Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 899–907

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

# Indian News Headlines Classification using Word Embedding Techniques and LSTM Model

Madhusmita Khuntia, Deepa Gupta*

*Computer Science & Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham*
*Bangalore-560099, India*

## Abstract

Newspapers introduce us to the latest happenings around the world. Going paperless creates more opportunities for newspapers, like broadcasting news coverage and presenting breaking news conveniently. News headlines are considered under the short text category and are vibrant subjects for researchers. Creating a dense vector from short texts has become a challenging and essential task in many applications such as recommender systems, context analysis, decision making, text classification, etc. This work not only targeted creating a classification model for the short text but also categorized the headlines with the 'unknown' category. Our work uses Bidirectional Encoder Representations from Transformers (BERT), cosine similarity index, word embedding, and Long Short-Term Memory (LSTM) network to classify news headlines in multiple categories. Our proposed method outperforms labeling the unlabeled data with the help of a BERT sentence encoder. The system uses LSTM to learn the headlines as input vectors and classify the headline text by the classifier. At the end of this experiment, the designed pipeline achieves remarkable precision at the class level.

*Keywords*— News headlines, Word Embeddings, Multi-label classification, LSTM, BiLSTM;

## 1. Introduction

With the evolution of network communication and e-commerce, short text data has increased enormously. There are many ways through which short texts are usually produced, for example, chats, news feeds, customer reviews, etc. Social media is a significant information source for many and a popular tool for politicians and companies to promote their products and services [1]. Short text categorization is an extensively considered and captivating subject in the text mining domain. Because of algorithmic limitations, the traditional text classification based on long text is not working as expected on the short texts, so some issues like non-standard ability and data sparsity are occurring. Several scientists applied different statistics to handle the data sparsity problem by applying internal and external semantic methods to short texts and improving classification data performance [2]. It is well noted that Machine learning is used for text analysis, where it is used to perform sentiment analysis [3]. Sentiment analysis is one method to try and understand people's emotions when they use text messages [28]. A document-based classification model generated word embeddings using Fasttext and the GloVe embedding. The authors Liu et al. [4] best performing network is trained for 300 epochs, and the test precision is recorded after each epoch.

* *E-mail address:* smita01madhu@gmail.com, g_deepa@blr.amrita.edu

10.1016/j.procs.2023.01.070

Similarly, content distributed on the news portals uses categorical words on their websites in the form of dictionary structure. These news types are often applied to various applications like news source recommendations and news category extraction related to time. According to Deng et al. [5], it is a common problem for data scientists to classify datasets whose domain is the news. Because of the restricted size of news headlines, the inference method of classifications gives an unacceptable result on short texts. Virtual platform has produced many outlooks for news headlines to present news more manifestly.

Our work has been done on short text to categorize each headline in a news dataset with incorrect and no labels. If headlines are unlabelled, it's usually impossible for users to find the desired info for which they are captivated. This paper proposes a multi-label classifier based on headlines description with specific categories and assesses model performance on highly complex data. Specifically, word embedding techniques such as Word2Vec, Glove, and DNN (deep neural network) with Long Short-Term Memory (LSTM) network have been used to observe the appropriateness of the model on more than two categorical data.

This research has organized as follow: first, some related works or literature survey has been mentioned in Section II. Section III explains the data source, and Section IV presents the pre-processing and algorithms applied to the data. Then, Section V describes the outcome gained by algorithms. At last, the conclusion is written in Section VI.

## 2. Related work

Word embedding techniques use probabilistic and semantic approaches to calculate the feature vectors. Numerous studies reported the text classification method for news headlines based on word embeddings and feature union proposed in Li, Gao et al. [6]. Word embedding, which represents words as low-dimensional vectors, has been shown to possess excellent properties for describing the semantic meanings of words. Apart from improvising the algorithms, different kinds of literature are also available to lead the importance of "domain-specific words" in the form of keywords which helps to improve the accuracy of English and mixed language with Wiki data [7-9]. Also, works associated with the improvisation of short text classification methodology were supported by the LDA topic model and KNN algorithm for performance measurement on news data [10-12]. A temporary text classification and question answering method based on word vector and LDA topic model is proposed with the factors of Grammatical Category-combined Weight and the Topic High-frequency Word by Shen Zheng et al. [13-16]. The SVM-based classification algorithm has shown significant performance improvement.

