Temporal Feature Aggregation for Text Classification Based on Ensembled Deep-Learning Models

Jiali Yu, Zhiliang Qin, Linghao Lin, Yu Qin, and Yingying Li

***Abstract*—In this paper, we focus on the text classification task, which is a most import task in the area of Natural Language Processing (NLP). We propose an innovative convolutional neural network (CNN) model to perform temporal feature aggregation (TFA) effectively, which has a highly representative capacity to extract sequential features from vectorized numerical embeddings. First, we feed embedded vectors into a bi-directional LSTM (Bi-LSTM) model to capture the contextual information of each word. Afterwards, we propose to use the state-of-the-art deep-learning models as key components of the architecture, i.e., the Xception model and the WaveNet model, to extract temporal features from deep convolutional layers concurrently. To facilitate an effective feature fusion, we concatenate the outputs of two component models before forwarding to a drop-out layer to alleviate over-fitting and subsequently a fully-connected dense layer to perform the final classification of input texts. Experiments demonstrate that the proposed method achieves performance comparable to the state-of-the-art models while at a significantly lower computational complexity. Our approach obtains the cross-validation score of 95.83% for the Quora Insincere Question Classification (QIQC) dataset, and the cross-validation score of 83.10% for the Spooky Author Identification (SAI) dataset, respectively, which are among the best published results. The proposed method can be readily generalized to signal processing tasks, e.g., environmental sound classification (ESC) and machine fault analysis (MFA).**

***Index Terms*—NLP, text classification, word embedding, bi-LSTM, Xception, WaveNet.**

I. Introduction

The target of Natural Language Processing (NLP) is to enable computers to understand language in a manner comparable to human beings. Human language is a complex symbol system through which information that varies due to subtle differences in words or intonations can be conveyed effectively. However, computers lack the excellent capacity to understand contextual information embedded or implied in human languages, a scenario that significantly motivates the development of NLP algorithms. To be more specific, the computer accepts an user's input in the form of embedded numerical vectors, and internally performs a series of mathematical operations such as pre-processing and calculations through pre-defined algorithms to simulate the human understanding of natural language and return results expected by users [1].

Manuscript received March 19, 2021; revised May 23, 2021.

The authors are with Weihai Beiyang Electric Group Co. Ltd., Weihai, Shandong, China (e-mails: {yujiali, qinzhiliang, linlinghao, qinyu, liyingying}@beiyang.com}; Corresponding author: Zhiliang Qin (qinzhiliang@beiyang.com)

In recent years, NLP has developed at a remarkable pace as an interdisciplinary subject. It is viewed as the intersection of artificial intelligence (AI), computer science, communication systems, and information engineering, which also involves the in-depth knowledge of statistics and linguistics. Prior to the neural network (NN) technology being applied to the NLP, the research in this area has evolved through a lengthy process starting from the formulation of rules to the mining of statistics [2], [3]. With the development of deep learning, the availability of large datasets and the deployment of light-weight models, great successes have been achieved in many NLP-related sub-tasks, such as named entity recognition (NER), text classification, similarity analysis, text generation, etc. Specially, text classification is generally recognized as the most important and the most representative sub-task, which further includes the applications of sentiment analysis, spam detection, author attributes, information retrieval [4], [5]

In the past few years, convolutional neural networks (CNN) working with the numerical representations of input texts, also known as word embeddings, have been widely used in solving text classification problems. Ciriket analyzed a word embedding method in a supervised task and measured the similarity of contexts by filling in the distribution of their alternatives, which achieves impressive results in multilingual dependency parsing [6]. On the other hand, recurrent neural networks (RNN) and its variants such as long-short-term memory (LSTM) and gated recurrent unit (GRU), have shown great successes in modeling sequential data [7]. The authors proposed a Bi-LSTM model with the attention mechanism that automatically focuses on words decisive for classification to capture the most important semantic information in a sentence [8]. With respect to deep-learning models, Chollet proposed the Xception architecture that features deep separable convolution layers and is viewed as a low-complexity adaptation of Google's Inception models [9]. In [10], Oord introduced a novel deep neural network, i.e., the Wavenet model, for generating raw audio waveforms, which can be vierwed as a sequence generation model. Most recently, the BERT module [11], which stands for the bi-direction encoder representations from Transformers, is becoming increasingly popular in the NLP area. However, the training and fine-tuning of BERT involves a high computational and time complexity, which prohibits its feasibility in mobile applications.

