An Automated Text Document Classification Framework using BERT

Momna Ali Shah[1], Muhammad Javed Iqbal[2], Neelum Noreen[3], Iftikhar Ahmed[4]
Department of Computer Sciences, UET Taxila, Pakistan[1, 2]
Department of Computer and Information Sciences, Gulf Colleges, Hafr Al Batin, Saudi Arabia[3]
Department of Information Technology, King Abdul Aziz University, Jeddah, Saudi Arabia[4]

***Abstract*—Due to the rapid advancement of technology, the volume of online text data from numerous various disciplines is increasing significantly over time. Therefore, more work is needed to create systems that can effectively classify text data in accordance with its content, facilitating processing and the extraction of crucial information. Since these non-automated systems use manual feature extraction and classification, which is error-prone and time-consuming by choosing the best appropriate algorithms for feature extraction and classification, traditional procedures are typically resource intensive (computational, human, etc.), which is not a viable solution. To address the shortcomings of traditional approaches, we offer a unique text categorization strategy based on a well-known DL algorithm called BERT. The proposed framework is trained and tested using cutting-edge text datasets, such as the UCI email dataset, which includes spam and non-spam emails, and the BBC News dataset, which includes multiple categories such as tech, sports, politics, business, and entertainment. The system achieved the highest accuracy of 91.4% and can be used by different organizations to classify text-based data with a high performance. The effectiveness of the proposed framework is evaluated using multiple evaluation metrics such as Accuracy, Precision, and Recall.**

***Keywords*—Deep learning; text classification; BERT**

I. INTRODUCTION

Text classification is a common problem in Natural Language Processing (NLP) that aims to classify the text data based on its content. This field has become drastically important due to increase in text based data. The increase in internet usage has resulted in the creation of diversified text data that is made available by numerous social media platforms and websites in different languages. This has resulted in exponential rise in the number of complex documents and texts that demand a deeper understanding of machine learning approaches to effectively identify texts in numerous applications. This field has wide range of applications such as sentiment analysis, email classification, news classification, movie review prediction, etc. [1, 2].

In NLP, numerous ML techniques have been developed over the past few years. A typical text classification system has four stages: preprocessing, feature extraction, feature selection, and classification. These applications must solve a number of issues relating to the nature and organization of the underlying textual information by condensing word variants into short representations while retaining the majority of the linguistic properties. However, there are certain limitations in the traditional methods. Firstly, it is difficult to capture text semantics using these techniques since they solely focus on word frequency attributes and completely ignore the contextual information stored in text. Second, the success of these statistical approaches in machine learning is often dependent on hand-crafted feature extraction and classification, which is time-consuming and error-prone. Moreover, it can be difficult for researchers to develop such pipelines and methods for text classification that can perform better [3, 4].

Hence, due to these problems, recent years have seen a complete shift from these traditional text classification methods towards much stronger state-of-the-art DL based methods. These algorithms do not require a feature extraction phase prior to data classification, as these systems are completely automated because these models are highly capable of extracting robust features from the dataset themselves during the learning phase. Due to which, the deep learning algorithms have achieved state-of-the-art performance in a variety of NLP tasks, hence, the researchers are keen in exploring the applicability of these algorithms in different tasks like question/answering, email classification, news categorization and much more [5, 6].

In this paper, we proposed a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) architecture for text classification. BERT is a brand-new language representation model that Google has introduced in 2018 [6, 7]. The model has succeeded in achieving state-of-the-art performance on text classification problems, hence, has increased the interest of researchers in fine-tuning and deployment of BERT on various text classification problems. In this paper, we fine-tuned BERT architecture on two state-of-the art datasets composed of email and news. The proposed framework is discussed in detail in Section III. The contributions of the proposed study are as follows:

- We developed an automated text classification framework to classify different types of text data.
- The proposed method initially preprocesses data by removing stop-words and extra characters so that classification performance can be improved.
- The preprocessed text is then classified via widely known DL based architecture called BERT by fine-tuning the architecture on our problem.