Short text classification with Wikipedia and Word2vec has been done to reduce the sparsity problem on massive amounts of data by author Liu Wensen et al. [17]. In this research, they first tried to find a set of articles that has higher relevancy to Wikipedia's core concepts. Then the semantic relatedness was established between the essential idea and its relevancy concept of Wikipedia (WLM algorithm). Lastly, the feature extraction for short text was performed. Finally, the author concluded that Wikipedia with the Word2vec model performed significantly better than other algorithms. In Olga Fuks et al. [18], word label TF-IDF and word embedding techniques for vector representation have been completed. Then accuracy comparison between different ML algorithms like Naive Bayes, Logistic Regression, Kernel SVM, and Random Forest was executed. Finally, the combined model of CNN and RNN achieved higher accuracy w.r.t ML algorithms. Some other works with word vector model like Word2Vec, WTTM (Word-network Triangle Topic Model), and BTM was performed. This work integrates the models to check the accuracy. The experimental results show that Word2Vec + WTTM model performs significantly better [19]. Many pieces of research are available to calculate short text's similarity index [20]. First, extract the word embedding of the news headlines from the BERT model. This way, the embedding space gets integrated with the whole text corpus. Second, introduce feature union to the classification of news headlines, which consists of two strategies based on the classification results of different feature types. Word embedding, which represents words as low-dimensional vectors, has been shown to possess excellent properties for describing the semantic meanings of words.

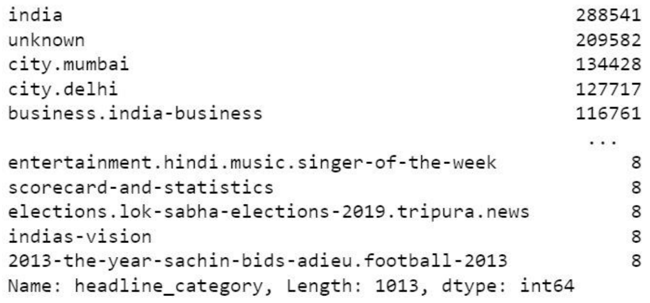
In Ma, Liu et al. paper, a novel hybrid embedding-based text representation method is introduced [21]. The proposed method employs an unsupervised joint learning framework consisting of two steps: constructing word embeddings for words using pre-trained word embeddings and learning the hierarchical classifier. The classification task is performed at the document level, and the hierarchical structure is exploited on the label. The effectiveness of the proposed method through comprehensive experiments on several benchmark datasets shows that it outperforms several state-of-the-art methods. According to Meng et al. [22], text categorization is a critical part of any news processing system. It goes more challenging with the NY times dataset because most of the text doesn't belong to any categories. Text classification is crucial to a news organization as it is the first step in building keyword lists for web crawling and is also part of information retrieval. The trickiest part of this task was to train the model. The model must learn the categories w.r.t to the news headlines, but there were not enough training samples.

In the wake of going through the above examinations, a conclusion can be made that the algorithmic productivity could vary if there is an occurrence of short text and long text. Even studies on embedding and most complex DNN models also failed to give the expected output because of the highly disordered dataset.

Our study differs from the above literature works mainly because this work not only emphasizes the algorithmic output and accuracy comparison, analyses the raw data, and highlights the different algorithms that have been used to prepare well-organized training data for model learning out of a completely disorganized dataset in text mining domain. At the end of the experiment, the combination of word embedding with deep learning models for short text classification has significantly improved performance.

## 3. Data source

The short text research work has been carried out on news headlines. The dataset is known as "India News Headlines" and is collected from Kaggle. The dataset contains news headlines of India between the period Jan-2001 to Dec-2020, which contains 34,24,067 news headlines (short text) with a file size of 226 MB. The dataset is in CSV format. The file contains three columns "Publish date," "headline category," and "headline text." The challenging and observable part of the data was "headline categories," as the ask in Kaggle was to convert 1013 existing headline categories into eight categories.



```
india                                              288541
unknown                                            209582
city.mumbai                                        134428
city.delhi                                         127717
business.india-business                            116761
                                                    ...
entertainment.hindi.music.singer-of-the-week            8
scorecard-and-statistics                                8
elections.lok-sabha-elections-2019.tripura.news         8
indias-vision                                           8
2013-the-year-sachin-bids-adieu.football-2013           8
Name: headline_category, Length: 1013, dtype: int64
```

Figure. 1. News headlines categories name and count.