Motivated by the need to develop light-weight architectures to achieve the state-of-the-art performance on mobile data terminals, in this paper we propose an innovative temporal feature aggregation (TFA) mechanism based on ensembled deep-learning modeling for typical NLP classification tasks. In the proposed architecture, we

doi: 10.18178/ijfcc.2021.10.2.575

concatenate the bi-directional LSTM layer with two sub-networks, i.e., Xception and WaveNet, to efficiently extract deep features of input texts with different word embedding mechanisms. Various pre-trained embedding methods will be described in Section II, including the Glove embedding, FastText embedding, paradigm embedding, and random embedding. First, we use word embedding to convert the text into a numerical vector, which serves as the input to the proposed algorithm. Afterwards, we feed numerical embeddings into the Bi-LSTM layer to model the contextual information of each word, the output of which is passed to the Xception and WaveNet models to generate temporal features, respectively. To facilitate an effective feature fusion and obtain an accurate representation of contextual information, the features of two models are serially concatenated before being forwarded to the classification layer. The main contributions of this paper are as follows: 1) propose an ensembled architecture for NLP incorporating two state-of-art deep learning models, i.e., Xception and WaveNet; 2) perform a comprehensive evaluation of the proposed model on two classical data sets for binary classification and multi-class classification, respectively; 3) achieve performance as comparable to the best results published to date, while at a significantly lower computational complexity than that of the BERT model.

The rest of the paper is organized as follows. In Section II, we introduce several embedding methods to convert input texts to numerical vectors, a procedure that is necessary for invoking deep-learning models. Following that, we present the proposed TFA architecture in detail. In Section III, the experiment setting is introduced and the performance results are presented. The conclusion and the future work are drawn in Section IV.

II. Network Architecture

The proposed architecture is composed of several key components, i.e., a word embedding layer, the Bi-LSTM layer, and the feature extraction module composed of the Xception model and the Wavenet model. In this section, we present each component in a sequential manner.

A. Word Embeddings

The first component of the proposed architecture is the word-embedding layer. Word embeddings, originally known asWord2vec, are effectively trained on millions of data sets and act as a widely-accepted measure of the similarity between semantic words [12]. The embedding method expresses an input text via a low-dimensional numerical vector and hence words with similar semantics will be close to each other in the vector space. An extension of Word2vec is known as GloVe (Global Vectors for Word Representation), which combines global statistics and Word2vec's context-based learning effectively [13]. It is a representation tool based on word co-occurrence matrices and delivers semantic characteristics between words. The semantic similarity between two words can be computed through operations on vectors, such as the Euclidean distance or the cosine similarity. The cost function of the model is given by (1) [13],

$$J = \sum_{i,j}^{N} f(X_{i,j})(v_i^T v_j + b_i + b_j - log(X_{i,j}))^2 \quad (1)$$

where v_i and v_j are the vectors of words i and j, respectively, b_i, b_j are two scalars, f is the weight function, and N is the size of the vocabulary. Another word embedding method is referred to as FastText [14], which uses a shallow neural network to implement word2vec and text classification functions simultaneously. The core of FastText is to superimpose and average the word and n-gram vectors of the entire document to obtain the document vector based on which the model performs the multi-classification of input texts. FastText has several advantages as compared with other embedding methods. First, FastText speeds up training and testing process while maintaining high precision. Second, FastText does not need pre-trained word vectors and instead, trains word vectors by itself. Third, FastText resorts to two criterions to achieve performance/complexity tradeoffs, i.e., hierarchical soft-max and n-gram. The basic idea of the hierarchical soft-max optimization is to construct a Huffman tree according to the frequency of the category to replace the standard soft-max operation and lower the complexity from N to log$N$ through a bifurcated graph structure. Fig. 1 shows an example of the hierarchical soft-max operation,

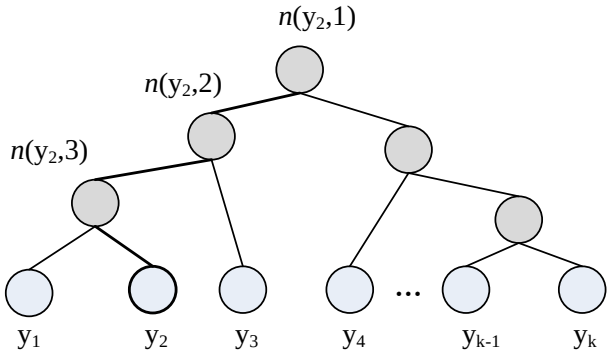

Fig. 1 An example of a hierarchical soft-max operation.

