Classification of Urdu News Articles: A Machine Learning Framework for Language-Specific Content Categorization

Taha Faisal Khan
26100076@lums.edu.pk

Haider Dawood Khan
26100053@lums.edu.pk

Muhammad Saram Hassan
26100197@lums.edu.pk

Shahreyar Ashraf
26100342@lums.edu.pk

Syeda Zahra Ali Tirmizy
26100147@lums.edu.pk

Abstract

This paper explores the underrepresentation of low-resource languages in natural language processing (NLP) by focusing on the classification of Urdu news articles into categories such as entertainment, business, and sports. To address the scarcity of publicly available Urdu datasets, a custom dataset was developed, sourcing articles from leading Urdu news platforms. Comprehensive preprocessing techniques, including normalization and stopword removal, were employed to refine the dataset, which was subsequently vectorized using the Bag-of-Words approach. The study implemented three machine learning models—Multinomial Naive Bayes, Logistic Regression, and a Neural Network—to evaluate their effectiveness in this task. Among these, the Neural Network achieved the highest accuracy (97.94%) on the test set, surpassing Logistic Regression (97.25%) and Naive Bayes (94.74%) by capturing complex, non-linear patterns in the data. These findings underscore the efficacy of machine learning in overcoming the linguistic and technical challenges inherent in low-resource languages like Urdu, highlighting its potential for broader applications in multilingual NLP.

Keywords

Urdu news classification, machine learning, natural language processing, content categorization, underserved languages, data scarcity, web scraping, feature engineering, multiclass classification, exploratory data analysis, data preprocessing, supervised learning

ACM Reference Format:
Taha Faisal Khan, Haider Dawood Khan, Muhammad Saram Hassan, Shahreyar Ashraf, and Syeda Zahra Ali Tirmizy. 2024. Classification of Urdu News Articles: A Machine Learning Framework for Language-Specific Content Categorization. In *Proceedings of (Machine Learning Fall).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The rapid advancements in machine learning and natural language processing have revolutionized how we interact with information. Large language models have become an indispensable tool in organizing, analyzing, and retrieving data with remarkable accuracy and scale. These benefits have mostly been accrued to high-resource languages such as English, which dominate NLP research due to the availability of extensive corpora. On the other hand, low-resource languages like Urdu, despite being spoken by over a 100 million people worldwide [1], are sparsely represented, followed by underutilization with regards to transformative NLP applications. This disparity deprives Urdu-speaking users of tools that could drive progress in crucial areas like education, agriculture, and communication.

A prominent gap exists in limited access to news content in an organized and personalized manner in the case of Urdu. To address this, we developed a machine learning-based system to classify Urdu news articles into predefined categories, such as entertainment, business, and sports. This can allow efficient hierarchal structured representation of data, which can aid in personalized recommendations to enhance user experience through machine learning. However, the major challenge was the scarcity of publicly available Urdu datasets, which urged us to create a custom dataset from leading Urdu news outlets like Geo Urdu, Jang, and ARY News.

Our research implemented and evaluated three machine learning models, each from scratch, namely Naïve Bayes, Logistic Regression, and a Neural Network. This helps to improve the state of NLP for low-resource languages and goes a little further in closing the gaps within linguistic divides, paving a path for inclusive technology. This project not only enhances the quality of Urdu news classification for improving recommendations but provides an overall approach toward empowerment for under-representative languages in a broader research scope of NLP.