- The performance of the proposed technique using different evaluation parameters and compare its performance with existing systems.
- In this study, we performed extensive experiments on publically available datasets to show the efficiency and robustness of our algorithm.
- Both of the proposed methods can accurately detect and classify text data effectively and can be deployed by various organizations to classify the text data.

The remaining paper is organized as follows. The literature is critically analyzed in Section II. The proposed methodology is discussed in detail in Section III. Whereas, Section IV evaluates the performance of the proposed technique and compares it with state-of-the-art methods. The study is concluded in Section V.

II. Literature Review

Text classification is a common NLP task that has a wide range of uses, such as sentiment analysis, email classification, detection of offensive language, spam filtering, etc. Now-a-days ML has become a subject of interest for text classification tasks, as these algorithms have shown considerable potential for acquiring linguistic knowledge.

A typical text classification system has four stages: preprocessing, feature extraction, feature selection, and classification. These applications must address a number of issues relating to the nature and structure of the underlying textual information for languages by translating word variants into compact representations while retaining the majority of the linguistic properties. However, these systems have several issues. Firstly, it is difficult to capture text semantics using these techniques since they solely focus on word frequency attributes and completely ignore the contextual structure information in text. Second, the effectiveness of these statistical methods for machine learning frequently depends on challenging technical features and the usage of vast linguistic resources.

The authors in Jang et al. [8] employed MLP to classify textual data. The authors succeeded in achieving 71% accuracy on MLP. However, the performance should be improved. She et al. [3] proposed a hybrid technique that solves CNN's fundamental limitation in expressing long-term contextual information while utilizing CNN's capacity to extract local data. Additionally, the model makes an effort to address LSTM's inherent flaws, which include its tendency to process data sequentially and rank as the poorest feature extractor. When compared to counterpart models, the hybrid model performed better, but its findings lagged below models that make use of an attention mechanism in terms of interest.

Urdu editorials were also classified by Sattar et al. [1] using NB. The authors reduced the dimensionality by eliminating terms with common frequency. With their study, they were able to prove that when Naive Bayes classifier is supplied text with frequent terms it outperforms the model when it isn't supplied those terms. However, these studies need to be incorporated on multiple Urdu categories rather than only headline classification. The authors in Antoun, et al. [7] used BERT model for Arabic text classification called Arabert which was trained on 24 gigabytes of data. Similarly, in another research, Abdul-Mageed, et al. [4] trained Arabic BERT architecture called MARBERT on 1B tweets. However, the systems mentioned in [4, 7] are computationally expensive.

Koswari et al. [5] proposed an ensemble approach using deep learning algorithms to classify text from news dataset and achieved 87% accuracy. However, the system obtained a low overall performance, hence, its accuracy should be improved. Cai et al. [9] classified news data by employing several deep learning architectures such as RCNN, CNN and RNN. Similarly, the study presented by Lenc et al. [6] proposed the use of CNNs as well as a simple multi-layer perceptron to extract features from Czech newspaper documents before applying multi-label document classification. This technique achieved F1 score of 0.84 using MLP with sigmoid functions. However, these studies only classify news data, hence need to test their architectures on other types of text data before deploying in real-world scenario.

III. Materials and Methods

Due to its vast applicability in businesses and organizations, text classification has become a very significant research area in NLP. The text classification algorithms aim to classify the text data based on its content and meta-data contained in it. This can be achieved by using ML and DL based algorithms to automate the process with an increase in data volumes. In this paper, we propose a novel and robust text classification framework employing a well-known DL based algorithm called BERT. The pipeline of proposed architecture is illustrated in Fig. 1.