In Fig. 1, K different class labels form all leaf nodes and K-1 nodes are used as internal parameters. The nodes and edges passing from the root node to a certain leaf node denoted by $L(y_i)$. Thus, $P(y_i)$ can be written as (2)

$$P(y_i) = \prod_{i=1}^{L(y_i)-1} \sigma(f(n(y_i,l+1) = LC(n(y_i,l))) \cdot \theta n(y_i,l)^T X) \quad (2)$$

where σ represents the sigmoid function, $LC(n)$ represents the left child of the nth node, $\theta_n(yi,l)$ is the parameter of the intermediate node $n(y_i,l)$, X is the input of the conventional soft-max operation and $f(x)$ is a special function defined as

$$f(x) = \begin{cases} 1, & \text{if } x == \text{true} \\ -1, & \text{otherwise} \end{cases} \quad (3)$$

For instance, the highlighted nodes and edges in Fig. 1 form the path from the root node to y_2 with path length $L(y_2)$=4. $P(y_2)$ can be expressed as (4),

$$P(y_2) = P(n(y_2,1), left) \cdot P(n(y_2,2), left) \cdot P(n(y_2,3), right)$$
$$= \sigma(\theta_{n(y_2,1)}{}^T X) \cdot \sigma(\theta_{n(y_2,2)}{}^T X) \cdot \sigma(-\theta_{n(y_2,3)}{}^T X) \quad (4)$$

In practical applications, several techniques that combine word embedding and sentence embedding are designed based on machine learning under unsupervised conditions. In [15], Wieting introduced Paragram-phrase embedding, which uses a novel training objective to learn robust sentence-level embedding [15]. Wieting also compares different techniques of word embedding and evaluates the learned embedding as well in predicting the sentiment classification label, and finds out that the best-performing mechanism is a single embedding layer gradually refined in a supervised manner.

B. Bi-directional LSTM (Bi-LSTM)

The Long-Short-Term-Memory (LSTM) layer is an improved type of the Recurrent Neural Networks (RNN) and is suitable for modeling time-series data such as texts, voice segments etc., due to its potentials to learn contextual information. Basically, an LSTM unit consists of three multiplication gates, which control the proportion of information that will be forgotten and passed to the next time step. Fig. 2 gives the basic structure of an LSTM unit.

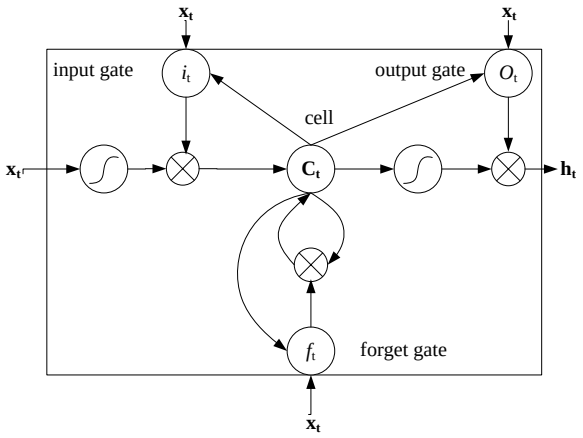

Fig. 2. A long-short-term-memory cell.

Formally, the equations to update an LSTM unit at time t are given by (5)