Machine Learning Fall, 2024,

ACM ISBN 978-1-4503-XXXX-X/24/06
https://doi.org/XXXXXXX.XXXXXXX

2 Methodology

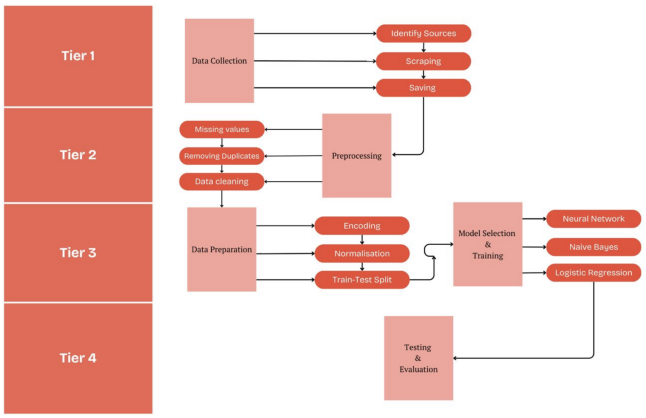

Figure 1: Data Processing and Model Training Pipeline

Figure 1 outlines the multi-tiered pipeline utilized in this study. It encompasses the stages of data collection, preprocessing, preparation, model selection, and evaluation. Each stage plays a critical role in ensuring the reliability and robustness of the models developed, as detailed in the subsequent sections.

2.1 Web Scraping

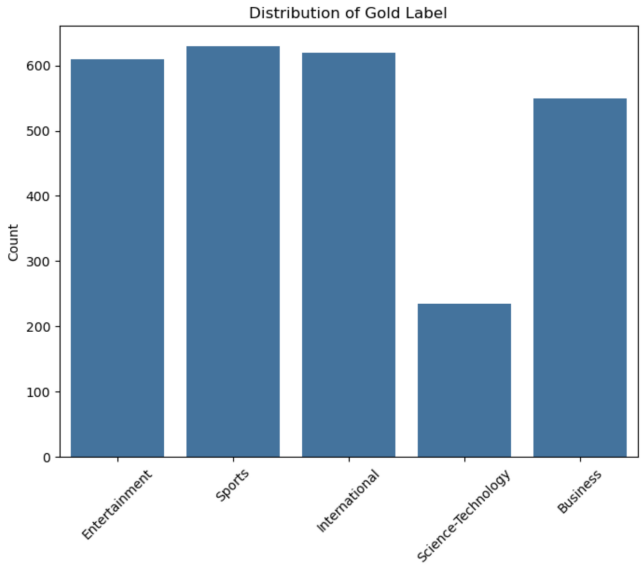

Figure 2: Gold Label Distribution

In response to the scarcity of publicly available Urdu datasets, we devised a web scraping pipeline to compile news articles from prominent Urdu media outlets, including Geo Urdu, Jang, ARY, Dawn, and Dunya News. Utilizing Python's requests library, HTTP GET requests were executed to retrieve web pages, while BeautifulSoup was used to parse the HTML structure and extract key information such as headlines, article content, and publication dates. To manage dynamic content and mitigate the risk of blocking, the pipeline incorporated randomized headers and strategic delays between requests, effectively simulating human browsing behavior. For each category, the depth of the search was limited to a maximum of 14 pages due to time and resource constraints. Moreover, the news channel were not scraped across different time-points but rather sequentially in one concurrent run. The result was a robust dataset of 2185 Urdu articles, encompassing the following categories: entertainment, sports, international, science-technology, and business. This dataset was then used for training machine learning models. Figure 2 shows the gold label distribution across the categories after scraping all 5 websites.

2.2 Dataset and Preprocessing

The dataset consisted of Urdu text samples labeled into predefined categories. A consistent preprocessing pipeline was applied across all models to ensure high data quality and effective feature representation:

- **Character Normalization:** Addressed Urdu linguistic constructs such as "hamza" and "alif" variants to standardize textual inputs.
- **Tokenization and Lemmatization:** Tokenized text into words and reduced them to their base forms, aiding generalization across morphological variations.
- **Stopword Removal:** Used a custom list to eliminate non-informative words, focusing on meaningful tokens.

For all models, numerical feature representation was achieved via the bag of words approach to convert the entirety of the text into a vectorized form. The training data was used to create the initial word-to-index frequency mapping, which was then used to convert the each article into a single vector. There were about 12000 unique words extracted from the training set after the initial preprocessing, which formed the basis of the bag of words. A simple 80-20 train-test split was used for the training and evaluation of the Multinomial Naive Bayes and Neural Network. The split was slightly altered into training (60%), validation (20%), and test (20%) subsets for Logistic Regression to perform hyper-parameter tuning. These steps rigorously prepared the dataset to be passed as input into the machine learning models implemented.

