Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 281–284
DOI: 10.15439/2023F3561
ISSN 2300-5963 ACSIS, Vol. 35

BERT-CLSTM model for the classification of Moroccan commercial courts verdicts

Taoufiq El Moussaoui
0000-0003-4879-7111
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz
Sidi Mohamed Ben Abdellah University
Fez, Morocco
Email: taoufiq.elmoussaoui@ucmba.ac.ma

Loqman Chakir
0000-0002-8261-9370
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz
Sidi Mohamed Ben Abdellah University
Fez, Morocco
Email: loqman.chakir@usmba.ac.ma

***Abstract*—The exponential growth of data generated by the Moroccan commercial court system, coupled with the manual archiving of legal documents, has led to increasingly complex information access. As data classification becomes imperative, researchers are exploring automatic language processing techniques and refining text classification methods. In this study, we propose a BERT-CLSTM model for the classification of Moroccan commercial court verdicts. By adding a Convolutional Long Short-Term Memory Network to the task-specific layers of BERT, our model can get information on important fragments in the text. In addition, we input the representation along with the output of the BERT into the transformer encoder to take advantage of the self-attention mechanism and finally get the representation of the whole text through the transformer. The proposed model outperformed the compared baselines and achieved good results by getting an F-measure value of 93.61%.**

I. Introduction

TEXT classification is a machine-learning task that assigns a document to one or more predetermined categories based on its content. It is a key problem in natural language processing, with diverse applications such as sentiment analysis, email routing, offensive language detection, spam filtering, news classification, and language identification.

Text mining [1] is one of the most important approaches for analyzing massive volumes of textual data. Also, it is used to discover previously unknown relationships and propose solutions to aid decision-making. Many technologies are utilized in the text mining process to attain these aims. Text summarization, translation, categorization, and information extraction are a few examples. This paper's content is limited to text classification.

Despite the advances made in text categorization performance, there is still significant potential for improvement, particularly in the Arabic language. According to Internet World Stats, Arabic is the fourth most common language online, with over 225k users, representing 5.2% of all Internet users as of April 2019. Arabic NLP is still a challenging task due to the Arabic language's richness, complexity, and complicated morphology. Arabic features are assorted in abundance aspects compared to other languages. There are several forms of grammatical, variations of synonyms word, and numerous meanings of words which differ based on factors such as the order of the word.

Traditional text classification methods use sparse vocabulary features to represent documents and treat words as the smallest unit. Documents represented by such approaches typically have high dimensionality and sparse data, so the classification accuracy is low. Later, with the rise of distributed representation, the usage of high-dimensional dense vector representation documents such as the word2vec or Glove models gradually becomes mainstream. The word vectors trained using this type of method represent the contextual semantic information of the text. Recently, with the emergence of deep learning, more researchers use deep learning neural networks for text classification, such as convolutional neural networks (CNN) and recurrent neural networks (RNN). The method using deep learning for text classification involves feeding text into a deep network to generate a representation of the text and then feeding the text representation into the softmax function to calculate the probability of each category. CNN-based models [2], [3], [4] may generate text representations with local information, whereas RNN-based models [5], [6] generate text representations with long-term information.

Nowadays, the process of classifying verdicts of Moroccan commercial courts is done manually. Court staff read each verdict and classify it into a predefined category. This process has many drawbacks. First, the processing time is long. Second, court staff may make mistakes while filing the document, and third, the confidentiality of citizens' data is not respected. Based on the disadvantages of the current system, we propose a BERT-CLSTM model that can provide the same service in a short time. The proposed model combines the advantages of CNN and RNN.

The rest of the paper is structured as follows: Section II presents the related literature. In section III, we give details on the dataset created for this task of classification, the data preprocessing, and the model. Section IV presents the experimental setup, evaluation measures, and results, and we discuss them. Finally, conclusions are drawn in Section V.

II. Related Work

Over the past few years, several researchers have addressed the issue of automatic categorization of legal data and the exploitation of these huge amounts of court-generated data to

assist in decision-making. The study presented in [7] aims to classify Arabic news articles based on their vocabulary features. They employed multi-label classifiers like Logistic Regression and XGBoost, with XGBoost achieving the highest accuracy at 84.7%, while Logistic Regression scored 81.3%. Additionally, ten neural networks were constructed, and CGRU proved to be the top-performing multi-label classifier with an accuracy of 94.85%.

Research [8] introduce AraLegal-BERT, a bidirectional encoder Transformer-based model that has been thoroughly tested and carefully optimized to amplify the impact of NLP-driven solution concerning legal documents. They fine-tuned AraLegal-BERT and evaluated it against three BERT variations for the Arabic language. The results show that the base version of AraLegal-BERT achieves better accuracy than the general and original BERT over the Legal text.