$$\begin{cases} \boldsymbol{i}_t = \sigma(\mathbf{W}_{\mathrm{i}}\boldsymbol{h}_{t-1} + \mathbf{U}_{\mathrm{i}}\boldsymbol{x}_t + \boldsymbol{b}_{\mathrm{i}}) \\ \boldsymbol{f}_t = \sigma(\mathbf{W}_{\mathrm{f}}\boldsymbol{h}_{t-1} + \mathbf{U}_{\mathrm{f}}\boldsymbol{x}_t + \boldsymbol{b}_{\mathrm{f}}) \\ \boldsymbol{c}_t = \boldsymbol{f}\boldsymbol{t}\boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\mathbf{W}_{\mathrm{c}}\boldsymbol{h}_{t-1} + \mathbf{U}_{\mathbf{c}}\boldsymbol{x}_t + \boldsymbol{b}_{\mathrm{c}}) \\ \boldsymbol{o}_t = \sigma(\mathbf{W}_{\mathrm{o}}\boldsymbol{h}_{t-1} + \mathbf{U}_{\mathrm{o}}\boldsymbol{x}_t + \boldsymbol{b}_{\mathrm{o}}) \\ \boldsymbol{h}_t = \boldsymbol{o}_t \tanh(\boldsymbol{c}_t) \end{cases} \quad (5)$$

where σ is the logistic sigmoid function, $\boldsymbol{x}_t$ is the input vector (e.g. word embedding) at time t, $\boldsymbol{h}_t$ is the hidden state (also called output) vector storing all the useful information at (and before) time t; $\mathbf{U}_{\mathrm{i}}$, $\mathbf{U}_{\mathrm{f}}$, $\mathbf{U}_{\mathrm{c}}$, and $\mathbf{U}_{\mathrm{o}}$ denote the weight matrices of different gates for input $\boldsymbol{x}$; $\mathbf{W}_{\mathrm{i}}$, $\mathbf{W}_{\mathrm{f}}$, $\mathbf{W}_{\mathrm{c}}$, and $\mathbf{W}_{\mathbf{o}}$ are the weight matrices for hidden state $\boldsymbol{h}_t$; and $\boldsymbol{b}_{\mathrm{i}}$, $\boldsymbol{b}_{\mathrm{f}}$, $\boldsymbol{b}_{\mathrm{c}}$, and $\boldsymbol{b}_{\mathrm{o}}$ denote the bias vectors.

For many sequence labeling tasks, access to the past (left) and future (right) contexts is beneficial to learning. However, in the LSTM network, the hidden state $\boldsymbol{h}_t$ only stores the past information. In order to capture information flow in both directions, we propose to use the Bi-LSTM layer to extract temporal feature, which is an elegant solution by combining a forward LSTM and a backward LSTM [16]. The basic idea is to present each sequence forward and backward to two separate hidden states, and subsequently connect two hidden states to form the final output.

C. Xception and WaveNet

The deep-learning model Xception is based on the kernel of the depthwise convolution operation as shown in Fig. 3, which differs from the regular convolution by performing operations in a two-dimensional plane. The number of convolution kernels is the same as the number of channels in the previous layer. The number of feature maps at the output of the depthwise convolution is the same as the number of channels in the input. Since this operation independently performs convolution operations on each channel of the input and does not effectively use the feature information arising from different channels, a pointwise convolution is needed to combine these feature maps to generate a new feature map. The operation of pointwise convolution is very similar to that of regular convolution with the size of the convolution kernel given by $1\times1\times M$, where M is the number of input channels. Therefore, the depthwise operation involves much fewer parameters as compared with the standard convolution operation.

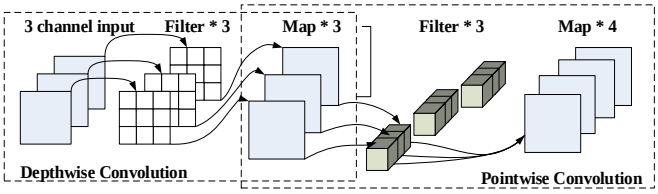

Fig. 3. Block diagram of depthwise separable convolution.

The Xception is known as a lightweight deep-learning model with comparable performance to more computationally intensive models. On the ImageNet dataset, Xception has a slightly higher accuracy rate than the Inception-v3 model while at a significantly lower number of parameters. The residual connection mechanism incorporated in the Xception model significantly speeds up the convergence process and achieves a significantly higher accuracy.

WaveNet is a deep neural network model for generating raw audio waveforms, and viewed as a best model for converting text to human voice in the automatic sound recognition (ASR) field. The model features an autoregressive (AR) mechanism that predicts the probability distribution of the current audio sample based on all previously generated samples. By using causal convolution, the WaveNet ensures that the order of modeling the data is maintained. The prediction $p(x_{t+1}|x_1,...,x_t)$ delivered by the model at time step t does not depend on any future time steps x_{t+1}, x_{t+2},..., x_t. The equivalent form of causal convolution is also known as mask convolution [17] as illustrated in Fig. 4, which can be realized by constructing a mask tensor and performing an element-wise multiplication on the mask with the kernel before proceeding with the convolution.