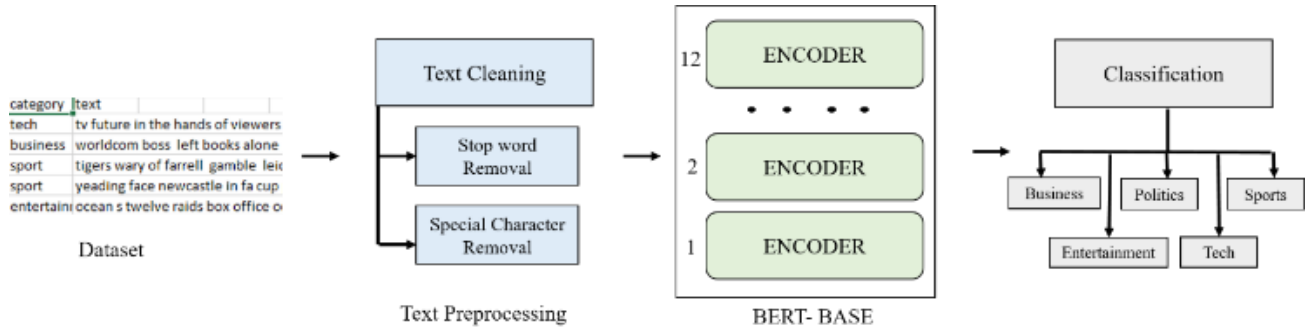

Fig. 1. Pipeline of proposed framework.

The proposed architecture is trained and evaluated on two publically available datasets. Initially, we cleaned the datasets by removing stop-words and special characters. Furthermore, we also converted the entire text in small case before actual processing. The cleaned dataset is then supplied to BERT architecture for feature extraction and classification.

A. Data Collection

The proposed framework is trained and evaluated on publically available datasets obtained from different sources. The BBC News dataset consists of a total of 2225 documents that consists of five classes namely business, entertainment, politics, sport, tech that was obtained from 2004-2005 [10]. Fig. 2 shows the dataset distribution showing the class name and number of documents in that class.

The second dataset is gathered from UCI Database [11]. We also performed exploratory data analysis of the second dataset. The database is composed of 5569 emails, of which 745 are spam and others are non-spam. Hence, non-spam emails count for 12% of the dataset and spam emails count for 88% of the whole database. The dataset is highly imbalanced, hence, in these recall and precision as an evaluation metrics are very useful. However, it may be noted that before supplying the database to proposed BERT architecture, we balanced the dataset and randomly chose equal numbers of instances from both classes to avoid the biasness in the classification architecture.

B. Data Preparation

Data cleaning is an essential phase in any NLP task, which aims to modify data in a format that is much easier for the algorithm to analyze or predict. In this phase, we cleaned the dataset by removing special characters and stop words. Special characters and symbols consist of non-alphabet letters such as "([/ (] [| @],]". Whereas, stop-words are a group of terms that are used frequently in a sentence or used to link sentences, some of these include "a," "the," "is," and "are." These terms need to be eliminated because they provide no information to the model but can be a cause of poor performance of any text classification model. Furthermore, the entire dataset is also converted in small case to help remove any ambiguity during learning process. Fig. 3 shows the preprocessing steps applied to the datasets.

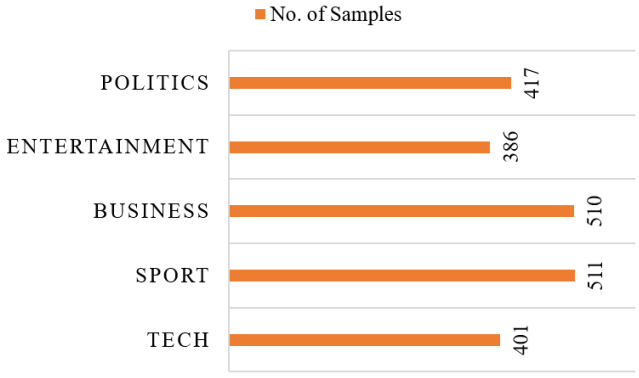

Fig. 2. No. of text samples in BBC news dataset.

(a) Last Star Wars, not for children, the sixth and final Star Wars movie may not be suitable for young children, film-maker George Lucas has said. He told us, TV show 60 minutes

(b) last star wars not for children the sixth and final star wars movie may not be suitable for young children film maker george lucas has said he told us tv show 60 minutes

Fig. 3. Dataset preparation, (a) Non-preprocessed text, (b) Preprocessed text.

