

UMT Artificial Intelligence Review (UMT-AIR)

Volume 3 Issue 2, Fall 2023


ISSN(P): 2791-1276, ISSN(E): 2791-1268

Homepage: <https://journals.umt.edu.pk/index.php/UMT-AIR>



Article QR



- Title:** **Determining Urdu News Type from Headline Text Using Deep Learning**
- Author (s):** Umair Arshad¹, Khawar Iqbal Malik², Hira Arooj³, Muhammad Fiaz⁴
- Affiliation (s):** ¹Robert Gordon University, Aberdeen, United Kingdom
²Riphah International University, Lahore Campus, Pakistan
³Department of Mathematics and Statistics, University of Lahore, Sargodha Campus, Pakistan
⁴Department of Computer Science, University of Lahore, Sargodha Campus, Pakistan
- DOI:** <https://doi.org/10.32350/umt-air32.05>
- History:** Received: August 2, 2023, Revised: October 26, 2023, Accepted: November 8, 2023, Published: December 2, 2023
- Citation:** U. Arshad, K. I. Malik, H. Arooj, and M. Fiaz, "Determining Urdu news type from headline text using deep learning," *UMT Artif. Intell. Rev.*, vol. 3, no. 2, pp. 00–00, Dec. 2023, doi: <https://doi.org/10.32350/umt-air.32.05>
- Copyright:** © The Authors
- Licensing:**  This article is open access and is distributed under the terms of [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)
- Conflict of Interest:** Author(s) declared no conflict of interest



A publication of

Department of Information System, Dr. Hasan Murad School of Management
University of Management and Technology, Lahore, Pakistan

Determining Urdu News Type from Headline Text Using Deep Learning

Umair Arshad¹, Khawar Iqbal Malik^{2*}, Hira Arooj³, and Muhammad Fiaz⁴

¹School of Computing, Robert Gordon University, Aberdeen, United Kingdom

²Riphah School of Computing & Innovation, Riphah International University, Lahore Campus, Pakistan

³Department of Mathematics and Statistics, University of Lahore, Sargodha Campus, Pakistan

⁴Department of Computer Science, University of Lahore, Sargodha Campus, Pakistan

ABSTRACT In recent years, the volume of data of regional languages available on the Internet has grown significantly. It helps people to express themselves by removing linguistic boundaries. Moreover, the accessibility of news articles on the web provides billions of web users with a source of knowledge. This research offers a classification model for categorizing Urdu news headlines text with deep learning (DL) techniques and different word vector embeddings. To improve the efficacy of various Urdu natural language processing (NLP) applications, this study included two neural word embeddings built by utilizing the most widely used approaches, namely Word2vec and pre-trained fastText. Both intrinsic and extrinsic evaluation methods were used to examine the integrity of the created neural word embeddings. The study employed a vast, fresh corpus of Urdu text containing 153,050 headlines categorized into 8 different classes. Then, text pre-processing techniques and two DL models, namely the Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) were applied. The results were compared based on embeddings. It was found that when a pre-trained fastText embedding was utilized, BiLSTM surpassed other DL models with an accuracy of 93.93%, precision of 93.86%, recall of 93.93%, and F1 score of 93.89%.

INDEX TERMS deep learning (DL), fastText, natural language processing (NLP), Urdu news classification, Word2vec

I. INTRODUCTION

Today, a substantial section of the population gets their news via technological gadgets. People may now access accumulated knowledge thanks to the vast volume of textual content available on the web in various spoken languages. Many news organizations have apps for phones and tablets that display their content in a continuous stream, often sorted by publication date. Every newspaper article

does not pique the interest of every reader. Many newspapers divide their articles into categories, allowing readers to see all articles about the world, entertainment, or science listed together. The challenge is learning about different categories; users have browsed multiple pages. To address this challenge, text classification was utilized in the current study to teach a computer to choose exciting articles based on a user's prior experience. A newspaper can utilize this information to produce a

*Corresponding Author: khawar.iqbal@riphah.edu.pk

flow of personalized stories for each user. The process of categorizing documents according to their types is known as text classification. It employs many skills and technologies including artificial intelligence (AI), natural language processing (NLP), and machine learning (ML). It also employs a supervised learning approach in which a model is trained using a vast data. Urdu is Pakistan's official language. It is also spoken in Bangladesh, India (South Asia), the United Kingdom (Europe), and Canada (North America). It is also spoken in many other countries, such as the United Arab Emirates (UAE) and the United States [1]. There are over 12 million native speakers of Urdu, with an estimated global population of 300 million speakers or more [2]. Still, it is considered to be a language that is resource poor. Arabic and Persian have heavily influenced Urdu's lexicon. Despite its widespread use, Urdu is deficient in linguistic resources. Despite the abundance of dictionaries, WordNet-like semantic ideas are still scarce.

The current study would substantially help the society, particularly news bloggers, media organizations, and news followers. Firstly, effective news classification would allow people to discover news kinds more correctly and quickly, mainly resolving the mess of news information. News categorization would help individuals to find news types more accurately and rapidly, primarily by resolving the chaos of news data. Much labor, time, and efforts are required to classify and analyze documents containing long texts. Short texts, such as tweets and reviews, have fewer words, belong to the same topic semantically, and are less noisy. On the other hand, large paragraphs of Urdu news have more feature redundancy, a more extensive vocabulary, more noise, and contain multiple topics, when compared with short and automatic

text classification. Secondly, text mining or text classification is a complex undertaking requiring numerous pre-processing procedures to convert unstructured data into structured data. The research problem entails automatically identifying Urdu news so that it can be displayed in the proper news part of the various news blogs for online users.

Most of the research on news text classification has been conducted with reference to English. There is considerable interest in this subject. The categorization of news has, therefore, begun in several languages including Indonesian, Arabic, and Chinese. Unfortunately, the Urdu vocabulary has a huge deficit. However, several scholars have found numerous methods for sentiment analysis in Urdu. Many ML approaches including Support Vector Machine (SVM), Decision Tree (DT), and naive Bayes were used in most previous studies. Nonetheless, automatic categorization has grown increasingly complex due to the growing size of corpora and the difficulty of dividing a document into fields. As a result, these ML algorithms are less accurate. Nevertheless, most previous studies used ML approaches as the proposed method relies on deep neural networks.

Recently, neural networks have performed efficiently in the field of text categorization. To overcome the limitations of ML models, a Deep Learning (DL) model is proposed based on LSTM and BiLSTM to fill the existing gap in text classification for Urdu. This is essential to achieve better results in the classification of Urdu news. In order to improve the efficacy of various Urdu NLP applications, this study includes two neural word embeddings built by utilizing two broadly used methods, specifically self-trained Word2vec and pre-trained fastText.

Furthermore, intrinsic and extrinsic evaluation methods were used to examine the integrity of the created neural word embeddings. The study used a large and current corpus of Urdu headlines (153,050) from 8 classes including science and technology, the world, entertainment, business and economics, sports, politics, weird, and health. Afterwards, text pre-processing techniques were used.