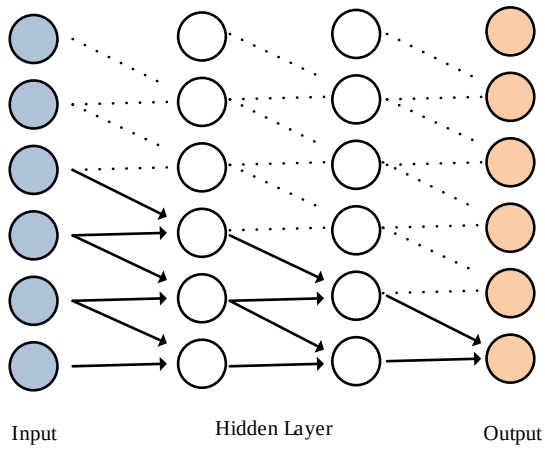

Fig. 4. Block diagram of causal convolution.

In Fig. 5, the input represents the audio signal sampling points. After passing through a casual operation, a gated convolution is introduced for control purposes. The expression for the gated convolution is given by (6),

$$z = \tanh(\mathbf{W}_{f,k} * x) \otimes \sigma(\mathbf{W}_{g,k} * x) \tag{6}$$

where the symbol $*$ means convolution operation and $\otimes$ is the position multiplication operator. A residual short-circuit is incorporated in the Wavenet to facilitate information flow and avoid the gradient diminishing problem. The final result is obtained by superimposing multiple skip-connections based on the intermediate results of each layer.

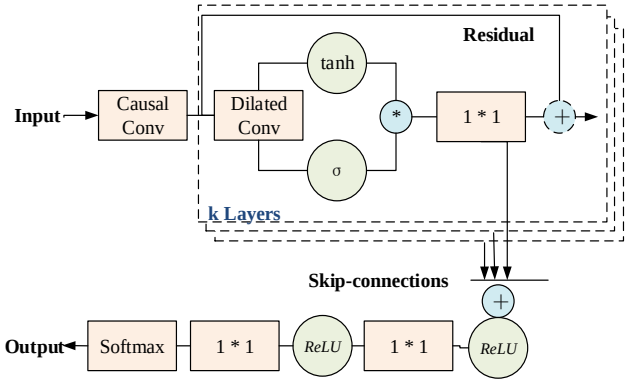

Fig. 5. Block diagram of the Wavenet model.

D. Architecture

Fig. 6 shows the block diagram of the proposed architecture. In NLP tasks, it is usually necessary to pass through several pre-processing steps to convert the input raw text into a format that is readable by the model. The text preprocessing are generally summarized as the following steps: perform tokenization, which is to separate the text into its individual constituent words; discard stopwords, which is to ignore any words that appear too frequently as its frequency of occurrence will not be useful in helping detecting relevant texts, stemming to combine variants of words into a single parent word that conveys the same meaning, and finally, vectorization to convert the text into a vector format.

We construct our architecture by feeding the output vector of the embedding layer to a Bi-LSTM layer, the output of which is further forwarded into two branches, i.e., the Xception and the WaveNet models. The bi-LSTM layer is able to learn contextual information from the embedded vectors. In order to make full use of the advantages of these two models, we concatenate deep features delivered by the Xception and WaveNet models. To alleviate the occurrence of over-fitting and achieve the effect of regularization, we introduce the drop-out operation prior to the fully connected layer, which randomly removes features propagated from the network in the training phase [18], [19]. The right-hand-side (RHS) of Fig. 6 shows in detail the structure of the Xception model.

In the next section, we provide details about the experiment on the proposed architecture. We perform experiments on two Kaggle data sets, named "Quora Insincere Questions Classification (QIQC)" and "Spooky Author Identification (SAI)", to verify the effectiveness of our proposed model.