C. Proposed Framework Design

The field of NLP focuses on developing computing methods to automatically interpret and represent human language. For a very long period, the bulk of approaches to examine NLP issues relied on labor-intensive, hand-crafted features and shallow machine learning models. As a result of linguistic information being represented via sparse representations, issues like the curse of dimensionality began to arise due to high-dimensional feature vectors. However, these issues in the traditional methodologies have been solved, thanks to advent of DL based algorithms such as Convolutional Neural Networks, Recurrent Neural Networks, etc. [12]. But, one of the major issues faced in DL architectures is lack of training data. The majority of task-specific datasets only contain some human-labeled training samples because NLP is a diverse area with numerous separate jobs. Modern DL-based NLP models, on the other hand, have improved on larger volumes of data containing millions, or billions, of annotated instances. Over the past decade, researchers have created a number of methods for training general purpose language representation models using the huge volume of content from the web in order to close this data gap. The models trained on massive datasets can now be utilized on smaller problem such as question/answering or sentiment analysis, etc. rather than training models from scratch [2].

BERT, proposed in 2018 by Google AI Language researchers created quite a stir in the ML community as it achieved good results in a wide range of NLP tasks. The framework is intended to assist computers in understanding the meaning of ambiguous words in textual data by establishing context through the use of surrounding material [13, 14]. The architecture of BERT is built using Transformers, where each output element is coupled to each input element and the weights between them are dynamically calculated based on their connection. Earlier language models could only read text input in one of two directions i.e. either left to right or from right to left, but not both simultaneously. However, BERT can read data simultaneously in both directions mainly due to transformers that help its enhanced understanding of linguistic ambiguity and context. Furthermore, earlier approaches like word2vec would map every word to a vector, which only captures a small fraction of its meaning in one dimension which is known as word embedding. But BERT is the first NLP technique that completely relies on self-attention techniques because of the bidirectional Transformers at its core that helps it understand complete meaning as the paragraph develops [15]. This capability of directionality enables the BERT to eliminate the left-to-right momentum due to which

the words are usually biased towards a particular meaning as a phrase proceeds, hence reading from both directions, accounts for the impact of all other words on the focus word, and compensates for the augmented meaning [13].

In this paper, a refined BERT base architecture is suggested (shown in Fig. 4) for the text classification problem. BERT-base has 110 parameters and was trained on an English language corpus. BERT-base contains 12 encoders layered on top of each other. BERT-Base features a larger feed forward network with 768 hidden units. In addition, the structure contains 12 attention heads. The system gets computationally expensive as the number of encoders and parameters rises. For these reasons, we chose the BERT-base model because it is lightweight and quick to train.

D. Experimental Configuration and Setup

We trained and evaluated our proposed BERT architecture on publically available datasets i.e. UCI Email dataset composed of Spam and Non-Spam emails. The second dataset consists of BBC News text dataset composed of 5 different classes namely tech, entertainment, sport, politics and business. The model is tested on different hyper-parameters and performed the best on 10 epochs, mini-batch size of 32, a learning rate of 0.001 and a dropout rate of 0.1 (meaning 10% of the random nodes are dropped during training process to lighten up the network). In this study, 75% of the dataset is used for training the model, whereas 25% of the dataset is used for testing purposes. The entire experiment is performed on Python using Anaconda software on a PC with 8GB RAM and Intel Core i5 processor.