Apart from a high number of categories, it's also observed that the categories' names were inappropriate. There were many instances where it belonged to an unknown type, as shown in figure 1. This concludes that the data is highly disorganized. So, it needs a lot of functional and algorithmic work to organize the categories w.r.t headlines before passing it as training data to the classification-based LSTM model.

The data organization process has been initiated by dividing the dataset into two data frames. One with category names 'India' and 'Unknown,' containing nearly 500k headlines, and the other data frame with random categories containing 290k headlines. The manual observation was performed upon 1011 categories (after filtering out "India" and "unknown" categories out of 1013 categories). The data were irrelevant since the same categories were written multiple times in different manners. For example, Asian games, Olympics, hockey, etc., can be defined as sports, similarly to other categories. Since the categories are distributed granulated, the number of categories is high. According to the requirement, 1011 categories need to be converted into 8 categories: regional, entertainment, economy, education, sports, health, politics, and others (categories that don't belong to the other seven categories), which are most popularly populated in online portals. The conversion technique of 1013 categories into 8 types is clearly explained in Section 4.

## 4. Methodology

The architecture followed for the classification work is shown in figure 2. The proposed methodology consists of different modules: first, create labels for unlabelled and wrongly labeled data by using logical functions, bidirectional encoder vector and cosine similarity, second data acquisition for classification model, third data pre-processing, fourth embedding vector creation and then classification model creation. In the flow diagram (figure 2), the data were collected and then divided into two parts 1) news headlines with random categories and 2) news headlines without any categories like unknown. In the first part, news headlines with random categories (nearly 1011) were compressed to the required 8 categories. The second part of data or news headlines without categories went through the sentence similarity process to label these news headlines with category names, after which the data whose sentence similarity index is more than 84% were merged (the block "organized labeled data") with proper categories data. Then the text pre-processing took place, after which the pre-processed data were converted into embedding vectors. Finally, the classification task was performed on embedding vectors.

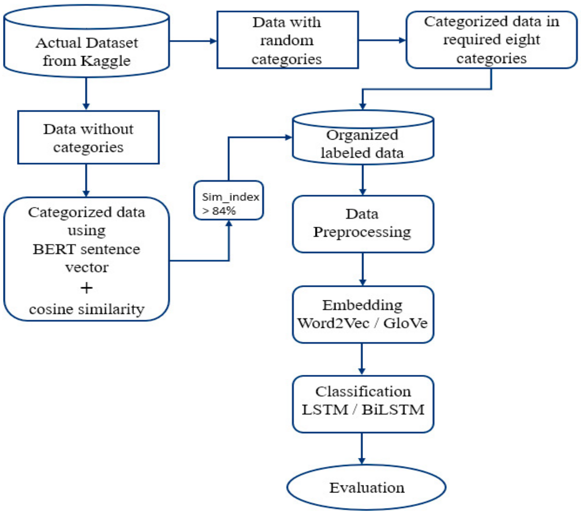

Actual Dataset from Kaggle
Data with random categories
Categorized data in required eight categories
Data without categories
Organized labeled data
Sim_index > 84%
Categorized data using BERT sentence vector + cosine similarity
Data Preprocessing
Embedding Word2Vec / GloVe
Classification LSTM / BiLSTM
Evaluation

Figure 2. Flow Diagram of a proposed headline classification framework

### 4.1 Data Labelling

Data observation helps to create eight arrays for the required eight categories (regional, entertainment, economy, education, sports, health, politics, and others). For example, from the above decided 8 categories, a list of objects for sports, like FIFA, Commonwealth Game, Asian game, Olympics, FIH hockey world cup, etc., were categorized under a single sports category. A similar approach was taken for the other seven categories. Finally, this work got the first data frame with eight required categories.

The second data frame creation was challenging since the headlines lacked categories. The bidirectional encoder representations from transformers for sentence vector creation (figure.3) has applied. A sentence-transformer library was imported (bert-base-nli-mean-tokens) to train and use the transformer model for generating sentence vectors [29]. BERT is good at creating dense vectors. BERT base contains 768 values for numerical representation of a single token so that the output of each encoder layer is also dense. Short texts have very few words, so, dense vector extraction for contextual word embedding is quite complex. Therefore BERT model was used for sentence encoder. Headline text with categories (2) data frame is taken as a base data for similarity index check w.r.t headline text with no categories (1). Once cosine similarity is calculated, the associated category of the base data frame is assigned to the second data frame where the similarity index was more than 84%. Finally, the final data frame with labels is created and passed for merging and creating training data.

