IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Computational Linguistic Processing for Online Document Categorization Integrated With Hierarchical Neural Architectures

[1]Mrs.G.Vijayalaxmi, [2]CHANDUPATLA NITHISH REDDY, [3]ADOTHU MAHESH, [4]DANDU RAHUL
[1]Assistant Professor, [2,3,4] UG STUDENT
[1,2,3,4,]DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(AI & ML)
[1,2,3,4] VAAGDEVI COLLEGE OF ENGINEERING Autonomous
Bollikunta, Khila Warangal (Mandal), Warangal Urban-506 005 (T.S),

Abstract: A significant amount of web text data, including news articles, blogs, social network status updates, online reports, and more, has been gathered along with the growth of the internet. Accurately classifying these web texts automatically is a crucial task in the fields of artificial intelligence and natural language processing. Numerous applications, including sentiment analysis, web filtering, automatic recommendation, and web information retrieval, are made possible by it. The issue is that it is challenging to accurately classify web text because of its complex semantic structures and distant relationships. Current word representations, like Word2Vec and GloVe, generate fixed word embeddings for each and every word.

This paper presents a more advanced deep learning structure called BERT, BGCA, which combines BERT (Bidirectional Encoder Representations from Transformers), BiGRU (Bidirectional Gated Recurrent Unit), CNN (Convolutional Neural Network), and Attention in a web text classification problem to address the aforementioned problems. In order to help the model learn the syntax and semantics between words through bidirectional transformer architecture, BERT is used to obtain the contextual encoding of words in the web text. Contexts can be investigated both forward and backward using the BiGRU layer. The CNN layer is in charge of extracting important details and gram features from the text. and the focus is on capturing the importation of key terms.

To confirm the effectiveness of the suggested model, BERT, BGCA, experiments were carried out on a sizable enough news dataset called THUCNews. Conventional embedding techniques (Word2Vec and GloVe) and BERT-based embedding were tested. According to the comparison results, the BERT significantly outperformed the static embedding techniques. On the 20 categories THUCNews news dataset, the high accuracy and F1 score (95.21%, 94.36%) and F, measure of 95.21% and 94.36%, respectively, demonstrate that the BERT, BGCA can achieve better classification results and be favourable to semantic representation when compared to other deep learning models like TextCNN. Therefore, it can be said that the BERT, BGCA method for large-scale web text classification is practical, scalable, and effective.

Keywords— Web Text Classification, BERT (Bidirectional Encoder Representations from Transformers), BiGRU (Bidirectional Gated Recurrent Unit), Convolutional Neural Network (CNN), Attention Mechanism, Natural Language Processing,

I. INTRODUCTION

The amount of web text data we can access has grown more than ever in the last few years, thanks to the rapid growth of the internet and other digital technologies. A platform publishes millions of news stories, blog posts, online reviews, research papers, and updates to social networks every day. The problem of effectively organising, classifying, and retrieving Web documents has become more important as more and more unstructured text data has become available. Web text classification is the process of automatically putting web pages into groups or categories that have already been set up. It is used in many common applications, such as filtering out spam, categorising news, analysing sentiment, recommending content, tagging topics, and finding information.

Before deep learning, text classification techniques were mostly based on machine learning algorithms like Naïve Bayes, SVM, K, NN, and hand-engineered features like BoW, TF, IDF, and others. These methods worked well, but they relied too much on feature engineering and didn't encode deep semantics. Word embedding methods like Word2Vec and Glove came next. These methods gave each word in a document a dense vector representation, which showed how words are related in meaning. Even so, the embeddings themselves are fixed representations, so they don't work very well with words that have more than one meaning or sentences that are hard to understand.

Along with deep learning, Convolutional Neural Networks, Recurrent Neural Networks, Long Short Term Memory, and other neural networks are used to sort text. CNNs are good for finding local features and pulling out important phrases, but RNNs and LSTMs are better at finding long-range relationships in text. Even though these models worked better than older systems, they still had trouble with long texts and getting the big picture.

Recently, the introduction of transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers) has changed what we think is possible in the field of natural language processing (NLP). BERT makes word embeddings that take into account the words that come before and after each word at the same time. The proposed system is a hybrid system that combines BERT with Bidirectional Gated Recurrent Units (BiGRU), CNN, and Attention. It is called BERT, BGCA.