Four sorts of experiments were conducted during the course of this study. Firstly, LSTM was used through word2Vec embedding for transforming words in vector and building the vocabulary. Secondly, LSTM was used with pre-trained fastText embeddings. Similarly, in the third phase, the BiLSTM model, which employs self-trained Word2vec embeddings, was employed. Finally, it was followed by BiLSTM with pre-trained fastText embeddings. After applying DL algorithms, the findings were compared and the performance was evaluated using various factors. It was found that BiLSTM and LSTM, through pre-trained fastText classifiers, have a substantial impact on others and provide more effective findings.

The rest of the paper is organized as follows. Section II discusses the specifics of the related work of various scholars on text classification. Section III discusses the methods and processes of execution. Section IV explains the experimental setup of the current work. Section V contains the results and discussion. Finally, Section VI states the summary, conclusion, and recommendations for future work.

II. Related Work

This section discusses the current research on categorizing Urdu news text as well as approaches used in adjacent domains that influence this subject. Indeed, previous researchers have developed several

strategies for improving the quality of categorization results which are addressed below.

Traditional text categorization utilizes feature engineering to obtain useful features that aid model performance. Text classification is a tried-and-true method of organizing a vast number of documents. Many text categorization techniques have been proposed [3], including SVM, KNN, naive Bayes, and decision tree. The analyses revealed that these algorithms are exceptionally efficient in traditional text categorization applications. Rasheed et al. proposed latent semantic indexing (LSI) classification of Urdu text using a hybrid feature selection approach. They employed the Urdu "ROSHNI" dataset [4]. The corpus contained 29,931 articles of sixteen categories. The study used filter selection methods and LSI to extract essential components from Urdu documents. The results suggested that the hybrid approach's SVM classification is more accurate and efficient. Similarly, Rasheed et al. applied ML techniques to classify Urdu texts. They employed 16,678 Urdu news documents, mainly from The Daily Roshni. The authors utilized three different classifiers [5]. WEKA application was used to examine the classification of Urdu text using decision tree, SVM, and KNN. The SVM classifier fared much better with a high accuracy and efficiency. In contrast, Tej Bahadur Shahi used naive Bayes, SVM, and neural networks to classify Nepali news. The corpus contained 20 different classes and a total of 4964 documents. With a classification accuracy of 74.65%, SVM with RBF kernel performed better than the additional three methods [6]. Arshad et al. [7] discussed the problem of Urdu news content classification. The authors compared the results obtained from 12 different ML classifiers of Urdu news texts.

The study used a recent collection of Urdu literature with approximately 0.15 million classified occurrences of 8 different classes. With an accuracy of 91.37%, SVM surpassed the other 11 methods after assessing for various ML approaches.

NLP makes use of DL tasks. For example, Nabeel et al. [8] used DL classification approaches to classify Urdu text. They used deep neural models for Arabic text classification [9]. Similarly, for Chinese text classification, the researchers used an attention method [10]. Nonetheless, text classification has become quite tricky in recent years. With the expansion of corpus sizes and document categories, it is critical to organize the document not only by a specific field but also by its broad category and subclass.

Miguel Molina-Solana et al. [11] showed how DL may be used to categorize tweets as containing or not containing fake news during NLP. Furthermore, they showed that DL algorithms can detect false news in tweets. Moreover, LSTMs with pre-computed embeddings beat the other tactics (validation AUC = 0.70), especially when avoiding the misclassification of the minority class. This research [12] proposed a novel method for text classification on multi sentiments on a large scale. Firstly, the text was translated into a graph. Then, an attention method was employed to depict the text's complete functionality. They also employed convolutional neural networks (CNNs) to extract features. When this method is compared with state-of-the-art approaches, the proposed method produces good results.

Several scholars have used DL in text categorization due to its advance features. For example, Li et al. used the BiLSTM-CNN technique to study the use of NLP in text categorization [13]. Their corpus

contained a total of 65,000 corpora. When they combined Bi-LSTM and CNN, classification performance improved by 0.84%. Ramdhani et al. used the CNN algorithm for Indonesian news text classification. CNN achieved a maximum accuracy rate of 90.74% [14]. Similarly, Somya Ranjan Sahoo and B. B. Gupta suggested an ML and DL classifier-based false news detection solution for Facebook users in a Chrome context [15]. They used the LSTM model and compared it to other ML approaches. Its DL system performed wonderfully with an accuracy rate of 99.4%.

Ivana et al. [16] used hierarchical categorization in Indonesian news articles. They examined if hierarchical multi-label classification can be used to classify Indonesian news articles and produce valuable findings. For this purpose, the authors proposed a hierarchical multi-label classification strategy. There are three different approaches, namely flat, global, and top-down approaches used for this purpose. The authors employed calibrated label ranking and binary relevance to handle multi-label classification. The data was gathered from a two-level hierarchical news website. The news stories were divided into ten categories and four subcategories. J48, naive Bayes, and SVM were employed as classification methods.

Deep neural networks have become very popular for text classification due to their high expressive capacity and low technical requirements. Despite their allure, neural text recognition algorithms in many applications suffer from a lack of trained data. A study [17] developed an attention network designed hierarchically for document text classification. The authors tested their model on enormous amounts of data. The datasets used in their research were yelp, IMDB, and amazon reviews.

They compared their model to baseline approaches, such as SVM, word base CNN, and LSTM. The model created the document's vector by first merging similar terms with the vector of the sentence. It then merged the vector of the document with the vector of the sentence.

Safder et al. developed a DL model to represent the concepts stated in this under-resourced Urdu language [18]. A total of 566 items belonging to sports, culinary, software, political, and entertainment categories collected 10,008 ratings. This project had two goals, namely (a) creating a human-annotated corpus for sentiment analysis in Urdu and (b) evaluating the existing model performance using the corpus. Recurrent convolutional neural network (RCNN), rule-based, N-gram, SVM, CNN, and LSTM were all employed in the evaluation. With an accuracy of 84.98% for the classifier and 68.56% for ternary classification, the RCNN model outperformed the traditional approaches. In addition, this study's corpora and code have been made publicly available to assist other academics working on the same subject.

Ahmed et al. [19] proposed an SVM-based framework for categorizing Urdu news headlines. This method used feature vectors' maximum indexes to categorize Urdu news based on headlines into the appropriate pre-defined categories. The suggested system was evaluated against the current state-of-the-art methods.

Hamza et al. also attempted to classify Urdu texts [20]. For this purpose, they built the top-notch Urdu news dataset COUNT19 with pre-identified categories. Additionally, they assessed the effectiveness of state-of-the-art ML algorithms both with and without the use of

stop words removal and by stemming pre-processing stages. For feature representation of the text, unigram and bigram features with term frequency-inverse document frequency (TF-IDF) were utilized. According to the analysis, multilayer perceptron (MLP) has the highest accuracy of any classifier at 91.4%. The results also demonstrated that stemming does not enhance classifier performance. Stop words removal, however, has a detrimental effect on classifier performance.