El-Alami et al [9] propose an Arabic text classification method based on Bag of Concepts and deep Autoencoder representations. It incorporates explicit semantics relying on Arabic WordNet and exploits Chi-Square measures to select the most informative features. To produce a high-level representation, they applied successive stacks of Restricted Boltzmann Machines (RBMs). Experiments showed that using the Autoencoder as a text representation model combined with Chi-Square and classifier outperformed state-of-the-art techniques.

Hazm et al [10] evaluated Arabic user comments on Twitter using a common form of Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM). Experiments revealed that LSTM outperforms standard approaches in terms of accuracy while requiring less parameter computation, less working time, and more efficiency.

Alhawarat and Aseeri [11] implemented a CNN multi-kernel architecture with word embedding (n-gram) to classify Arabic documents of news. Regarding the current studies on Arabic text classification, their approach achieves very high precision using 15 of the publicly available datasets.

Galal et al [12] concentrated on classifying Arabic Text using a convolution neural network (CNN). They implemented GStem a new algorithm focused on extra Arabic letters and word embedding distances to group related Arabic words. Their studies have shown that it improves the accuracy of the CNN model when using it as a preprocessing stage.

Boukil et al [13] proposed a technique for categorizing Arabic datasets. They utilized an Arabic stemming algorithm to select and reduce features and employed Term Frequency Inverse Document Frequency (TFIDF) for feature weighting. Their study compared the CNN model and standard machine learning methods on a benchmark dataset. The authors found that the CNN model outperformed traditional methods, especially for large and complex datasets.

Al-khurayji and Sameh [14] proposed a novel method for Arabic text classification using Kernel Naive Bayes (KNB) classifier. Their approach involved preprocessing documents through tokenization, stop word removal and word stemming. They utilized Term Frequency-Inverse Text Frequency (TF-IDF) for feature extraction and represented terms as vectors. Experimental results on the collected dataset demonstrated the superiority of their methodology, showing excellent precision and efficiency compared to other baseline classifiers.

III. Research Methodology

In this section, we present the corpus created to train and evaluate our model, also the data preprocessing process, and finally, we explain the model architecture.

A. Dataset

The dataset created to train and evaluate our proposed classifier is a collection of Arabic verdicts issued from the Moroccan commercial courts. It consists of 2821 documents and 66900015 words. The average document size is 23715 words. Documents are categorized under four main classes: Unfair competition, Arbitration, Insurance, and Commercial lease.

The dataset was split into a training dataset (75% of each class) used to build the model and a testing dataset (25% of each class) used to evaluate the model's performance. The distribution of documents in each class is presented in Table I.

TABLE I
DISTRIBUTION OF DOCUMENTS IN EACH CLASS.

Class	Train	Test	Total
Unfair competition	523	175	698
Arbitration	509	163	672
Insurance	511	183	694
Commercial lease	557	200	757
Total	2100	721	2821

B. Data preprocessing

The first step in our preprocessing process was removing stop words, removing foreign characters, and punctuation by applying basic functions. Then, we execute the stemming algorithm, which transforms all words into their stems.

C. Model

The proposed BERT-CLSTM model exhibits two key characteristics. Firstly, it employs CLSTM to transform the task-specific layer of BERT, allowing our model to capture local and long-term text representations. Secondly, the output of BERT, along with the CLSTM representation, is fed into the transformer encoder. This facilitates the use of self-attention to focus the final text representation on essential segments. The architecture of the BERT-CLSTM model is illustrated in Figure 1.

1) CLSTM encoder: In order to make the representation of text focus on the information in the text. We use a convolution filter to extract features from T. Assume that the size of the convolution window is 1 × k, the output of the CLSTM encoder is:

$$P = CLSTM_k(T) \tag{1}$$

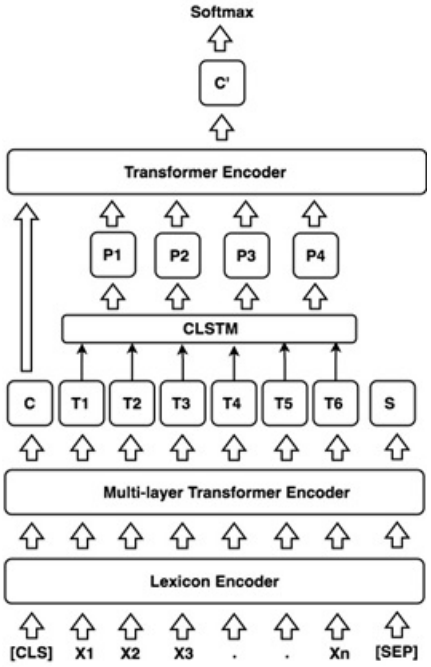

Fig. 1. Model architecture.