3 Model Implementation and Testing

Three machine learning models were implemented from scratch to classify Urdu news articles: Multinomial Naive Bayes (MNB), Logistic Regression, and a Neural Network. Each model was selected based on its suitability for text classification and ability to handle high-dimensional, sparse datasets.

3.1 Multinomial Naive Bayes (MNB)

The Multinomial Naive Bayes (MNB) model was chosen for its efficiency and effectiveness in text classification, particularly for high-dimensional and sparse datasets. Its probabilistic nature and assumption of feature independence make it well-suited for Bag-of-Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) representations.

The MNB model was implemented from scratch to deepen our understanding of its underlying mechanics. Preprocessed text data was transformed into a BoW representation, capturing word frequencies for each document. Class priors were calculated from the training set, and likelihoods for each word in each class were estimated using maximum likelihood estimation with Laplace smoothing to handle unseen terms. Predictions were made by calculating posterior probabilities for each class using Bayes' theorem, selecting the class with the highest probability. To avoid numerical underflow, logarithmic transformations were applied to probabilities. The model's efficiency ensured robust performance even with the large feature space.

3.2 Logistic Regression

Logistic Regression was selected as a baseline model for its simplicity, interpretability, and effectiveness in both binary and multi-class classification tasks. It is particularly well-suited for data with linear relationships between features and class labels.

Logistic Regression was implemented from scratch. However, this was regularized with the L2 norm to preventing over-fitting and improve generalization. The validation set was used to test the model across a range of values of lambda (the regularization parameter) and the one yielding the highest accuracy on the validation set was selected. The preprocessed text data was transformed into a BoW representation. A sigmoid function was applied to map linear combinations of features to probabilities between 0 and 1. Weights were initialized randomly and updated iteratively using gradient descent to minimize binary cross-entropy loss. A learning rate controlled the magnitude of updates during optimization to ensure convergence. For multi-class classification, a one-vs-rest approach was adopted, training separate classifiers for each class. Once the optimal value was found, the entire train dataset was used to train the regularized logistic regression classifier. Figure 3 shows the training loss of each of the one-vs-rest classifiers on the optimal lambda value across gradient descent iterations.

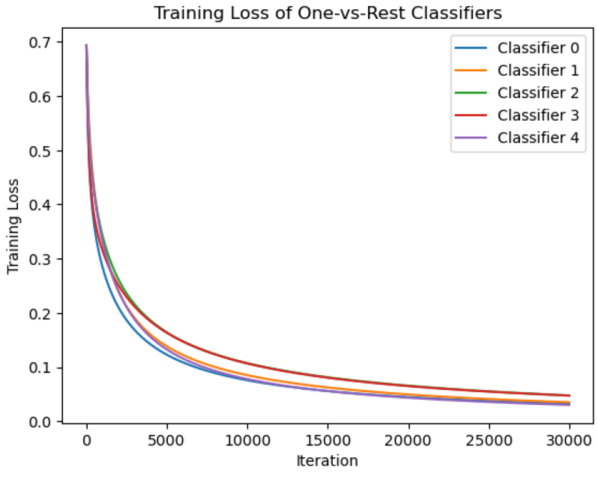

Figure 3: Training Loss of One-Vs-Rest Classifiers for optimal Lambda

3.3 Neural Network

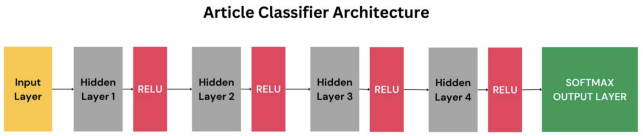

Figure 4: Article Classifier Architecture

A Neural Network was chosen to capture potential non-linear relationships in the data, which simpler models like Logistic Regression and Naive Bayes might not adequately represent. Its layered architecture allows for learning complex patterns, making it ideal for high-dimensional and sparse text features.