Similarly, Javed et al. [21] suggested a hierarchical multi-layer LSTMs approach based on a deep neural network for the hierarchical text classification of Urdu news material. Their aimed at the hierarchical text categorization of Urdu news content. They initially used a text representing layer to obtain the numerical representation of the text and its categories by applying it to embed. After obtaining consolidated and ordered representation of Urdu news text and labels, they used Urdu Hierarchical LSTM Layer, an end-to-end fully connected deep LSTMs network, to carry out automatic feature learning. The results demonstrated that their approach works significantly better than both the industry standards and CNN.

III. Proposed Methodology

The proposed method takes a sequential approach, with the proposed model including sections such as data gathering, pre-processing techniques, feature engineering, classification methods, and performance measures. Figure 1 illustrates the numerous components in the proposed system, with the properties of each module described in the sections below.

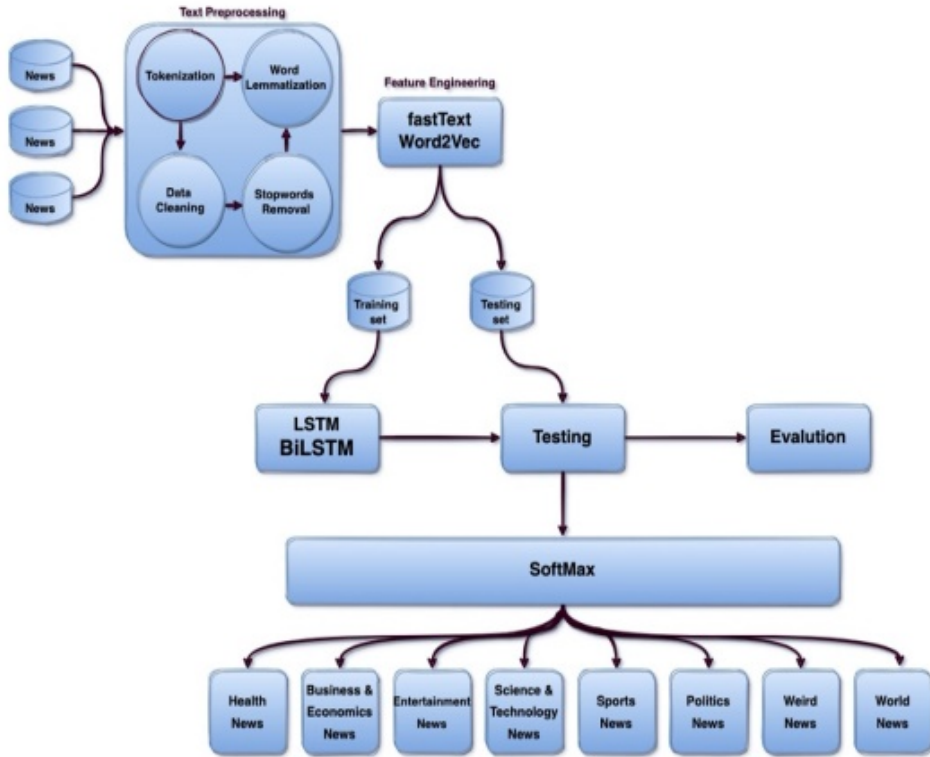


FIGURE 1. Proposed model framework

A. DATASET GATHERING

Any NLP activity needs a standard dataset. In this investigation, two types of datasets were utilized. The first dataset is a self-created dataset. The authors scraped the data from a wide range of internet news sources including Samaa news (1), Geo news (2), and BBC Urdu (3). For crawling Urdu news websites, python tools were utilized, namely Scrapy and Beautiful Soup. Furthermore, separate scrapers for each website were developed. The remaining data was obtained from the publicly available Mendeley website (4). Finally, the two datasets were combined. Figure 2 displays dataset distribution based on the number of documents for various types. There are 8 classes in all. The

categories are politics, entertainment, business and economics, health, world, sports, weird, and science and technology.

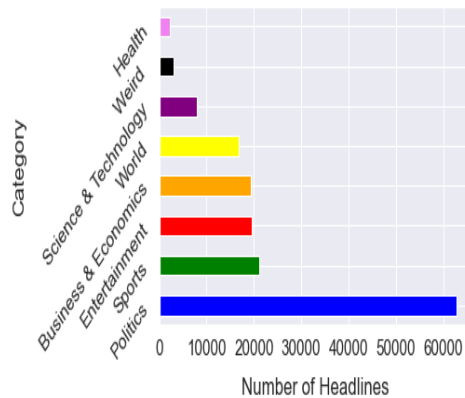


FIGURE 2. Dataset distribution

TABLE I
CORPUS CHARACTERISTICS

Category	Articles	Medals	Distinct Words
Politics	62,765	1,271,576	59,336
Sports	21,147	3,784,080	109,212
Entertainment	19,619	3,601,916	126,676
Business & Economics	19,419	2,780,660	79,956
World	16,853	3,306,97	27,603
Science & Technology	7,993	1,437,394	52,588
Weird	3,074	561,36	10,760
Health	2,180	389,98	6,410

When the two sets of data were combined, the documents were found to have 153,050 headlines in the corpus. Table III concludes the dataset characteristics for each class.

B. DATA PRE-PROCESSING

For the data, simple pre-processing techniques were used. Before using any ML method or deep neural network, data pre-processing is necessary. It aids in the removal of redundant material from the corpus by stripping the content of URLs, special symbols, English characters, and numeric values, leaving behind only the intended Urdu language text. This regulates the organized representation of textual data and increases the experiment's accuracy and efficiency. The following are the pre-processing processes used in the current study.

1) TOKENIZATION

NLP begins with this phase. It breaks down phrases into usable tokens or unique statements. It generates token charts from the raw text, with each token representing a word. Inaccuracy in tokenization may influence the findings of the experiment. Urdu is a less linguistic instrument, and its signals suffer from issues such as spatial

incoherence between words and detecting sentence limits [22]. Figure 3 depicts the conversion of Urdu text into tokens.

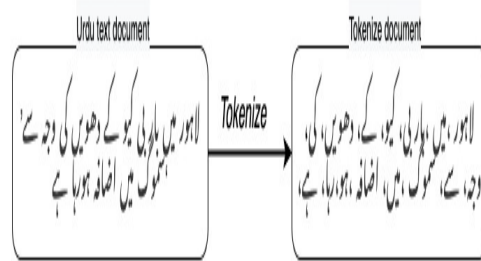


FIGURE 3. Urdu tokenized document

2) DATA CLEANING

Special signs or symbols, such as >, /, @, #, \$, %, &, *,), ""(, _, -, +, =, [,], as well as numbers with limited meaning in categorization were removed from the corpus.

3) STOP WORDS REMOVAL

Stop words were used to eliminate needless words. This technique is utilized to rid the terms that do not have much significance. Meaningless and useful words coexist in natural languages. Stop words are known in Urdu as *Harooof-e-Jar* and in English as conjunctions [21]. Before applying NLP, stop words are commonly omitted from the corpus. The study results can be improved by eliminating these terms. There are about 265 stop words in the Urdu language, which were removed. Some Urdu stop words are shown in Figure 4.