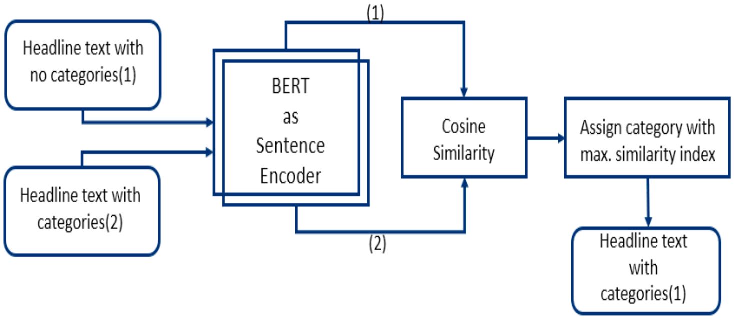

Headline text with no categories(1)
Headline text with categories(2)
(1)
BERT as Sentence Encoder
(2)
Cosine Similarity
Assign category with max. similarity index
Headline text with categories(1)

Figure. 3. Flow Diagram to label the unlabelled data.

### 4.2 Data Acquisition

Data cleaning and fetching is the essential part of this work—the data frame creation by applying two different methodologies to categorize the headline text in eight categories. Then by merging both data frames, the final data frame was created for the classification task, as shown in table 1. The first data frame contains 2.9 million records. The second data frame before processing had 490K records, of which only 200k got a similarity index of more than 84% in all categories. Based on their number of documents, the two datasets have been merged and created into a new dataset.

Table. 1: Final categories before and after manual sampling.

| First Data Frame | | Second Data Frame | | Final Data Frame | |
|---|---|---|---|---|---|
| regional | 1923932 | politics | 35992 | regional | 353000 |
| entertainment | 268897 | education | 29811 | entertainment | 285656 |
| others | 233600 | health | 26179 | others | 258633 |
| economy | 158837 | economy | 25248 | economy | 184085 |
| education | 138454 | others | 25033 | education | 168265 |
| sports | 136695 | regional | 21724 | sports | 156705 |
| health | 43900 | sports | 20010 | health | 70079 |
| politics | 21629 | entertainment | 16759 | politics | 57621 |

This research also observed that more than half of the data lie under regional categories in data frame one, creating high data imbalance issues for model training. So, a manual sampling technique was applied, and a random selection was performed on nearly 300k regional data out of 1.9 million. Finally, 1534044 total data were extracted with assigned categories which are better-balanced data for classification model training.

### 4.3 Data Pre-processing

Text pre-processing is a crucial step for further text analysis, which helps to clean irrelevant words from the required texts. All text pre-processing techniques like punctuation removal, lower case conversion, and stopword removal have been used. If any special characters are present in the final data, those are also removed using pre-processing techniques. Deleted stopwords have no adverse effect on the result of the Word2Vec or Glove algorithm [23-26], but algorithms that process data without stopwords have a lower training time. After pre-processing, the embedding task was executed upon the headline or short text explained in the next session. The data used for news headline classification were very generic; therefore, out of vocabulary (OOV) word was null for Word2Vec and BERT model, but in the GloVe embedding model, we found less than 100 words that were not present in the embedding dictionary. The presence of OOV word (67) with respect to unique words (171355) is 0.03% in the GloVe model.

### 4.4 Headline Text Embedding

According to our architecture diagram, embedding techniques were applied to the pre-processed data. Then some filtering techniques were used on the dataset. The maximum length of the headlines was calculated to determine the window size. Once the maximum size of headlines was calculated, the next task was to find the number of unique words in the whole dataset. Then the tokenizer function was used to tokenize the entire text after which embedding techniques were applied to the final dataset using the Word2Vec (CBOW) library and GloVe library. The encoded categories were created by applying one hot encoding technique. The final embedded data with encoded classes were passed as an input for our classification model as described in section 4.5.

### 4.5 Headline Text Classification

The Glove model is designed based on supporting global word-to-word co-occurrence counts considering the entire corpus. Word2vec, on the other hand, supports co-occurrence within a local context like neighbouring words. The whole dataset was split as train and test labels after data were identified as embedded vectors. From the output of Word2Vec and Glove embedding, 80% of the data were randomly used for training. The internal architecture of the program is shown in figure 4.