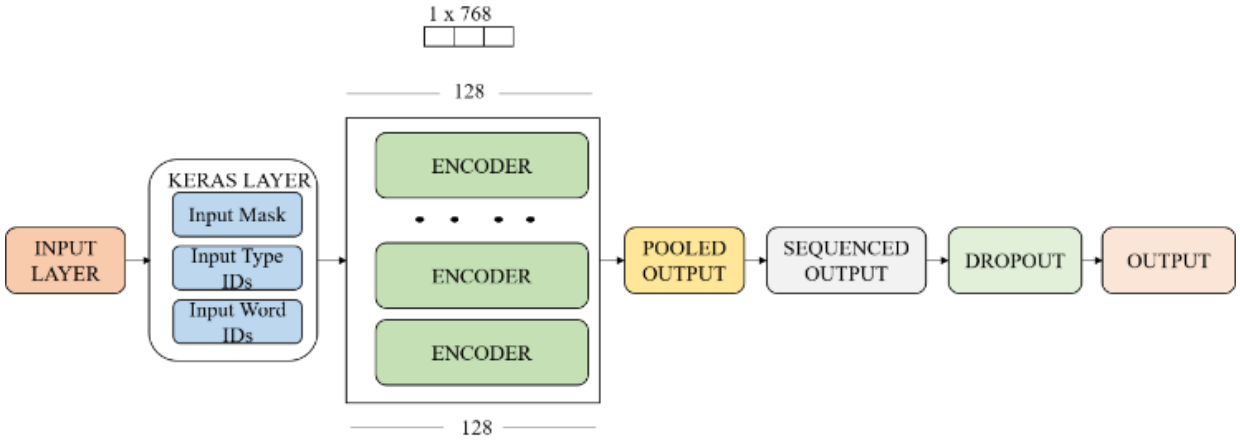

Fig. 4. Proposed framework architecture.

IV. Proposed Method Results

A. Evaluation Parameters

The proposed method is evaluated using different metrics such as precision, recall and accuracy. Confusion matrices help in showing tabular counts of observed and expected values. Different evaluation matrices such as True Positives, True Negatives, False Positives and False Negatives can be calculated using confusion matrices as well. TN depicts the total number of negative cases that were correctly identified. Similar to this, TP denotes the accurately identified positive cases. FP shows the negative cases that were by mistake classified as positive, while FN shows the positive instances wrongly classified as a negative [16, 17]. The value for accuracy, precision and recall can be calculated from the following equations.

$$ACC = \frac{TN+TP}{TP+FN+TN+FP} \quad (1)$$

$$REC = \frac{TP}{FN+TP} \quad (2)$$

$$PRE = \frac{TP}{FP+TP} \quad (3)$$

B. Experiment # 01: Classification of Emails using BERT

In this study, we employed BERT on email dataset containing both spam and non-spam emails. We initially cleaned the dataset before feeding it to BERT architecture. In this study, we employed cased BERT architecture so we changed the text to smaller case and then removed the stop words, keywords, etc.

The proposed method achieved a training accuracy of 92.3% whereas values obtained from precision, recall and f1score are 0.92 and 0.91 respectively as shown in Fig. 5. Whereas, the system achieved testing accuracy, precision and recall of 91.2%, 0.91 and 0.91 respectively as shown in Fig. 6. The confusion matrix of the proposed technique is shown in Fig. 7.

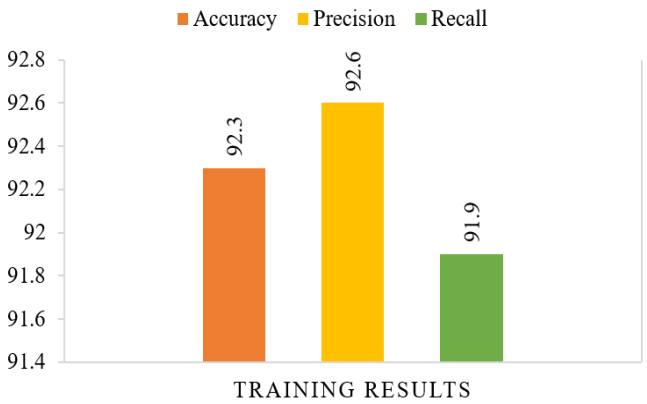

Fig. 5. Training results on UCI email dataset.