کونی کون کس نے سے آتے کے گئے آتی آئی آئیں آئے آنا آ
تھا تک کونی کو اگر کا یہی یہاں یہ یعنی گیا

FIGURE 4. List of stop words

4) LEMMATIZATION

A lemmatizer is similar to a stemmer in that it converts a word that has transformed

surface forms into its lemma or root form. Urduhack library was utilized to lemmatize the selected Urdu text. Figure 5 depicts the list of lemmatized words.

Input Word	Root / Lemma
مہنگیں	مہنگا
دکٹیں	دککت
نباتات	نبات
اسلامیات	اسلام

FIGURE 5. Lemmatized words

C. Feature Extraction

Textual data properties were sought after pre-processing. Word embeddings turn

each word that has an index into a fixed-size vector of constant integers. Figure 6 illustrates the pre-processing approaches used in this research.

In word vectors, a sequence represents each distinct word in the lexicon. Embedded algorithms form embedding matrices. As seen in the equation, it has a size of $s*d$ (1) [23], while the vocabulary size is s , and the dense vector dimension is d . Word representation models come in a range of shapes and sizes. This study aimed to evaluate which of the two word-embedding models, namely Word2vec and fastText (pre-trained), best conveyed the categorization relationship.



FIGURE 6. Example of pre-processing techniques

$$\begin{matrix}
 f_{1,1} \dots f_{1,d} \\
 f_{2,1} \dots f_{2,d} \\
 f_{3,1} \dots f_{3,d} \\
 \vdots \\
 M = \dots f_{s,1} \dots f_{s,d}. \quad (1)
 \end{matrix}$$

1) WORD2VEC

Word2vec is a well-known technique for converting natural language representations into distributed vector representations [24]. Since it can record contextual word-to-word exchanges in a multidimensional environment, it is extensively used as the first step for predictive models in semantic and information retrieval tasks. Skip-gram and continuous bag of words (CBOW) are distinct [25]. The target word is

inferred from context words by the CBOW component, while the skip-gram element implicates context words from an input text by using the skip-gram component [26]. The self-trained Word2vec has a 352121 vocabulary size. Table II shows the Word2vec model parameters.

TABLE II
PARAMETERS USE TO TRAIN
WORD2VEC MODEL

Vector size	Window_size	Workers	Min_count
100	5	10	1

2) PRE-TRAINED FASTTEXT

Facebook team developed fastText, a

framework for word embedding and text categorization. To build word vectors, the second version of fastText pre-trained Urdu embedding was used, which is available online. It delivers 300-dimensional word vectors trained with the CBOW model [27]. fastText, like Word2vec, obtains the representations of each phrase and the n-grams contained within each word to efficiently learn the representation of out-of-vocabulary (OOV) terms. Each training step evaluates the average of the values of the representations to create a single vector. As a result of these derived properties, neural word embeddings can retain meaningful sub-words, while being more computationally expensive than Word2vec and Glove. This is the result of using a variety of metrics. Further, fastText neural word embeddings outperform Word2vec. The trained fastText has a 1170883-vocabulary size. fastText performs better because it can generate word vectors for unfamiliar words or from vocabulary terms. It does this by considering the morphological properties of words to generate the word vector for an unfamiliar word. It often outperforms other tools for this task, since morphology pertains to the syntax or structure of the words. Secondly, unusual words function nicely with fastText.

D. DEEP LEARNING (DL) CLASSIFICATION

The improved and consistent performance of neural networks on vast and complicated datasets attracted researchers from a wide range of disciplines at the time of their initial presentation as an ML technique in 2006. Deep neural networks proved to be cutting-edge in several fields including

computer vision, voice recognition, NLP, and many others. CNN and RNN are the most common DL model designs. CNN has shown that it can learn data in a non-sequential form, which is better than learning data in time series. RNNs have a particular design that aids in learning time series data and retaining information over time. Two deep NN-based models mentioned in this section are used excessively, namely the LSTM and BiLSTM.

1) LONG SHORT-TERM MEMORY (LSTM)

Standard text processing systems process text separately. Due to a lack of historical data, many techniques cannot relate older data to the current data during processing. CNN successfully captures the local context between nearby words but falls short of capturing long-term associations. LSTM uses memory cells to maintain the network's state across time, enabling them to remember long-term associations between words. As a result, LSTM can extract more specific and valuable data from the entire text. Figure 7 displays an LSTM model.

The LSTM model comes with a word embedding layer. The size of the input layer is the same as the size of the word vector. LSTM units, which consist of input, forget gate, and output (that performs operations), are present in the hidden layer. These gates protect the LSTM from exploding and receding gradients. The mathematical LSTM model for a word sequence looks like this

$$it = \sigma(wi[ht-1, xt] + bi) \quad (2)$$

$$ft = \sigma(wi[ht-1, xt] + bf) \quad (3)$$

$$ot = \sigma(wi[ht-1, xt] + bo) \quad (4)$$

The input gate represents the bi biases for individual gates, the sigmoid function, the wi weight for the respective gate, the ht-1 output from the previous

LSTM gate at timestamp t-1, the text input at the current time stamp, and the ht-1 output from the previous LSTM gate at timestamp t-1 (x) [23]. There are two sequential levels, one with 128 units and the other with 64.

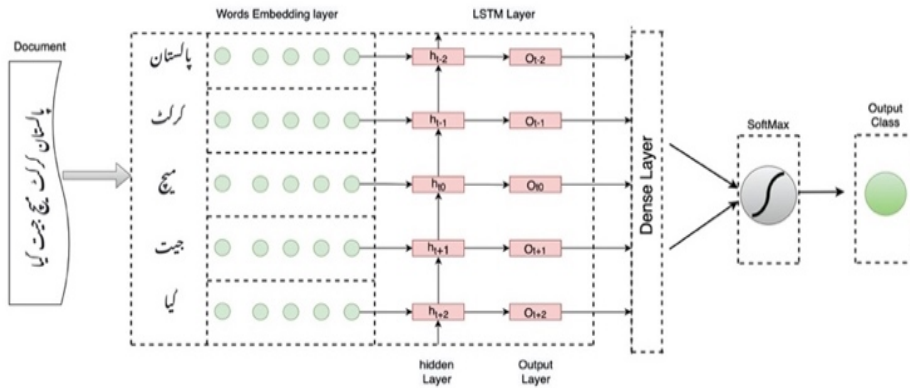


FIGURE 7. LSTM proposed model

2) BIDIRECTIONAL LONG SHORT-TERM MEMORY (BiLSTM)

Since it evaluates the current input utilizing both future and prior information, BiLSTM is by far the most effective RNN architecture. It is unlike LSTM, which relies solely on historical data. Figure 8 shows a BiLSTM model. Unlike the LSTM architecture, the BiLSTM architecture in

this study has two hidden layers, one for the backward pass and the other for the forward pass. Both concealed layers are subject to a single layer, the output layer that links to a fully connected layer for ultimate output. In addition, the BiLSTM comes with a word embedding layer. The size of the input layer is the same as the word vector's size. BiLSTM uses two sequential layers, one with 128 units and the other with 64 units.