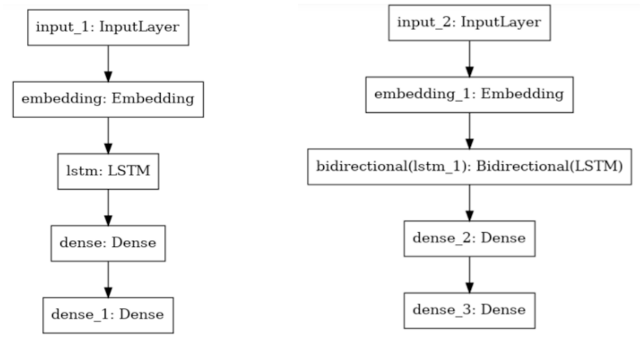


Figure 4: Internal Architecture of deep learning models.

Our model creates an embedding layer that helps to calculate sequence input and pass the output to the LSTM/BiLSTM network as an input. Our LSTM/BiLSTM network contains two dense layers with an input layer, softmax, and relu as activation functions. All these operations were performed to classify the news headlines.

Python was used as a programming language for the project work. Gensim Word2Vec model frameworks were chosen to perform the embedding task. For this work, GloVe-6B has been selected from the Stanford website. The Keras API with TensorFlow was used to create LSTM and BiLSTM networks. NLTK and other tool kits were used for the text pre-processing task. The NLTK tool kit contains extensive communal features purposely designed for text processing.

## 5. Results and discussion

### *5.1 Performance Evaluation and Analysis*

The task of evaluating a Short Text Multiclass Classification is difficult because the content is challenging to capture in a sentence having a lesser number of words. Evaluation can be done in different ways based on the system's effectiveness. This research focused on the classification with the existing algorithms like using BERT sentence vector to label the data, using Word2Vec and GloVe as embedding techniques, and classification using the LSTM and BiLSTM models.

Text categorization without appropriate training data is complex and requires correctly enough hand-labeled data to apply supervised methods [27]. Therefore, it was learned that a human assessor's assessment would be the right evaluation measure for labeling or defining the basic categories of data. Instead of classifying the raw data into 1013 inappropriate categories, this research defined eight categories. Using two different approaches, labeled data preparation has been done for different classification models and discovered that the accuracy of a statistical model is better w.r.t other approaches.

### *5.2 Experimental Setup and Results obtained*

In this project, two different models have been used to get better accuracy for this highly challenging data. First, bidirectional encoder representations from transformers as sentence vectors are used to label the unlabelled data. Then by applying a filter on the similarity index, the labeled data were selected whose similarity index was more than 84% and appended them to the final dataset. After that, embedding was performed on the final dataset using Word2Vec and GloVe vectors. The Word2Vec model was trained with 150 embedding dimension size and fed to three-layer LSTM and BiLSTM networks to perform the classification task. Then statistical model GloVe was introduced with a 150 embedding dimension size to the similar LSTM/BiLSTM network.

Table 2. Weighted average comparison table for different models

| **Models** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Wor2Vec+LSTM | 63% | 64% | 63% |
| Glove+LSTM | 64% | 65% | 64% |
| Word2Vec+BiLSTM | 64% | 64% | 64% |
| GloVe+BiLSTM | 5% | 23% | 9% |

Table 3. Macro average comparison table for different models

| **Models** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Wor2Vec+LSTM | 60% | 56% | 57% |
| Glove+LSTM | 60% | 56% | 58% |
| Word2Vec+BiLSTM | 59% | 57% | 58% |
| GloVe+BiLSTM | 3% | 12% | 5% |

Table 4. Accuracy comparison table for different models

| **Models** | **Accuracy** |
|---|---|
| Wor2Vec+LSTM | 63.31% |
| Glove+LSTM | 64.57% |
| Word2Vec+BiLSTM | 64% |
| GloVe+BiLSTM | 22.88% |

The evaluation parameters like f1-score, precision, and recall are defined in Table 2 and Table 3 for weighted average and macro average with four different models. Table 4 shows the classification accuracies of embedding models on the news headlines dataset with the LSTM/BiLSTM model. The GloVe + LSTM model demonstrates a higher classification performance with 150 embedding dimension size than the other embedding models. Even though the accuracy gap is not high between Word2Vec + LSTM models, GloVe + LSTM is giving higher accuracy in most categories.