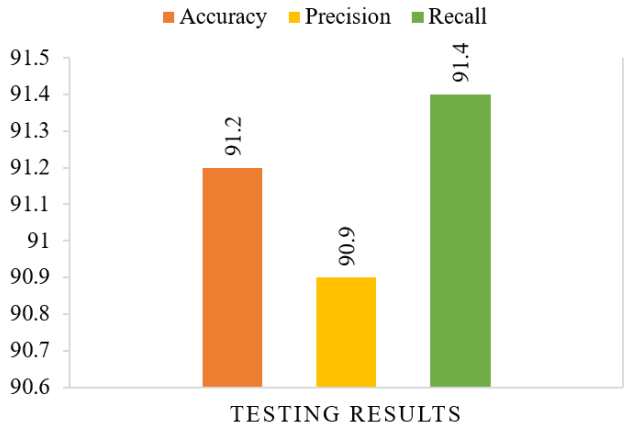

Fig. 6. Testing results on UCI email dataset.

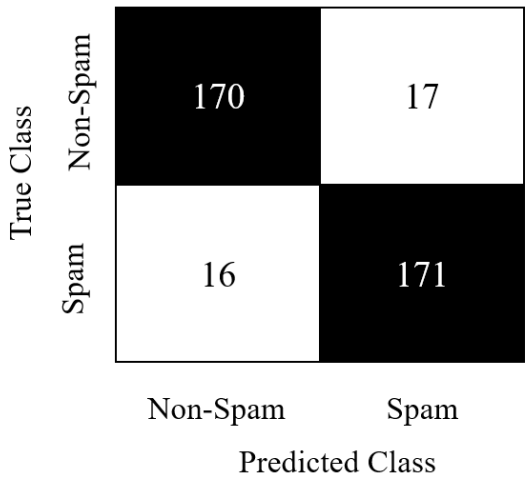

Fig. 7. Confusion matrix obtained from the proposed method.

The proposed approach is evaluated on a publicly available dataset comprised of spam and non-spam emails. We trained the BERT architecture by testing out various hyper-parameters and reached the final conclusion on 10 epochs, 32 mini-batch size and Adam optimizer. The different tested hyper-parameters are shown in Table I. The optimal values are also highlighted in the table.

TABLE I. HYPER-PARAMETER OPTIMIZATION ON UCI EMAIL DATASET

Hyper-parameter	Value/s
Epochs	4, 8, **10**
Mini-batch	8, 16, **32**
Learning Rate	0.1, 0.01, **0.001**
Optimizer	RMSProp, **Adam**

C. Experiment # 02: Classification of News using BERT

In this section, we discuss the results obtained from the proposed BERT architecture on BBC News dataset. The dataset is composed of five different news categories such as tech, entertainment, sport, politics and business. The proposed method achieved a training accuracy of 89.1% and a testing accuracy of 88.8%. The confusion matrix of the proposed technique is illustrated in Fig. 8. We also evaluated the performance of our proposed framework on precision and recall obtained from the confusion matrix. Training scores of precision and recall are 0.66 and 0.90 respectively as shown in Fig. 9, on the other hand, testing scores for precision and recall are 0.66 and 0.89 respectively as shown in Fig. 10.

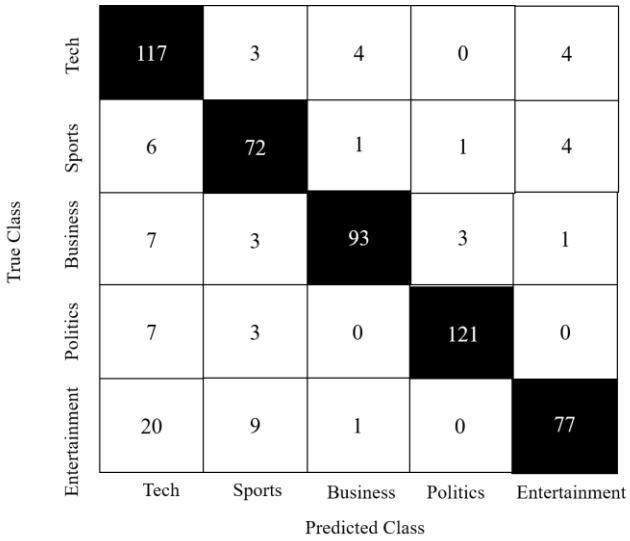

Fig. 8. Confusion matrix obtained on BBC news dataset.