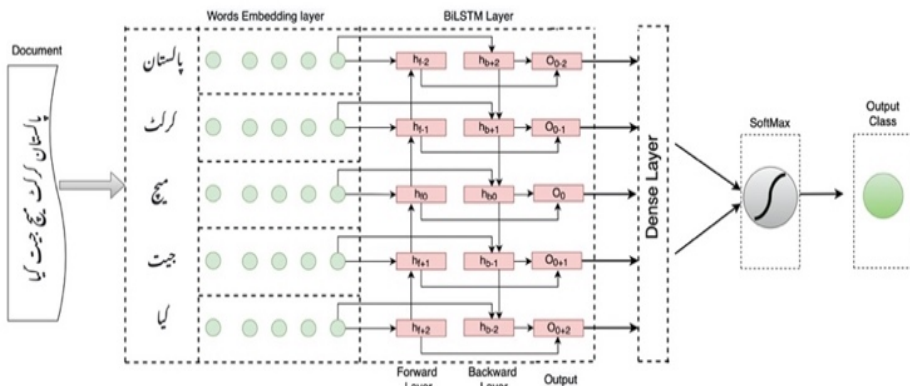


FIGURE 8. BiLSTM proposed model

3) DENSE LAYERS

The output of the previously hidden layer merges into a dense layer that ultimately connects to conclude the DL model. The dense layer receives all nodes from the preceding layer. Since the ReLU function is only employed in hidden layers, all DL models have 32 units in the dense layer using the ReLU activation function. The final dense layer forecasts the class using DL models. It has one neuron for each class in a multi-classification for 8 neurons.

4) OVERFITTING REGULARIZATION

To decrease the DL model's overfitting, several strategies are applied. The next section discusses how a handful of them were used in this research. Firstly, on the selected Urdu datasets, the authors tweaked DL model's hyperparameters. Choosing the optimal model parameter implies getting the best model performance.

4.1. Dropout

- Dropout randomly sets specific nodes to zero during training and eliminates them. It prevents the model from learning the same values over and over again when there are a lot of parameters.
- LSTM: The first layer of the LSTM model uses a dropout regularization value of 0.3 and the recurrent dropout is also 0.3. This study used a dropout regularization value of 0.15 for the LSTM second layer. The value of recurrent dropout is 0.15.
- BiLSTM: The first layer of BiLSTM has a dropout value of 0.3 and the value of recurrent dropout is also 0.3. The second layer takes a dropout size of 0.15 and a recurrent dropout value of 0.15.

4.2. Early Stopping

- This is a method used for preventing overfitting in which the classifier's efficiency on the testing dataset stops improving, regardless of the number of epochs utilized. The call-back function is used to complete training in the DL news categorization system.
- The call-backs record the performance metrics for each epoch, such as validation loss, validation accuracy, training accuracy, and training loss value. The two most significant aspects of this profession are monitoring and persistence.
- Monitor: Test accuracy has been designated a monitor indicating that it would keep managing test data loss.
- Patience: Patience was set to 2, so that if the accuracy rate does not increase after two epochs, the model would automatically drop the learning rate value.
- Optimizer: It is a DL technique that can adjust network weights repeatedly depending on training data, instead of using the standard stochastic gradient descent procedure. It is gaining prominence in NLP and computer vision. With a learning rate reduction function, the Adam optimizer was applied. This optimizer has a learning rate of 0.01.
- Categorical Cross Entropy Loss Function: In categorization with many classes' issues, categorical cross-entropy is a loss function. These are issues in which an example can only fall into one of the several categories and the model must figure out which one it is. Its legal purpose is to compare

and quantify the differences between two probability distributions.

E. SOFTMAX

There is wide use of several sigmoids to define the SoftMax function. It produces values ranging from 0 to 1, showing the likelihood that a data item belongs to a specific class. The SoftMax function was utilized to address multiclass classification issues in this project. This function returns/prevents the chance of a data point from falling into different classifications. In a multiclass issue, the number of neurons in the output layer is the same as the number of classes in the target. This study incorporated 8 classes. Thus, there would be 8 neurons in the output layer. The SoftMax activation function's equation is as follows (5):

$$Softmax = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{5}$$

The Z in the above equation signifies the neurons, which are the result of the output

layer. The exponential is a non-linear function. By dividing these values by the total of exponential values, SoftMax normalizes and transforms them into probabilities. These are the odds of a data point falling into each class. Table V summarizes the parameter settings for DL models.

IV. EXPERIMENTAL SETUP

Different embeddings were used to build an investigational setup to compare numerous DL approaches for evaluating the DL methods designed for Urdu news headline text categorization. These methods were tested for sentence-level categorization tasks in the domains of science and technology, world, entertainment, business and economics, sports, politics, weird, and health. The primary evaluation criterion was accuracy. The experimental setup is given in section 3.0. Table III depicts the gist of determiner patterns for all DL frameworks. The given portions elaborate the testing and outcomes.

TABLE III

SUMMARY OF THE PARAMETERS USED IN ALL DEEP LEARNING MODELS

Paramentes	LSTM		BiLSTM	
Words Embedding	Word2vec	fastText	Word2vec	fastText
Trainable Params	271,400	271,400	403,240	403,240
Non- Trainable Params	15,000,000	15,000,000	31,266,100	10,000,000
Lagers	2 LSTM		2 BiLSTM	
	2 Dense		2 Dense	
Units	128, 64		128, 64	
	32, 8		32, 8	
Loss	Categorical Cross entropy		Categorical Cross entropy	
Bath size	128		28	64
Embedding dimension	300		300	100
Epochs	20		20	
Vocab_size	312660	1170883	12660	1170883

Learning rate	0.01	0.01
Optimizer	Adam	Adam
Activation function	ReLU	ReLU
Classification function	SoftMax	SoftMax

The hypothesis was tested on the news dataset containing 122440 (80%) training and 30610 (20%) test instances. Table IV shows the training and testing details of the dataset used in the experiment. The number

of epochs and data size were used as factors in the two trials. Following are the details of the data utilized in the experiment, as well as the settings and evaluation methodologies.

TABLE IV
TRAINING AND TESTING DETAILS

Dataset	Training set	Testing set	Words (Avg)	Max_length
153,050	122440	30610	710	300

A. DEVELOPMENT ENVIRONMENT

All tests were carried out in the Jupyter Notebook 6.3.0 environment, using the TensorFlow library and KERAS. For plotting, the Matplotlib Python package was utilized. All experiments were carried out using an Intel Core i5-1.6 GHz Dual-Core processor, 8 GB of RAM, and Windows 10 operating system.

B. PERFORMANCE MEASURES

To evaluate the performance of categorization, the most used performance measures were employed, namely recall, precision, accuracy, and F-measure. These were calculated as follows:

- Precision: Precision can be used as a statistic to calculate the perfection of a classifier's output.