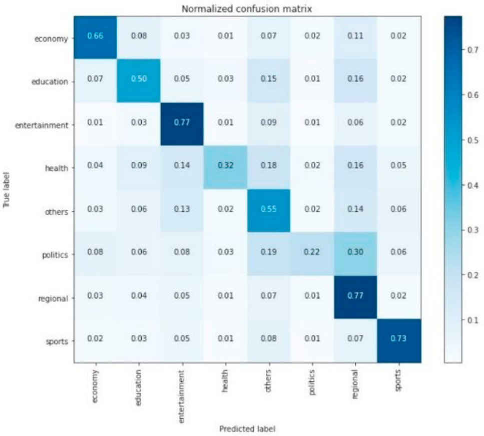


Figure 5. Confusion matrix for GloVe + LSTM model.

From figure 5, the classification accuracy for individual classes can be observed. Also, it can be concluded that the precision, recall, and f1-score for politics and health categories are very low compared to other types since the amount of training data from politics and health categories were low, causing a drop in accuracy. The class imbalance also contributed to poor performance in politics and health categories. By adding more training data or using a good class balancing algorithm, model performance can be improved.

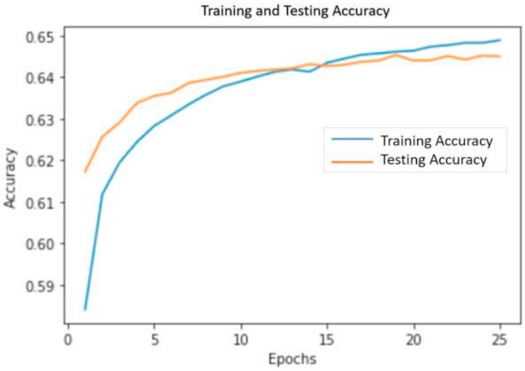


Figure 6. Training and testing accuracy graph for GloVe + LSTM model.

The graph shows the testing and training accuracy concerning other algorithms. Suppose the analysis in the diagram (figure 6) is taken place. In that case, the variation between training and validation data which is quite the same in both cases is notable in the GloVe + LSTM model. The validation accuracy is constantly growing with respect to every epoch. The results would contrast with other models if the number of epochs increased. Since this work used 25 epochs, after analyzing all the parameters, it can be concluded that the GloVe + LSTM model performs better on our dataset than other models.

## 6. Conclusion

For news headline classification, the experimental approach targets checking the efficiency of these newer embedded models derived from the deep learning community. The resultant tables shown in Section V illustrate that the LSTM network can learn the dissimilarities between the news headlines with eight classes in the given dataset.

The presented approach uses various word-embedding algorithms and a deep neural network technique to increase the classification results of the structure on the Kaggle news headlines dataset. So, by using the Word2Vec, Glove, LSTM, and BiLSTM, this research can conclude that GloVe with the LSTM model is an optimal model for this news headlines dataset. Suppose the training data of the GloVe model was inappropriate. In that case, it may produce a suboptimal vector representation, leading to the LSTM network being learned using this incorrect vector representation. It could also cause poor accuracy/result. Based on the successful consequence received on the word vector, enhancements can also be introduced for LSTM to produce better results.

In the future, the improvement in accuracy could be seen by putting more effort into data preparation tasks like the manual observation of headline text w.r.t headlines categories needs to be done to prepare correctly labeled data. Apart from that, the result can be enhanced by increasing the number of iterations and selecting advanced classification models like BERT, ELMO, etc.

## References

[1] Kozlowski, M., Rybinski, H. (2019) "Clustering of semantically enriched short texts." *J Intell Inf Syst* **53**, 69–92.

[2] Trueman, T. E., Kumar, A., Narayanasamy, P., & Vidya, J. (2021). "Attention-based C-BiLSTM for fake news detection." *Applied Soft Computing*, pp 1-6.

[3] Sun, F., & Chu, N. (2020). "Text Sentiment Analysis Based on CNN-BiLSTM-Attention Model." *In 2020 International Conference on Robots & Intelligent System (ICRIS)* pp: 749-752.

[4] Liu, Sisi & Lee, Kyungmie & Lee, Ickjai. (2020). "Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation." *Knowledge-Based Systems*. 197.