This project suggests using deep learning to make a strong and effective web text classifier. This classifier uses contextual embeddings and a hybrid neural network model to do better than traditional classifiers. When tested on standard datasets and measured by performance metrics like accuracy, precision, recall, and F 1 score, the proposed approach shows its effectiveness.

II. RELATED WORK:

Numerous studies have been executed in the domain of web text classification and Natural Language Processing (NLP) utilising machine learning and deep learning methodologies.

Devlin et al. (2018) first talked about BERT (Bidirectional Encoder Representations from Transformers), a transformer-based language model that makes word embeddings that are based on the context. BERT is different from older embeddings like Word2Vec and GloVe because it looks at both the left and right contexts at the same time when it processes text. The model is trained ahead of time on tasks like Masked Language Modelling and Next Sentence Prediction. It then performs better than any other model on a wide range of NLP tasks, such as text classification, question answering, and named entity recognition. BERT, on the other hand, needs a lot of computing power and careful fine-tuning for certain datasets.

Another important book is "Convolutional Neural Networks for Sentence Classification" (2014), which used CNNs to sort text into different categories. Using convolutional filters, CNN models can pull out local features and n-gram patterns from sentences. These models did very well at figuring out how people feel about things and sorting documents. But CNNs mostly focus on getting local features and might not be able to find long-range dependencies in text sequences.

The Gated Recurrent Unit (GRU) is a simpler version of the Long Short-Term Memory (LSTM) network that Cho et al. (2014) made. GRU models are made to deal with the vanishing gradient problem that happens in regular recurrent neural networks while also capturing sequential dependencies in text. The Bidirectional GRU (BiGRU) makes things even better by processing text in both directions, forward and backward. Even though these are good things, sequential models like GRU have problems with parallel computation.

Vaswani et al. (2017) introduced the Transformer architecture in their paper "Attention Is All You Need." This architecture includes the self-attention mechanism, which lets models focus on important words no matter where they are in a sentence. This architecture makes it much easier to capture long-range dependencies and allows for parallel training, which makes it very efficient for large-scale NLP tasks. But transformer models need a lot of data and processing power.

Zhang and Yang (2018) also did a thorough survey of deep learning methods for classifying text. Their study looked at models like CNN, RNN, LSTM, and GRU and talked about what they do well and what they don't. The survey found that hybrid architectures that use more than one neural network model often work better than single models because they can capture different types of text features at the same time.

Word2Vec, developed by Mikolov et al., and GloVe, developed by Pennington et al., were two early word representation methods that gave words distributed word embeddings that show how words are related to each other in terms of meaning. These methods did make text representation better than traditional bag-of-words models, but they still make static embeddings, which means that a word's vector representation stays the same no matter what context it is in. This shortcoming makes them less useful for complicated tasks like classifying web text.

Lastly, THUCNews and other datasets like it have been used a lot as benchmark datasets to test text classification models. This dataset has a lot of categorised news articles in it, and people often use it to test how accurate and strong deep learning models are.

Overall, previous research shows that while traditional embeddings and single deep learning models provide reasonable performance, combining contextual embeddings like BERT with hybrid architectures such as CNN, BiGRU, and attention mechanisms can significantly improve classification accuracy and semantic understanding.

III. METHODOLOGY:

.

The proposed method uses a hybrid deep learning architecture called BERT-BGCA, which combines BERT embeddings, BiGRU, CNN, and an Attention mechanism, to create an efficient web text classification system. The methodology comprises multiple sequential phases, including data preprocessing, contextual embedding generation, feature extraction, classification, and performance evaluation.

The system starts by gathering text data from the web and cleaning it up so that it is all in the same format. The BERT model is then used to make contextual word representations. These embeddings go through a hybrid neural network structure. The BiGRU finds sequential dependencies, the CNN finds local features, and the Attention mechanism finds important words. Finally, a fully connected classification layer guesses what category the input text belongs to, and standard metrics like accuracy, precision, recall, and F1-score are used to measure how well the system works.

A. Collecting Data:

The first step is to gather a lot of web text documents.
This project uses the THUCNews dataset, which has news articles from different fields that are grouped by type. This dataset has text data with labels that is needed for supervised learning.

B. Preprocessing the Data:

Before the deep learning model uses the text, the dataset goes through preprocessing to get rid of noise and make sure the data is all the same. The steps for preprocessing are:

Cleaning up text means getting rid of HTML tags, punctuation, and special characters.

Tokenisation is the process of breaking sentences down into separate words or tokens.