It is computed as follows:

$$Precision = \frac{TP}{TP + FT} \quad (6)$$

In this situation, precision was used as a performance measure metrics and the precision range was 93.86% to 92.33%.

- Recall: Recall measures the accuracy of the classification findings/results. It can be determined as follows:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

The recall ranged between 93.93% and 92.82%, BiLSTM with fastText outperformed other methods.

- F-measure: The harmonic mean of recall and precision is the F-measure. It can be deduced as follows:

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

The range of F1-measure utilized as the performance metrics matrix was 93.89% to 92.39%.

Accuracy: The most widely used statistic for classifier performance is accuracy. It can be determined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

Accuracy ranged from 93.93% to 92.50%, while the performance of BiLSTM with

fastText was superior.

- For various classes (A, B, and C), these notions are defined below.
- True Positive (TP): For class A, TP is the number of features correctly identified as belonging to that class by a classifier.
- True Negative (TN): The number of features that do not connect to class A but a classifier predicts them to correspond to class A are referred to as TN.
- False Positive (FP): The number of documents with labels that are not in class A but a classifier predicts them to be in class A.
- False Negative (FN): The number of documents that have true labels and belong to class A but a classifier projects them to belong to a different class.

V. RESULTS AND DISCUSSION

A. RESULTS

This section presents the results of various experiments conducted to evaluate the performance of several DL models. As described in section 3, two different word vector illustrations were presented as input to DL models. As a result of the embeddings, each model underwent four experiments. Additionally, the impact of alternative embedding representations on model performance was investigated. On the test set of Urdu news datasets, Figure 9 compares the training, testing loss, and accuracy of various DL models. During testing, it was discovered that the LSTM Word2vec model overfitted after only six epochs, while the BiLSTM Word2vec model overfitted after just two epochs; hence, both BiLSTM and LSTM training were stopped after 20 epochs.

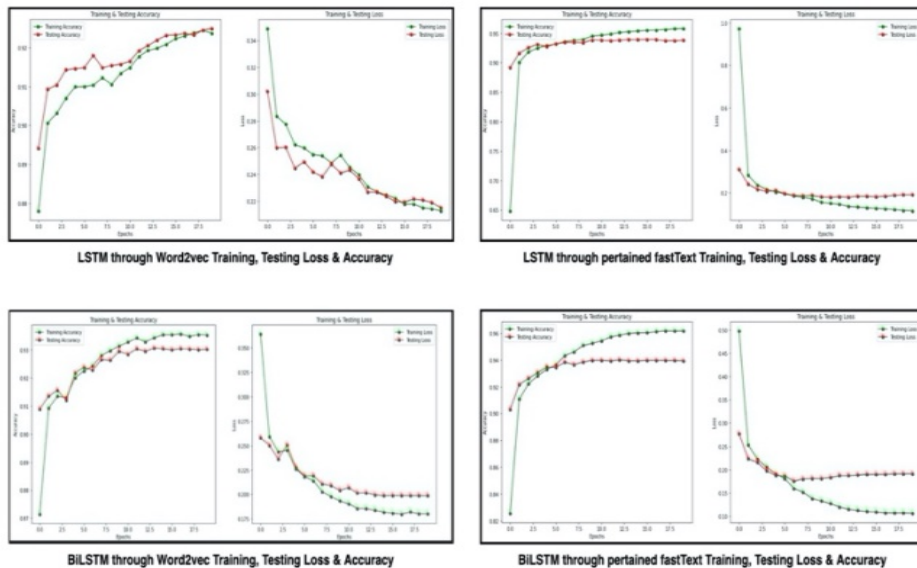


Figure 9. Comparison of training, testing loss, and accuracy of various DL models

Similarly, Figure 10 displays the confusion matrix of several DL models. BiLSTM, through trained fastText, predicted better than the rest of the models. The confusion matrix, which relies on predicted and actual label variances, was found to be a significant source of error detection. The sloping side of BiLSTM through trained fastText confusion matrix illustrates a

correct forecast, with the accurate tag of the science and technology class (1511), business and economics (3786), politics (12020), entertainment (3600), weird (423), sports (4263), health (360), and the world (2789). Figure 11 depicts class performance measures utilizing various proposed DL-based techniques based on Word2vec and fastText embeddings.

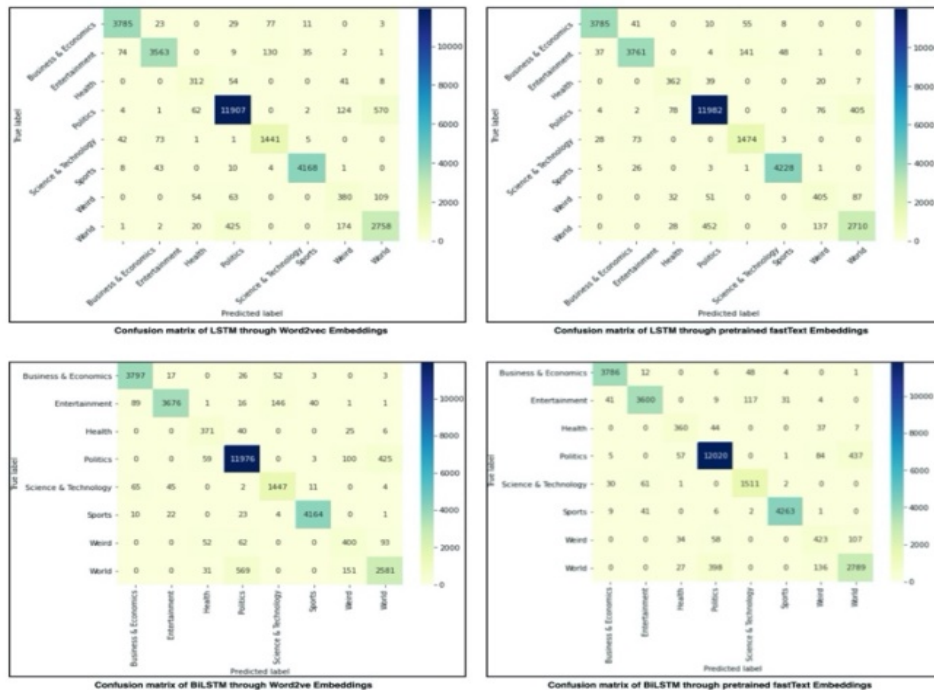


FIGURE 10. Comparison of confusion matrix of DL models

BiLSTM and LSTM via fastText classifiers significantly influenced others because they provided more than 93% of the measured matrix. In contrast, other classifiers found only a rare category above the 90% threshold. All models were not required to perform well according to the defined matrix; only a few model matrices provided with good results. The strategy was to split the dataset into training and testing halves, thus allowing the authors to

look into the causes of misclassification on the test set. The current findings improved since the authors employed a large recent corpus of Urdu literature and the pre-trained fastText model to enhance the word vectors. The second reason is that the authors increased the number of neurons in the Bi-LSTM model and adjusted the learning rate to 0.01, which improved accuracy.