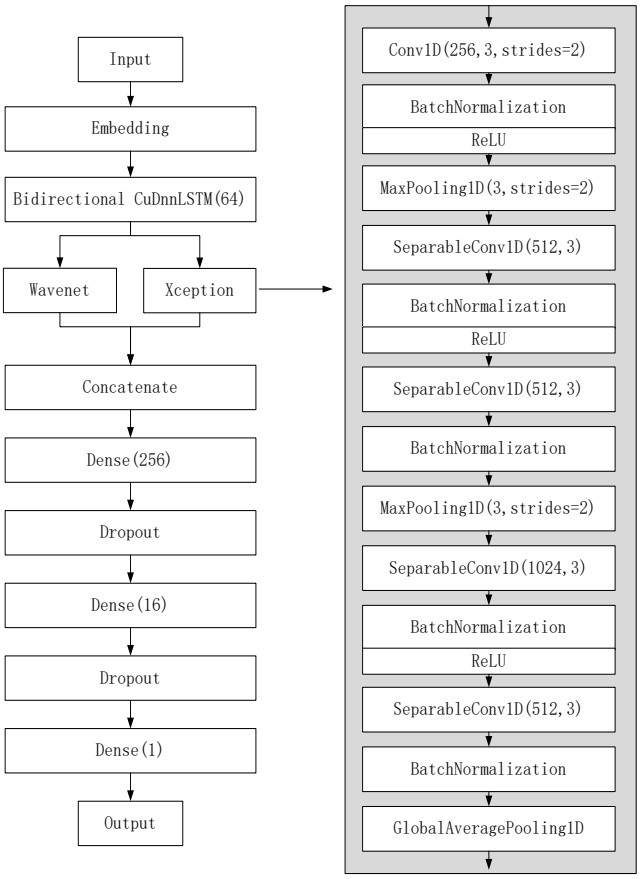

Fig. 6. Block diagram of the proposed architecture.

III. Numerical Results

A. Binary-Class Text Classification

We first evaluate the proposed model on the QIQC dataset, which can be obtained from the weblink (https://www.kaggle.com/c/quora-insincere-questions-classification). In this dataset, we are given a series of questions and are asked to classify whether the problem is sincere or insincere. Besides, insincere questions include toxic or divisive comments as shown in Table I, which complicates

the task.

In the training set, there are 1,306,122 training samples, each of which has an identifier, question (question text) and the category (target). In order to obtain effective information and alleviate over-fitting, we randomly select one-tenth of the training data as a cross-validation set, which is used to evaluate the performance of the model. That is, we perform a 10-fold split and fit the same model up to several times on the same training split and then successively evaluate on the out-of-fold subset.

TABLE I. QUESTION SAMPLES IN THE QIQC DATASET

Labels	Questions
Sincere (0)	"Why my package still is ISC since May 31, 2017 and I don't have updated?" "How difficult is it to find a good instructor to take a class near you?" "How were the Calgary Flames founded?"
Insincere (1)	"How do I find smart friends, if I am in a school of dumb people?" "Why are there so many sensitive liberals on Quora?" "Why does every vegan think that they are saving the planet?"

We use the Keras network library for training our model. Four pre-processing techniques are used to vectorize input texts including random embedding, which involves the least amount of computational complexity and serves as a benchmark, as well as three pre-trained embedding methods, i.e., Glove, WikiNews, and Paragram embeddings, as shown in Table II. The training generally requires less than 10 epochs to converge and the performance on cross-validation (CV) sets is presented in Table II. The evaluation metric used in the experiment is the F1 score, which is a recommended metric in the case of an unbalanced dataset. Table II shows that the WikiNews FastText embedding achieves the best CV-score of 95.83%, and the Glove embeddings achieves the best F1-score of 0.6648. Compared with random embedding methods, the pretrained embeddings are shown to provide better performance results.

TABLE II: EVALUATION RESULTS WITH VARIOUS EMBEDDINGS

Embeddings	CV-score	F1-score	Threshold
Random	95.12%	0.6548	0.10
Glove	95.70%	0.6648	0.24
WikiNews	95.83%	0.6617	0.43
Paragram	95.66%	0.6527	0.21

B. Multi-class Text Classification

To extend the applicability of our proposed model to multi-class text classification, we conduct experiments on the SAI dataset, which contains excerpts from several scary storywriters and can be obtained from the weblink (https://www.kaggle.com/c/spooky-author-identification), with the following specific target: predict who was writing a sentence of a possible spooky story among the following authors, i.e., Edgar Allan Poe (EAP), HP Lovecraft (HPL) and Mary Wollstonecraft Shelley (MWS). The dataset is presented in the format of a csv file and contains the author names along with a set of sentences. The authors are represented as one of three categorical names, i.e., EAP, HPL, and MWS, as given by Table III. The overall data includes 19,579 observations with 7,900 excerpts (40.35%) of EAP, 5,635 excerpts (28.78%) of HPL, and 6,044 excerpts (30.87%) of MHS, respectively.