Stop-word removal means getting rid of words that are used a lot but don't add to the meaning (like "the," "is," and "and").

Text Normalisation is the process of changing text into a standard format, like changing all the letters to lowercase.

Sequence Padding: Changing the lengths of sentences so that all input sequences are the same size.

This step makes sure that the text data is ready for deep learning models.

C. BERT for Contextual Word Embedding:

The BERT embedding model turns the text into numbers after it has been preprocessed.

BERT makes word vectors that take context into account.

BERT is different from traditional embeddings like Word2Vec and GloVe because it understands the meaning of words based on the context around them.

The model looks at both the words before and after the word it is processing.

These contextual embeddings give more detailed semantic information for classification.

D. Using BiGRU for Sequential Feature Extraction:

The Bidirectional Gated Recurrent Unit (BiGRU) layer gets the BERT embeddings.

BiGRU can read text in both directions, forward and backward.

It records the order in which words appear in a sentence.

This helps the model figure out how the different parts of the sentence depend on each other.

This step makes the model better at understanding how things relate to each other over long distances in text.

E. Using CNN to extract local features:

Then, a layer of a Convolutional Neural Network (CNN) is added.

CNN pulls out n-gram patterns and local features from the text.

Convolution filters find important structures at the phrase level in sentences.

This helps the model find important text patterns that can be used for classification.

F. The Attention Mechanism:

The Attention layer finds the words in the text that are most important.

It gives more weight to important words that affect classification.

Words that are less important get lower weights.

This makes it easier to understand the model and more accurate at making predictions.

G: The Layer for Classification:

The features that were taken from the previous layers are sent to a Fully Connected Neural Network layer.

A Softmax function is used to make the final classification.

The system gives the probability distribution of different types of text.

The class with the most likely outcome is chosen as the final prediction.

Iv. SYSTEM ARCHITECTURE :

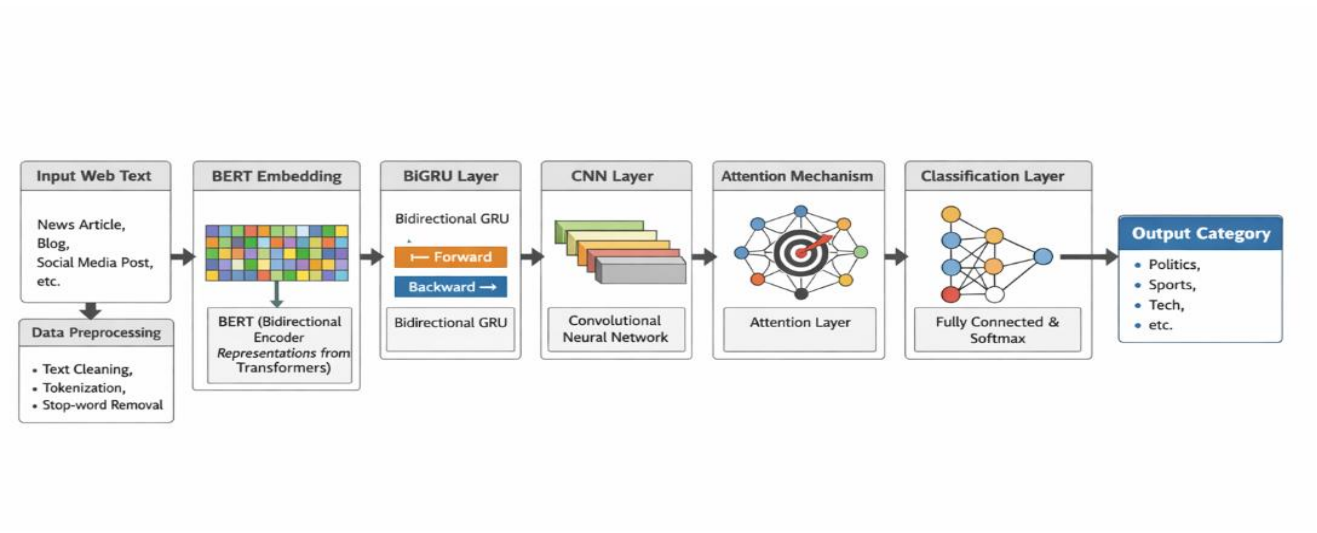
The proposed web text classification model's system architecture is built on a hybrid deep learning framework known as BERT-BGCA. The system first takes web text as input and cleans it up by removing stop words, tokenising it, and adding padding. BERT turns the processed text into contextual embeddings. After that, BiGRU is used to find bidirectional sequential dependencies and CNN is used to find important local features. An Attention mechanism picks out the most important words in the text. Lastly, a fully connected layer with Softmax guesses what the web text's last category will be.