[5] Jianfeng Deng, Lianglun Cheng, Zhuowei Wang, (2021) Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification, Computer Speech & Language, Volume 68.

[6] Li, W., Gao, S., Zhou, H., Huang, Z., Zhang, K., & Li, W. (2019). "The automatic text classification method based on bert and feature union." *In IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)* pp. 774-777.

[7] Mengjia Fan, Yangsen Zhang and Jiayuan Li, (2015) "Word similarity computation based on HowNet," *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1487-1492.

[8] S. Sharma, S. S. Panigrahi, B. Paul and N. Panigrahi, (2018) "Detection of Topic from Unstructured Text With Mixed Languages," *International Conference on Information Technology (ICIT)*, pp. 151-154.

[9] Z. H. Kilimci and S. Akyokuş, (2019) "The Evaluation of Word Embedding Models and Deep Learning Algorithms for Turkish Text Classification," *4th International Conference on ComputerScience and Engineering (UBMK)*, pp. 548-553.

[10] Q. Chen, L. Yao and J. Yang, (2016) "Short text classification based on LDA topic model," *International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 749-753.

[11] S. Manna and O. Phongpanangam,(2018) "Exploring Topic Models on Short Texts: A Case Study with Crisis Data," *Second IEEE International Conference on Robotic Computing (IRC)*, Laguna Hills, pp. 377-382.

[12] A. Anantharaman, A. Jadiya, C. T. S. Siri, B. N. Adikar and B. Mohan, (2019) "Performance Evaluation of Topic Modeling Algorithms for Text Classification," *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, pp. 704-708.

[13] Zuo, S.-Z & Wu, C.-H & Zhou, Y.-Q & He, H.-C. (2006). "Chinese short-text categorization based on the key classification dictionary words." **13:** 47-49.

[14] G. Veena, S. Athulya, S. Shaji and D. Gupta, (2017) "A graph-based relation extraction method for question answering system," *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 944-949.

[15] L. R. Pillai, V. G., and D. Gupta, (2018) "A Combined Approach Using Semantic Role Labelling and Word Sense Disambiguation for Question Generation and Answer Extraction," *Second International Conference on Advances in Electronics, Computers, and Communications (ICAECC),* pp. 1-6.

[16] V. G, D. Gupta, A. Anil and A. S, (2019) "An Ontology Driven Question Answering System for Legal Documents," *2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pp. 947-951.

[17] Liu Wensen, Cao Zewen, Wang Jun and Wang Xiaoyi, (2016) "Short text classification based on Wikipedia and Word2vec," *2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1195-1200.

[18] Olga Fuks, (2018) "Classification of News Dataset" Stanford University, ofuks@stanford.edu.

[19] J. Ge, H. Wang and Y. Fang, (2020) "Short Text Classification Method Combining Word Vector and WTTM," *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1994-1997.

[20] H. Seki and S. Toriyama, (2019) "On Term Similarity Measures for Short Text Classification," *IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA),* pp. 53-58.

[21] Ma, Y., Liu, X., Zhao, L., Liang, Y., Zhang, P., & Jin, B. (2022). "Hybrid embedding-based text representation for hierarchical multi-label text classification." *Expert Systems with Applications*.

[22] Meng, X., & Shao, J. F. (2021). "Overview of Chinese Text Classification." *In Advancements in Mechatronics and Intelligent Robotics* pp. 539-544.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, (2013) "Efficient estimation of word representations in vector space."

[24] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, (2013) "Distributed representations of words and phrases and their compositionality," CoRR.

[25] S. Hochreiter and J. Schmidhuber, (1997) "Long short-term memory," *Neural Computation* **9 (8)**: 1735–1780.

[26] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, (2000) "Learning to forget: Continual prediction with lstm," *Neural Comput.*

[27] S. Usmani and J. A. Shamsi, (2020) "News Headlines Categorization Scheme for Unlabelled Data," *International Conference on Emerging Trends in Smart Technologies (ICETST)*, pp. 1-6.

[28] K. S. Naveenkumar, R. Vinayakumar and K. P. Soman, (2019) "Amrita-CEN-SentiDB 1: Improved Twitter Dataset for Sentimental Analysis and Application of Deep learning," *10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-5.

[29] Nils Reimers and Iryna Gurevych. (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." *In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.