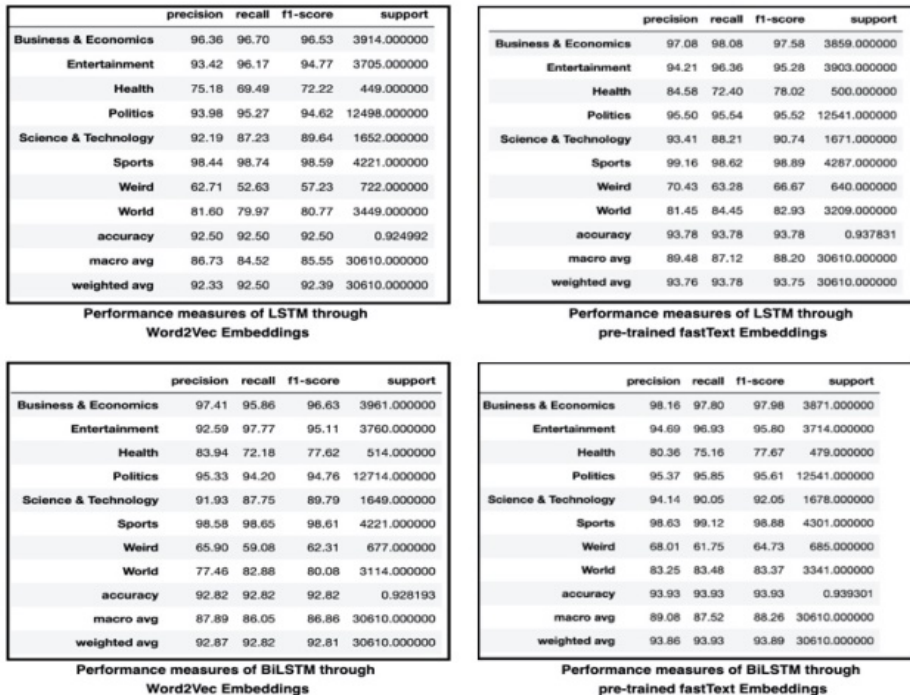


FIGURE 11. Classes’ performance measures utilizing proposed DL-based techniques

B. DISCUSSION AND COMPARATIVE ANALYSIS

In the initial stages, this research experimented with traditional ML models on the selected dataset and achieved satisfactory results[7]. These models, namely K-nearest neighbor, linear SVC, logistic regression, naïve Bayes, nearest centroid, perceptron, random forest, ridge classifier, SVM classifier, decision tree, and passive aggressive demonstrated varying levels of accuracy, precision, recall, and F-score. However, this investigation took a step further by incorporating DL models, specifically LSTM and BiLSTM, utilizing both Word2vec and pre-trained fastText

embeddings. Remarkably, the proposed DL models, particularly BiLSTM with fastText embeddings, substantially outperformed the traditional ML techniques. Table V shows enhanced accuracy, precision, recall, and F-scores for DL models, emphasizing the efficacy of these approaches, particularly in the context of a larger and more intricate dataset. This comparison highlights the superior performance and potential of these models in Urdu news classification as compared to the traditional ML methods. Although the current dataset has not been utilized by others, the current findings surpass those of previous studies conducted on smaller datasets. This study stands out due to its utilization of a larger and more comprehensive dataset.

TABLE V
COMPARATIVE ANALYSIS OF ML AND DL MODELS

No	Models	Accuracy	Precision	Recall	F-Score
1	K-Nearest Neighbors_classifier	0.84	0.76	0.74	0.74
2	Linear SVC	0.91	0.85	0.82	0.84
3	Logistic regression	0.90	0.86	0.78	0.81
4	Naïve_Bayes	0.83	0.81	0.64	0.65
5	Nearest centroid	0.75	0.70	0.75	0.68
6	Perceptron	0.89	0.82	0.80	0.81
7	Random forest	0.63	0.54	0.335	0.37
8	Ridge_classifier	0.89	0.85	0.78	0.80
9	SVM_classifier	0.91	0.85	0.82	0.84
10	Decision tree	0.73	0.57	0.47	0.49
11	Passive Aggressive	0.89	0.82	0.81	0.81
12	LSTM(Word2vec)	92.50	92.33	92.50	92.39
13	LSTM(fastText)	93.78	93.76	93.78	93.75
14	BiLSTM(Word2vec)	92.82	92.87	92.82	92.81
15	BiLSTM(fastText)	93.93	93.86	93.933	93.89

The proposed model employed pre-trained fastText embeddings for word representations. This is a significant finding, as fastText is known for its ability to effectively handle morphological variations and sub-word information, making it particularly well-suited for languages like Urdu with complex linguistic features. This choice of word embeddings enhanced the model's ability to grasp contextual nuances, thereby improving performance and generalization. While the proposed model has potential for biases, the findings align with established practices for maximizing model effectiveness, particularly in resource-constrained settings. Additionally, models utilizing word embeddings from Word2vec also demonstrated promising results.

C. LIMITATIONS

While the results showcase the efficacy of the proposed DL model in classifying Urdu news headlines, it is essential to

acknowledge certain limitations that may impact the model's performance in real-world scenarios. These limitations merit careful consideration for a more nuanced interpretation of the findings and act as a guide to future improvements. The proposed model's ability to handle diverse news topics in Urdu, while significant, may encounter challenges in the case of highly specialized or nuanced language used within specific domains. Furthermore, the imbalance in the dataset across various news categories may introduce biases and affect the model's accuracy. Understanding and addressing these limitations are critical steps towards refining the model and enhancing its applicability across a broader spectrum of Urdu news classification tasks.

D. CONCLUSION

This study compared the performance of two DL models for classifying Urdu news content. It marks an essential turning point in the classification of Urdu news. It

included two alternative neural word embeddings and a publicly available and self-created dataset. Firstly, this study used DL techniques to categorize text in Urdu news headlines. Since DL algorithms are not applied often in Urdu textual data categorization, it faced several challenges including linguistic complexity, a lack of resources, and viable remedies. The study also analyzed the impact of various pre-trained and self-trained word embeddings on all DL architectures. Two distinct embeddings were used to determine the correctness of each model. BiLSTM proved to be the most effective model using fastText, outperforming all others regardless of embeddings. The authors used the fastText (pre-trained) and Word2vec embeddings in testing to find the most acceptable word embedding representation for each dataset. It was found that fastText frequently performed better than other embeddings because it can create word vectors for unknown words or vocabulary terms. Secondly, fastText works well with unique words.

E. FUTURE DIRECTIONS

A news organization may use this study in the future, as well as blogs and chatbots. Decision-making with massive datasets would improve by employing additional DL algorithms, such as BiLSTM GRU and several hybrid method-based DL classifiers.