The Neural Network was implemented using PyTorch, with an architecture comprising:

- **Input Layer:** The initial vocabulary vector was shortened to the top 6000 most frequent words due to computational constraints. However, this also served to prevent over-fitting and unnecessary computation since the vocabulary consisted of niche words (either misspelled or of different lexicon) that only occurred once in thousands of samples.
- **Hidden Layers:** Four fully connected layers with Rectified Linear Unit (ReLU) activations to model non-linear relationships.
- **Dropout Regularization:** Applied during training to prevent the model from over-fitting and learning noise with probability=0.4.
- **Output Layer:** A softmax activation to produce multi-class probability predictions of the 5 classes. The argmax was taken and that label was assigned to the particular input.

Figure 4 illustrates this architecture. The model was trained with the Adam optimizer and cross-entropy loss over 50 epochs, using PyTorch's Dataloader for efficient batching. Figure 5 shows the Training and validation accuracies that were monitored to ensure effective learning and detect overfitting.

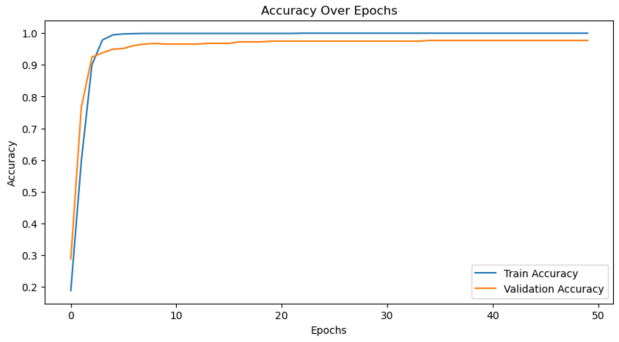

Figure 5: Accuracy over Epochs on Train and Validation Set

3.4 Model Comparison

Model	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Logistic Regression	97.25	97.34	97.39	97.30
Naive Bayes	94.28	93.98	93.88	94.16
Neural Network	98.86	98.91	98.82	99.00

Figure 6: Performance Comparison

The performance of the models was compared using evaluation metrics, which are summarized in Figure 6. The Neural Network achieved the highest accuracy (98.86%), outperforming Logistic Regression (97.25%) and Naive Bayes (94.28%). It achieved the highest macro f1-score as well, highlighting its effectiveness. The Neural Network's ability to capture non-linear relationships contributed to its superior performance, while Logistic Regression performed well due to the linear nature of some feature-class relationships. Naive Bayes, while efficient, was limited by its assumption of feature independence, which did not align with the dependencies observed in the dataset. The superiority of the neural network on the test set was not only found in the accuracy metric but across all metrics, highlighting its robustness and impressive performance in unseen scenarios.

4 Conclusion

Using the evaluation metrics mentioned above, we determined that that the Neural Network outperformed Logistic Regression and Naive Bayes. Achieving an accuracy of 98.86% highlights the Neural Network's ability to capture complex, non-linear relationships in the dataset, which the simpler models struggled to model effectively. Its four layered architecture enabled it to learn nuanced patterns in the high-dimensional, sparse feature space. Moreover, the application of dropout regularization helped prevent overfitting, ensuring robust generalization to unseen data. The high recall of 98.09% further demonstrates its strength in minimizing false negatives.

The Neural Network's advantage over Logistic Regression lies in its ability to handle non-linear decision boundaries, while Logistic Regression, as a linear model, is limited to capturing linear relationships between features and class labels. Logistic Regression performed well, achieving 97.25% accuracy, partly due to the linear nature of some relationships in the dataset and the use of L2 regularization, which enhanced its generalization. However, it could not fully leverage the dataset's complexities, especially when interactions between features were non-linear.

