully implemented and compared three different machine learning models for text classification:

- **Multinomial Naive Bayes**
- **Logistic Regression**
- **Neural Networks**

All three models performed exceptionally well in terms of accuracy and Macro F1 score, with Neural Networks outperforming the others due to its ability to capture complex non-linear relationships in the data. Despite the challenges posed by a small sample size, linguistic intricacies of the Urdu language, and the limitations of the stopwords library, we successfully implemented the classification task. The results demonstrate that even simpler models such as MNB and Logistic Regression can be effective in text classification tasks, while more complex models like Neural Networks tend to offer higher performance when dealing with intricate data patterns.

Overall, this work illustrates the potential of such machine learning models in handling Urdu text classification and provides insights into the challenges faced when working with languages that have rich semantic contexts. Future work could involve expanding the dataset, improving the preprocessing steps, and exploring more sophisticated approaches to handle the linguistic complexity of Urdu.

5 ACKNOWLEDGMENTS

We would like to thank our assigned TA, Shaheer Humayun and Warda Jamil for there timely support and guidance throughout the project. It wouldn't have been possible without them!