TABLE III. SAMPLE IN THE SPI DATASET

Authors	Excerpts
Edgar Allen Poe (EAP)	"These bizarre attempts at explanation were followed by others equally bizarre." "Meantime the whole Paradise of Arnheim bursts upon the view."
HP Lovecraft (HPL)	"Even now They talked in Their tombs." "He cried aloud once, and a little later gave a gasp that was more terrible than a cry."
Mary Shelley (MWS)	"The visits of Merrival to Windsor, before frequent, had suddenly ceased." "Perpetual fear had jaundiced his complexion, and shrivelled his whole person."

The data pre-processing and embedding parameters are the same as the binary classification experiment. Similarly, we randomly choose one-tenth of the data from the original training set as an out-of-fold validation set, and evaluate the performance of the model on the cross-validation set. We conduct experiments with random embedding and three pre-trained embeddings, which give the CV score of 0.7465 (Random), 0.8310 (Glove), 0.8055 (WikiNews), and 0.8230 (Paragram), respectively. The confusion matrices of four embeddings are shown in Fig. 7, which shows that the proposed architecture achieves impressive performance even over such an ambiguously defined dataset. Compared with the best known results based on the BERT model, we have achieved very similar performance by using the proposed temporal feature fusion approach.

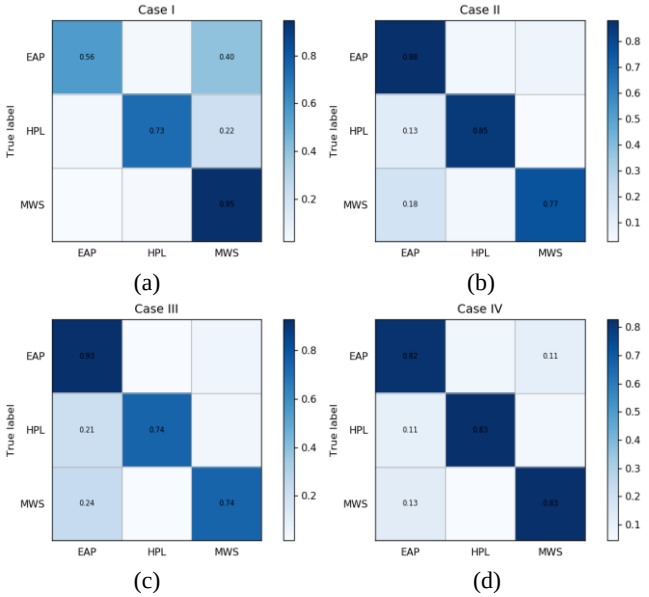

Fig. 7. Confusion matrix of various embeddings for the multi-class classification task. (a) Case I (Random). (b) Case II (Glove). (c) Case III (WikiNews). (d) Case IV (Paragram).

The Xception model is a deeply separable model that improves network efficiency and performance with limited hardware resources, while the WaveNet model inherently allows us to take advantage of the efficiency of the convolutional layer while alleviating the challenge of learning long-term dependencies over a large number of time steps. The proposed architecture innovatively extracts global temporal features from the Bi-LSTM layer, which are subsequently forwarded to the state-of-art convolutional

models to obtain refined local temporal features to increase the representative capacity. Besides, feature concatenation proves to deliver superior performance as compared with other arithmetic operations as useful information is preserved to the largest extent. Besides, we introduce the dropout layer to prevent overfitting and improve the generalization ability of the model by reducing the number of intermediate inputs prior to the final classification layer.

IV. Conclusion

This paper proposed an innovative method for text classification, which uses the concatenated features of the ensembled Xception and WaveNet models to achieve temporal feature fusion. Feature concatenation enables the proposed architecture to extract both global and local features from input texts effectively. The proposed method achieves performance comparable to the computationally intensive BERT models by resorting to low-complexity operations including separable and dilated convolutions.