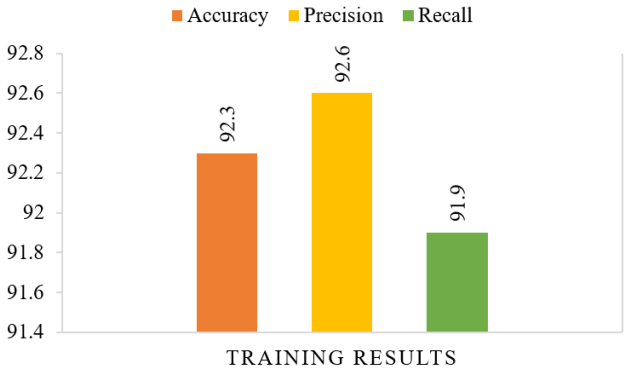

Fig. 9. Training results in terms of accuracy, precision and recall on BBC news database.

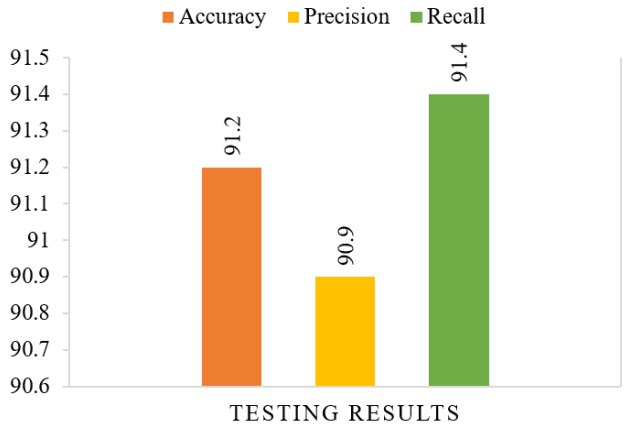

Fig. 10. Testing results on BBC news dataset.

In this experiment, we tested our different hyper-parameter settings to train the BERT architecture before reaching the final conclusion. The final parameters are 10 epochs, 32 mini-batch size and adam optimizer. The different tested hyper-parameters are shown in Table. II Moreover, the optimal values are also highlighted in the table.

TABLE II. HYPER-PARAMETER OPTIMIZATION ON BBC NEWS DATASET

Hyper-parameter	Value/s
Epochs	4, 8, **10**
Mini-batch	8, 16, **32**
Optimizer	RMSProp, **Adam**
Learning Rate	0.1, 0.01, **0.001**

D. Comparison with Existing Systems

One of the very common problems in NLP is text classification that aims to classify text data according to its contents. With the emergence of ML and DL based approaches, the researchers are keen to explore the results of these algorithms to solve this classification problem. However, most of the systems have certain limitations such as poor accuracy, use of single datasets or no data preparation prior to classification. Hence, there is a need to develop a robust and efficient system that can classify text data based on its content accurately.

Hence, in this thesis, we propose a novel and robust text classification method employing one of the very famous DL architecture known as BERT. The proposed method is trained and evaluated on publically available datasets and achieved the 91% and 89% accuracy on different datasets. Since accuracy as a single metric is not sufficient to assess the performance of a classification system, hence, we also evaluated the performance of our proposed strategy using other evaluation parameters namely precision and recall. The results prove the efficacy and robustness of the proposed technique and our devised framework can be deployed by organizations to classify text data. The comparison of our proposed framework with existing methods is described in Table III.

TABLE III. COMPARISON OF PROPOSED METHOD WITH EXISTING SYSTEMS

Reference	Technique	Result/s
Pappagari et al. [18]	RoBERTa & CNN	ACC= 84.7% & 86%
Briskilal et al. [19]	BERT	ACC= 85%
Jang et al. [8]	MLP	ACC= 71%
Semberecki et al. [20]	LSTM	ACC= 86.2%
Lenc et al. [6]	MLP	F1-Score=0.84
Proposed Method	**BERT**	**ACC=91.4%**

V. CONCLUSION AND FUTURE WORK

With the increase in data volumes, automatic text classification has become a necessity for organizations and businesses. The automated systems help them improve their performances overtime and save a lot of time and resources compared to manual systems. This has resulted in increased interest of researchers in this domain of NLP. In this thesis, we propose a novel and completely automated text classification technique employing DL frameworks. The proposed framework uses a fine-tuned BERT architecture to classify text data based on its content. The architecture proposed in this study is case sensitive, hence, the text is preprocessed by changing it in small case. Moreover, additional keywords and stop words are also removed because they can result in poor overall performance.