Naive Bayes, although efficient and effective for text classification in general, performed the weakest with 94.28% accuracy. This is primarily due to its core assumption of feature independence, which is often unrealistic for text data where contextual relationships between words are common. While the use of Laplace smoothing mitigated issues with unseen terms, the simplifying assumptions of the model hindered its ability to fully capture the dependencies present in the dataset.

4.1 Limitations

Despite the promising results achieved in this study, there are several limitations to consider. The primary limitation is the scarcity of publicly available Urdu datasets, which constrains the diversity and representativeness of the training data. This lack of data makes it challenging for models to generalize effectively to unseen text, particularly for niche topics or domains. The dataset curated for this project had its own limitations as well. Mainly, its limited size and lack of articles across more categories and especially a larger time frame limited the effectiveness of our machine learning models. This compromises the models on unseen text, especially because scraped data was localized to a particular event, making it ineffective against different topics and contexts. Thus, more time could have been spent curating a more rigorous dataset that spanned across more categories, news channels, and a larger time frame. Additionally, the reliance on traditional feature representations such as Bag-of-Words, while effective, may not capture the full contextual nuances of Urdu text. Each word was associated with an index, which simply maintained its frequency. Urdu, being a rich and complex language and vast vocabulary, makes it difficult for models like Bag-of-Words (BoW) to capture the full spectrum of words used in the language. This results in significant limitations in the feature representation, as the dataset's vocabulary is unlikely to cover all possible words and their variations. Additionally, Urdu allows for multiple ways of writing the same word, often varying in punctuation marks. These differences, while minor to a human reader, can create inconsistencies in the dataset and hinder the model's ability to generalize effectively. Another significant issue is the lack of standardized Unicode representations for Urdu alphabets. This creates discrepancies where the same word might be written using Persian, Arabic, or Urdu-specific script conventions, resulting in different character representations. Such inconsistencies not only alter the word recognized by the system but can also change its semantic meaning, further complicating text processing tasks. Addressing these issues would require careful preprocessing, standardization efforts. The information that the bag of word could capture in our models was very limited and an approach relying on word embeddings and relating them to each other would've been significantly more effective.

Another limitation lies in the computational requirements of the Neural Network, which, despite its superior performance, demands significantly more resources than simpler models like Logistic Regression and Naive Bayes. Furthermore, the models were evaluated on a fixed set of categories, potentially limiting their applicability to dynamic or overlapping categories in real-world scenarios. Addressing these limitations would be crucial for further improving the performance and applicability of such classification systems.

4.2 Future Work

Implementing a category classification model for Urdu news articles presents unique challenges and opportunities for future exploration. To enhance classification accuracy further, we must address current limitations. One of the main limitations that should be addressed is the lack of Urdu language datasets. Access to diverse and representative datasets is necessary for the model to learn the nuances of the language and, consequently, improve its generalization. Future work should focus on augmenting datasets that are diverse and inclusive of a wide range of Urdu text domains. Using techniques such as web scraping or leveraging transfer learning with pre-trained models that incorporate labeled data has the potential to overcome the dataset limitation, enabling better model training and generalization.

Another potential direction for future work is experimenting with encoder-decoder architectures. These models are particularly well-suited for handling long or complex text sequences, such as news articles, by capturing both global and local contextual nuances. By integrating attention mechanisms, the model can focus on the most relevant parts of the text, improving its understanding of the content and enhancing classification accuracy. Fine-tuning pre-trained multilingual models like mBERT, mT5, or MarianMT for Urdu could further optimize performance, as these models are already trained on diverse, multilingual datasets and can leverage their existing knowledge to better adapt to Urdu text.

Improving model interpretability is another critical avenue for exploration. Incorporating explainability techniques such as SHAP or attention heatmaps can help identify the contributions of specific words or phrases in the classification process. This would build trust in the model's classification decisions and identify biases and areas for improvement.

Acknowledgments

We thank our supervisor, Dr. Agha Ali Raza and our TAs, for guidance and support.

References

[1] Urdu speaking countries. (July 2024). Retrieved December 7, 2024 from https://www.worlddata.info/languages/urdu.php.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30, pp.5998-6008.