P is $\{P_1, P_2, P_3, ..., P_r\}$ and $r = mk + 1$. Through convolution operation, we can obtain the representation of sentences in windows with size K. For example, P_r is the representation of $T_{(r-1)}, T_r, T_{(r+1)}$.

2) Transformer encoder: Similar to multi-layer transformer encoder, to integrate information in P, we adopt transformer encoder to map the local representation P into the representation of whole text. The input of the transformer encoder is $\{C, P_1, P_2, P_3, ..., P_r\}$, and we use C' which corresponds to C as the representation of the whole text.

3) Output layer: The output of the model is represented as follows:

$$y = softmax(tanh(WC' + b)) \tag{2}$$

Where $\tanh$ presents the hyperbolic tangent function:

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3}$$

IV. Results And Analysis

This section presents the experimental setup, including the parameters used for training and testing the model. We detail the evaluation measures, highlight the results, and provide a discussion of the findings.

A. Experimental setup

To evaluate our classifier, we used the created dataset (See section III.A). Table II shows the model hyper-parameters.

TABLE II
Major hyper-parameters of the model.

Step	Parameter	Value
BERT Embedding	Size of BERT	1024
BERT Embedding	BERT layer	Last 4
CLSTM Layer	Number of filters	128
CLSTM Layer	Filter sizes	3,4,5
CLSTM Layer	Pooling function	Max Pooling
CLSTM Layer	Dropout probability	0.5
CLSTM Layer	Number of epochs	30
Training	Optimizer	Adam
Training	Learning rate	1e-3
Training	Loss	Cross Entropy

B. Evaluation measures

We used a variety of metrics to evaluate the performance of the model. We began by calculating the **Precision** measure, which indicates the number of documents correctly predicted by the model out of all documents predicted. The precision measure may be calculated using the equation 4:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives} \tag{4}$$

Second, the **Recall** measure, which indicates the number of documents accurately predicted by the model out of the total of documents in the dataset. The recall measure can be computed using this equation 5:

$$Recall = \frac{True\_Positives}{True\_Positives + False\_Negatives} \tag{5}$$

Third, **F1-measure** which is a metric that can be calculated based on the precision and recall using the following equation 6:

$$F1\_measure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{6}$$

The last metric that we use is **Accuracy**. It is a metric used in classification problems used to tell the percentage of accurate predictions. We calculate it by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{Number\_of\_correct\_predictions}{Total\_number\_of\_predictions} \tag{7}$$

C. Results

We applied the baseline models which are: Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Naive Bayes (NB), and Random Forest (RF). Then, for deep learning models, we use the CNN algorithm with the fastText embedding and of course, our proposed model which is BERT-CLSTM. Table 3 illustrates the details of the model's evaluation findings. Precision, Recall, F1-measure, and Accuracy are denoted by the letters 'P', 'R', 'F', and 'Acc'. respectively.

TABLE III
PERFORMANCE OF OUR MODEL AGAINST THE BASELINE MODELS.

Models	P (%)	R (%)	F (%)	Acc (%)
LR + count vectors	86.07	86.07	86.07	86.07
LR + word level (TF-IDF)	86.07	86.07	86.07	86.07
LR + N-gram vectors	86.07	86.07	86.07	86.07
LR + CharLevel vectors	84.81	84.81	84.81	84.81
XGBoost + count vectors	87.34	87.34	87.34	87.34
XGBoost + word level (TF-IDF)	85.44	85.44	85.44	85.44
NB + count vectors	85.44	85.44	85.44	85.44
NB + word level (TF-IDF)	84.81	84.81	84.81	84.81
NB + N-gram vectors	87.97	87.97	87.97	87.97
NB + CharLevel vectors	79.74	79.74	79.74	79.74
RF + count vectors	84.17	84.17	884.17	84.17
RF + word level (TF-IDF)	87.97	87.97	87.97	87.97
CNN + FastText	90.98	90.52	90.75	90.68
BERT + CLSTM (our model)	**93.81**	**93.42**	**93.61**	**93.55**

D. Analysis and discussion

Table III shows that our model outperformed the baseline models and the CNN model in terms of Precision, Recall, F1-Measure and Accuracy. In term of F1-Measure, our proposed model outperforms the CNN+FastText model by **2.86**, the RF+TF-IDF model and the NB+N-Gram by **5.64** as well as the XGBoost+count vectors model by **6.27** and the LR+N-Gram model by **7.54**.