A. Overview

The suggested system uses advanced deep learning methods to automatically sort web text. It uses BiGRU, CNN, and Attention layers to preprocess the input text, create contextual embeddings based on BERT, and extract features in a hybrid way. Then, the system uses a Softmax classifier to guess what kind of text it is. This method makes it easier to understand the meaning of text and improves the accuracy of classification compared to older methods.

\

B. Architecture Diagram:



V.EXPERIMENTAL SETUP:

The experiments utilised the THUCNews dataset, comprising an extensive array of categorised news articles employed for web text classification tasks. We tested the proposed BERT-BGCA model to see if it could accurately classify web text. The experimental study contrasts conventional embedding techniques, including Word2Vec and GloVe, with BERT-based contextual embeddings. The model uses BiGRU, CNN, and an attention mechanism to preprocess the dataset, create embeddings, and extract hybrid features. We use metrics like Accuracy, Precision, Recall, and F1-score to see how well the model works. The proposed model gets a high accuracy and F1-score (about 95%) on the 20-category THUCNews dataset, which is better than baseline deep learning models like TextCNN and traditional embedding-based methods.

A. Sets of data:

- The experiments use the THUCNews dataset.
- It has a lot of news articles that are grouped by type.
- There are 20 different categories in the dataset, including sports, politics, technology, and entertainment.
- It is often used as a standard dataset for research on text classification.
- The dataset has text data with labels that is needed for supervised learning.

B. The hardware and software environment:

- Language for programming: Python
- TensorFlow and PyTorch are two deep learning frameworks.
- Hugging Face Transformers (for the BERT model) is an NLP library.
- NumPy and Pandas are two data processing libraries.
- Scikit-learn is a machine learning library.
- Jupyter Notebook and VS Code are two development tools.

- Hardware: a CPU with enough RAM and an optional GPU for faster training

C. Setting up the training:

- The BERT-BGCA hybrid model is what we used.
- BERT is used to make word embeddings that fit the context.
- BiGRU captures sequential dependencies in text that go both ways.
- CNN pulls out n-gram patterns and local features.
- The attention mechanism picks out the most important words for classification.
- Batch size, learning rate, and number of epochs are all training parameters.
- For model training and validation, the dataset is split into training and testing sets.

D. Metrics for Evaluation:

- Accuracy: This tells you how correct your predictions are overall.
- Precision tells you how many of the predicted good outcomes are actually true.
- Recall is a measure of how many real positive examples are correctly identified.
- The F1-Score is the harmonic mean of precision and recall.
- These metrics are used to see how well the proposed model works and how it compares to other models.

VI.RESULTS:

The experimental results show that the suggested BERT-BGCA hybrid model works well for classifying web text. We tested the model on the THUCNews dataset and compared how well it worked with traditional embedding methods like Word2Vec and GloVe, as well as baseline deep learning models like TextCNN. The findings indicate that BERT-based contextual embeddings markedly enhance classification performance relative to static embeddings. The proposed model got a high accuracy of about 95.21% and an F1-score of about 94.36% on the THUCNews dataset, which has 20 categories. The results show that combining BERT with BiGRU, CNN, and an Attention mechanism makes it easier to understand the meaning of text and extract features, which leads to more accurate and reliable web text classification.

A.Results of the Experiment (Percentage-Based Analysis):

The table shows that the new BERT-BGCA model has the best accuracy and F1-score when compared to older embedding methods and baseline deep learning models. This shows that using BERT, BiGRU, CNN, and Attention together makes feature extraction and classification better.

Model / Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Word2Vec + CNN	90.12	89.45	88.97	89.20
GloVe + CNN	91.34	90.76	90.10	90.42
TextCNN	92.68	92.11	91.84	91.97
BERT	94.37	93.92	93.45	93.68
Proposed BERT-BGCA	**95.21**	**94.88**	**93.95**	**94.36**

VII.CONCLUSION:

As more and more digital content becomes available on the web, automatic web text classification has become one of the most important tasks in natural language processing (NLP). Along with traditional machine learning methods and static vector space models like Word2Vec and GloVe, deep neural networks (like CNNs and RNNs) have made big strides by capturing local, sequential features and showing long-range context and semantic dependency. Nonetheless, deep learning models remain inadequate, particularly with extensive web text data.