In the future, the authors intend to broaden their scope by including additional languages and leveraging datasets from diverse organizations to enhance and market solutions. Their upcoming efforts involve rigorous testing of the developed models on varied datasets, encompassing different languages and text types. This comprehensive approach aims to evaluate the models across diverse linguistic and

contextual domains, ultimately strengthening the robustness and applicability of the proposed DL solutions.

REFERENCES

- [1] M. Iqbal, B. Tahir, and M. A. Mehmood, "CURE: Collection for Urdu information retrieval evaluation and ranking," in *Int. Conf. Digit. Fut. Transform. Technol.*, May 2021, pp. 1–6, doi: <https://doi.org/10.48550/arXiv.2011.00565>.
- [2] A. Daud, W. Khan, and D. Che, "Urdu language processing: A survey," *Artif. Intell. Rev.*, vol. 47, pp. 279–311, 2017, doi: <https://doi.org/10.1007/s10462-016-9482-x>.
- [3] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv*, arXiv:1607.01759, 2016, doi: <https://doi.org/10.48550/arXiv.1607.01759>
- [4] I. Rasheed, H. Banka, and H. M. Khan, "A hybrid feature selection approach based on LSI for classification of Urdu text," in *Machine Learning Algorithms for Industrial Applications*, S. Das, S. Das, N. Dey, and A. E. Hassanien, Eds., Springer, 2021, pp. 3–18, 2021, doi: https://doi.org/10.1007/978-3-030-50641-4_1
- [5] I. Rasheed, V. Gupta, H. Banka, and C. Kumar, "Urdu text classification: A comparative study using machine learning techniques," in *13th Int. Conf. Digit. Inform. Manag.*, Sep. 2018, pp. 274–278, doi: <https://doi.org/10.1109/ICDIM.2018.8847044>

- [6] T. B. Shahi and A. K. Pant, "Nepali news classification using Naive Bayes, support vector machines and neural networks," in *Int. Conf. Commun. Info. Comput. Technol.*, Feb. 2018, pp. 1–5, doi: <https://doi.org/10.1109/ICCICT.2018.8325883>
- [7] K. I. Malik, "Urdu news content classification using machine learning algorithms," *Lahore Garri. Univ. Res. J. Comput. Sci. Info. Technol.*, vol. 6, no. 1, pp. 22-31, 2022, doi: <https://doi.org/10.54692/lgurjcsit.2022.0601274>
- [8] M. N. Asim, M. U. Ghani, M. A. Ibrahim, W. Mahmood, A. Dengel, and S. Ahmed, "Benchmarking performance of machine and deep learning-based methodologies for Urdu text document classification," *Neural. Comput. Applic.* vol. 33, pp. 5437–5469, 2021, doi: <https://doi.org/10.1007/s00521-020-05321-8>
- [9] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Info. Process. Manag.*, vol. 57, no. 1, Article no. 102121, 2020.
- [10] J. Xie *et al.*, "Chinese text classification based on attention mechanism and feature-enhanced fusion neural network," *Computing*, vol. 102, pp. 683–700, 2020, doi: <https://doi.org/10.1007/s00607-019-00766-9>
- [11] J. A. Díaz-García, C. Fernandez-Basso, M. D. Ruiz, and M. J. Martin-Bautista, "Mining text patterns over fake and real tweets," in *Int. Conf. Info. Process. Manag. Uncert. Knowledge-Based Syst.*, 2020, pp. 648–660, Springer, doi: https://doi.org/10.1007/978-3-030-50143-3_51
- [12] J. Gong *et al.*, "Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification," *IEEE Access*, vol. 8, pp. 30885–30896, 2020, doi: <https://doi.org/10.1109/ACCESS.2020.2972751>
- [13] X. Xiao, S. Lian, Z. Luo, and S. Li, "Weighted res-unet for high-quality retina vessel segmentation," in *9th Int. Conf. Info. Technol. Med. Edu.*, 2018, pp. 327–331, doi: <https://doi.org/10.1109/ITME.2018.00080>
- [14] M. A. Ramdhani, D. S. A. Maylawati, and T. Mantoro, "Indonesian news classification using convolutional neural network," *Indo. J. Elect. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 1000–1009, 2020.
- [15] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Appl. Soft Comput.*, vol. 100, Article e106983, 2021, doi: <https://doi.org/10.1016/j.asoc.2020.106983>
- [16] I. C. Irsan and M. L. Khodra, "Hierarchical multilabel classification for Indonesian news articles," in *Int. Conf. Adv. Info. Concept. Theo. Appl.*, 2016, pp. 1–6, doi: <https://doi.org/10.1109/ICAICTA.2016.7803108>
- [17] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. 2016 Conf. North Am. Chap. Assoc. Comput.*

- Linguist. Human Lang. Technol.*, 2016, pp. 1480–1489.
- [18] I. Safder *et al.*, "Sentiment analysis for Urdu online reviews using deep learning models," vol. 38, no. 8, p. e12751, 2021, doi: <https://doi.org/10.1111/exsy.12751>
- [19] K. Ahmed, M. Ali, S. Khalid, and M. Kamran, "Framework for Urdu News headlines classification," *J. Appl. Comput. Sci. Mathemat.*, no. 21, 2016, doi: <https://doi.org/10.1111/exsy.12751>
- [20] S. A. Hamza, B. Tahir, and M. A. Mehmood, "Domain identification of urdu news text," in *22nd Int. Multi. Conf.*, 2019, pp. 1–7, doi: <https://doi.org/10.1109/INMIC48123.2019.9022736>
- [21] T. A. Javed, W. Shahzad, and U. Arshad, "Hierarchical text classification of urdu news using deep neural network," 2021, doi: <https://doi.org/10.48550/arXiv.2107.03141>
- [22] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. Fayyaz, "Exploring deep learning approaches for Urdu text classification in product manufacturing," *Enter. Info. Syst.*, vol. 16, no. 2, pp. 223–248, 2022, doi: <https://doi.org/10.1080/17517575.2020.1755455>
- [23] U. Naqvi, A. Majid, and S. A. Abbas, "UTSA: Urdu text sentiment analysis using deep learning methods," *IEEE Access*, vol. 9, pp. 114085–114094, 2021, doi: <https://doi.org/10.1109/ACCESS.2021.3104308>
- [24] H. Liu, "Sentiment analysis of citations using word2vec," *arXiv*, arXiv:1704.00177, 2017, doi: <https://doi.org/10.48550/arXiv.1704.0177>
- [25] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, 2015, doi: <https://doi.org/10.1016/j.eswa.2014.09.011>
- [26] H. Peng, Y. Song, and D. Roth, "Event detection and co-reference with minimal supervision," in *Proc. 2016 Conf. Empiri. Methods Nat. Lang. Process.*, 2016, pp. 392–402.
- [27] F. Mehmood, M. U. Ghani, M. A. Ibrahim, R. Shahzadi, W. Mahmood, and M. N. Asim, "A precisely xtreme-multi channel hybrid approach for Roman Urdu sentiment analysis," *IEEE Access*, vol. 8, pp. 192740–192759, 2020, doi: <https://doi.org/10.1109/ACCESS.2020.3030885>