Figure 2 shows accuracy scores for models. In term of Accuracy, our proposed model outperforms the CNN+FastText model by **2.87**, the RF+TFIDF model and the NB+N-Gram by **5.58** as well as the XGBoost+count vectors model by **6.21** and the LR+N-Gram model by **7.48**.

The proposed method has a better classification effect, and the reason is that the text representation matrix has strong feature representation ability and is more representative, which can provide more category information for text classification. This also highlights the advantage of using CLSTM, which gives the representations of text with local and long-term information, unlike the traditional methods that are based on the bag-of-words model.

V. CONCLUSION

Arabic is considered one of the most difficult languages to process, due to its high morphological ambiguity, writing style, and lack of capitalization. Therefore, every NLP task involving this language requires a lot of feature engineering and preprocessing. In this paper, we present our model that classifies the Moroccan commercial court verdicts into four categories. The model outperformed the compared baselines and achieved good results by getting an F-measure value of 93.61%.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco [Alkhawarizmi/2020/36].

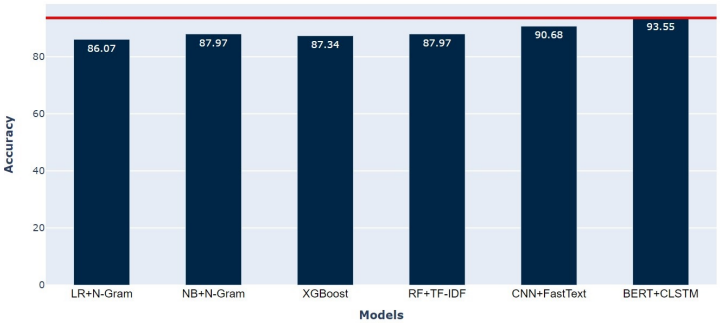

Fig. 2. Accuracy scores for models.

REFERENCES

[1] C. Blake, "Text Mining," *Annual review of information science and technology,* vol. 45, 2011, pp. 121–155.
[2] X. Zhang, J. Zhao, Y. LeCun, "Character-level convolutional networks for text classification," *Advances in Neural Information Processing Systems,* 2015.
[3] A. Conneau, H. Schwenk, L. Barrault, Y. LeCun, "Very deep convolutional networks for text classification," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,* vol. 1, 2017, pp. 1107–1116.
[4] Y. Kim, "Convolutional neural networks for sentence classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* vol. 1, 2014, pp. 1746–1751.
[5] K. Sheng Tai, R. Socher, C.D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* vol. 1, 2015, pp. 1556–1566.
[6] M. Huang, Q. Qian, X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," *ACM Transactions on Information Systems,* vol. 35, 2017, pp. 1–27.
[7] H. El Rifai, L. Al Qadi, A. Elnagar, "Arabic text classification: the need for multi-labeling systems," *Neural Computing and Applications,* vol. 34, 2022, pp. 1135–1159.
[8] M. AL-Qurishi, S. AlQaseemi, R. Soussi, "AraLegal-BERT: A pretrained language model for Arabic Legal text," *Proceedings of the Natural Legal Language Processing Workshop 2022,* 2022, pp. 338–344.
[9] F. El-Alami, A. El Mahdaouy, S.O. El Alaoui, N. En-Nahnahi, "A deep autoencoder-based representation for Arabic text categorization," *Journal of Information and Communication Technology,* vol. 3, 2020, pp. 381–398.
[10] W.H.G. Gwad, I.M.I. Ismael, Y. Gultepe, "Twitter Sentiment Analysis Classification in the Arabic Language using Long Short-Term Memory Neural Networks," *International Journal of Engineering and Advanced Technology,* vol. 9, 2020, pp. 235–239.
[11] M. Alhawarat, A.O. Aseeri, "A Superior Arabic Text Categorization Deep Model (SATCDM)," *IEEE Access,* vol. 8, 2020, pp. 24653–24661.
[12] M. Galal, M.M. Madbouly, A. El-Zoghby, "Classifying Arabic text using deep learning," *Journal of Theoretical and Applied Information Technology,* vol. 97, 2019, pp. 3412–3422.
[13] S. Boukil, M. Biniz, F. El Adnani, L. Cherrat, A.E. El Moutaouakkil, "Arabic Text Classification Using Deep Learning Technics," *International journal of grid and distributed computing,* vol. 11, 2018, pp. 103–114.
[14] R. Al-khurayji and A. Sameh, "An Effective Arabic Text Classification Approach Based on Kernel Naive Bayes Classifier," *International Journal of Artificial Intelligence and Applications,* vol. 8, 2017, pp. 01–10.