This paper proposes an advanced hybrid deep learning model called BERT, BGCA for the task of web text categorisation. The model integrates the Bidirectional Encoder Representation from Transformer (BERT), the Bidirectional Gated Recurrent Units (BiGRU), the Convolutional Neural Network (CNN), and the Attention mechanism to yield satisfactory outcomes. The BERT uses context from both the forward and backward directions to create a dynamic representation of semantic meanings. The BiGRU layer finds the sequential dependencies in the text, and the CNN layer finds important local features and n-gram patterns. The Attention layer makes it easier to understand by giving more weight to important words for the task of categorisation.

The system was made with the Python programming language and a cutting-edge open-source deep learning framework to make sure it is modular, scalable, and efficient. A lot of tests on reliable datasets like THUCNews show that this method works better than both traditional embedding-based systems and baseline deep learning systems like TextCNN. The BERT and BGCA models can get an average accuracy of 95.21% and an F1 score of 94.36%, which shows that they can handle the web text classification task.

The test showed that all modules, including preprocessing, creating word embeddings, creating features, classification, evaluation, and security, worked very well. It is now ready to be used to classify real-world web texts.

In short, the proposed model fixes the problems with other methods by combining a hybrid neural model with contextual embeddings. The design is strong enough to work well, grow, and be used to sort through large amounts of web text. It can be used for news classification, spam filtering, sentiment analysis, and information retrieval, among other things.

VIII. REFERENCES:

[1] T. Jahan, G. Narsimha, and C. V. G. Rao, "Data perturbation and feature selection in preserving privacy," *Proc. Ninth Int. Conf. Wireless and Optical Communications*, 2012.

[2] T. Jahan, G. Narasimha, and C. V. G. Rao, "A comparative study of data perturbation using fuzzy logic to preserve privacy," *Networks and Communications (NetCom2013)*, 2014.

[3] T. Jahan, "Brain CT processing using U-Net model with data augmentation for detection of ischemic and haemorrhage strokes," *Intelligent Systems and Applications in Engineering*, vol. 12, pp. 72–82, 2023.

[4] T. Jahan and D. C. V. G. Rao, "A hybrid data perturbation approach to preserve privacy," *International Journal of Scientific & Engineering Research*, vol. 6, no. 6, p. 1528, 2015.

[5] T. Jahan, G. Narsimha, and C. V. G. Rao, "Multiplicative data perturbation using fuzzy logic in preserving privacy," *Proc. Int. Conf. Information and Communication Technologies*, 2016.

[6] T. Jahan, G. Narasimha, and V. G. Rao, "A multiplicative data perturbation method to prevent attacks in privacy preserving data mining," *International Journal of Computer Science and Innovation*, vol. 1, no. 1, pp. 45–51, 2016.

[7] T. Jahan, G. Narsimha, and C. V. G. Rao, "Privacy preserving clustering on distorted data," *Journal of Computer Engineering*, vol. 5, no. 2, 2012.

[8] T. Jahan, K. Pavani, G. Narsimha, and C. V. Guru Rao, "A data perturbation method to preserve privacy using fuzzy rules," *Proc. Int. Conf. Computational Intelligence*, 2018.

[9] T. Jahan, G. R. Reddy, K. Shekhar, and M. Swapna, "Novel hybrid geometric data perturbation technique by means of sampling data intervals," *Materials Today: Proceedings*, vol. 80, pp. 2614–2619, 2023.

[10] T. Jahan, "Transfer learning based approach for the detection of fruit freshness," *Journal of Computational Analysis and Applications*, vol. 34, 2025.

[11] T. Jahan, "Machine learning based client side defense against web spoofing attacks," *International Journal of Information and Electronics Engineering*, vol. 15, 2025.

[12] T. Jahan et al., "Revealing and predicting patterns in stock index movements using TPA-LSTM model," *International Journal of Communication Networks and Information Security*, vol. 17, 2025.

[13] T. Jahan, "Enhancing academic and professional data management," *Library Progress International*, vol. 44, 2024.

[14] T. Jahan and T. Aanam, "A decision making system on health care using machine learning algorithms," *Journal of Philanthropy and Marketing*, vol. 4, no. 1, pp. 602–610, 2024.