The preprocessed text data is then fed to fine-tuned BERT architecture for classification. The proposed technique is trained and evaluated on publically available text datasets i.e. BBC News Dataset and UCI Email dataset. The proposed technique achieved accuracy of 91.4% on UCI Email database and 89.1% on BBC News Dataset. We also compared the proposed system's performance with existing techniques. The results prove the efficiency and robustness of our method. Hence, it can be deployed in businesses to reduce the workload of manual text classification that will save time and energy required in the manual procedure. In the future, we would like to explore text data in various other languages and also explore other DL architectures.

ACKNOWLEDGMENTS

This research work was funded by Postgraduate Studies and Scientific Research, Gulf Colleges, Hafr Al Batin, Saudi Arabia. The authors acknowledge technical and financial support from Gulf Colleges, Hafr Al Batin, Saudi Arabia.

REFERENCES

[1] S. A. Sattar, S. Hina, N. Khursheed, A. J. I. J. o. S. Hamid, and Technology, "Urdu documents classification using naïve bayes," vol. 10, p. 29, 2017.

[2] J. Devlin and M.-W. Chang. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Available: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

[3] X. She and D. Zhang, "Text classification based on hybrid CNN-LSTM hybrid model," in 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 2018, vol. 2, pp. 185-189: IEEE.

[4] M. Abdul-Mageed, A. Elmadany, and E. M. B. J. a. p. a. Nagoudi, "ARBERT & MARBERT: deep bidirectional transformers for Arabic," 2020.

[5] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Rmdl: Random multimodel deep learning for classification," in Proceedings of the 2nd international conference on information system and data mining, 2018, pp. 19-28.

[6] L. Lenc and P. Král, "Deep neural networks for Czech multi-label document classification," in International Conference on Intelligent Text Processing and Computational Linguistics, 2016, pp. 460-471: Springer.

[7] W. Antoun, F. Baly, and H. J. a. p. a. Hajj, "Arabert: Transformer-based model for arabic language understanding," 2020.

[8] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, and J. W. J. A. S. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," vol. 10, no. 17, p. 5841, 2020.

[9] J. Cai, J. Li, W. Li, and J. Wang, "Deeplearning model used in text classification," in 2018 15th international computer conference on wavelet active media technology and information processing (ICCWAMTIP), 2018, pp. 123-126: IEEE.

[10] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 377-384.

[11] C. Kaul, S. Manandhar, and N. Pears, "Focusnet: An attention-based fully convolutional network for medical image segmentation," in 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), 2019, pp. 455-458: IEEE.

[12] Elvis. Deep Learning for NLP: An Overview of Recent Trends. Available: https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d.

[13] B. Lutkevich. BERT language model. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.

[14] R. Horev. (2018). BERT Explained: State of the art language model for NLP. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270?gi=928a1e3b0ac9.

[15] I. Tenney, D. Das, and E. J. a. p. a. Pavlick, "BERT rediscovers the classical NLP pipeline," 2019.

[16] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in data democracy: Elsevier, 2020, pp. 83-106.

[17] A. J. I. J. o. R. S. Hay, "The derivation of global estimates from a confusion matrix," vol. 9, no. 8, pp. 1395-1398, 1988.

[18] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical transformers for long document classification," in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 838-844: IEEE.

[19] J. Briskilal, C. J. I. P. Subalalitha, and Management, "An ensemble model for classifying idioms and literal texts using BERT and RoBERTa," vol. 59, no. 1, p. 102756, 2022.

[20] P. Semberecki and H. Maciejewski, "Deep learning methods for subject text classification of articles," in 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), 2017, pp. 357-360: IEEE.