References

[1] J. Yang, L. Bai, and Y. Guo, "A survey of text classification models," in *Proc. 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*, 2020, pp. 327–334.

[2] B. Alshemali and J. Kalita, "Improving the reliability of deep neural net-works in NLP: A review," *Knowledge-Based Systems*, vol. 191, p. 105210, 2020.

[3] M. Gupta, V. Varma, S. Damani, and K. N. Narahari, "Compression of deep learning models for NLP," in *Proc. 29th ACM International Conference on Information and Knowledge Management*, 2020, pp. 3507–3508.

[4] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," *arXiv preprint arXiv:1906.02243*, 2019.

[5] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," *arXiv preprint arXiv:2004.03705*, 2020.

[6] V. Cirik and D. Yuret, "Substitute-based code word embeddings in supervised NLP tasks," *arXiv preprint arXiv:1407.6853*, 2014.

[7] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, "Time-series performance prediction based on long short-term memory (LSTM) neural network model," *Journal of Petroleum Science and Engineering*, vol. 186, p. 106682, 2020.

[8] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 207– 212.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[10] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499* 2016.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint, arXiv:1910.01108*, 2019.

[12] F. Almeida and G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv: 1901.09069*, 2019.

[13] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. 2014 conference on Empirical Methods in Natural Language processing (EMNLP)*, 2014, pp. 1532–1543.

[14] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[15] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "From paraphrase database to compositional paraphrase model and back," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 345–358, 2015.

[16] X. Yu, W. Feng, H. Wang, Q. Chu, and Q. Chen, "An attention mechanism and multi-granularity-based bi-LSTM model for Chinese Q&A system," *Soft Computing*, vol. 24, no. 8, pp. 5831–5845, 2020.

[17] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. International Conference on Machine Learning*, PMLR, 2016, pp. 1747–1756.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

Jiali Yu was born in China, on July 27, 1994. Jiali Yu was graduated from University College Dublin with a major in computer science, which is a national university of Ireland, and received a Master of Science degree on February 14, 2019.

She now works as an algorithm engineer in the Weihai Beiyang Electrical Group, Shandong, China. Her current research interests are mainly in machine learning, natural language processing, and speech signal recognition.

Zhiliang Qin was born in 1974. He obtained the B. Eng. degree from the Beijing Institute of Technology (BIT) in 1995, the M. Eng. degree from the Graduate School of China Academy of Engineering Physics (CAEP) in 1998, and the Ph.D. degree from the Nanyang Technological University (NTU), Singapore in 2003. From 2002 to 2019, he worked at the Agency for Science, Technology, and Research (A*STAR) in Singapore as the Scientist in the area of algorithm developments for machine learning, signal processing, data analytics, optimization theories, and data storage systems. From 2019 to present, he is the Deputy Chief Engineer at the Weihai Beiyang Electric Group. Co. Ltd, Weihai, Shandong, China. Dr Qin published around 80 SCI and EI technical papers, and authored three U.S. patents. He frequently takes the role of being the reviewer of international research journals and being the Technical Committee Member (TPC) of international academic conferences on artificial intelligence (AI) and signal processing, including the ICSPS 2020, MLMI2020, ICCCR2021, AIACT 2021, ICEEMT 2021, etc. He is an IEEE senior member and serves on the Editorial Board of the Journal of Communications (JCM), the American Journal of Computer Science and Technology (AJCST), and the Journal of Computer and Communications (JCC).

Linghao Lin was born in China, on June 10, 1995. He graduated from the University of Science and Technology Beijing with a major in control engineering, and received the M. Eng. degree in January, 2021.

He now works as an algorithm engineer in Weihai Beiyang Electrical Group in Shandong, China. His current research interests include multiple object track (MOT), object detection and artificial intelligence.

Yu Qin was born in Weihai, China, on June 25, 1992. He received the B.S. and the M.S. degree in the School of Mathematics from the Shandong University, Ji'nan, China, in 2014 and 2017, respectively.

He is currently with Weihai Beiyang Electrical Group co. Ltd, Weihai, 264200, China. His current research interests include artificial intelligence, speech recognition, natural language processing.

Yingying Li received the B.Eng. in the School of Communication Engineering from Harbin Institute of Technology, Weihai, China, in 2008. and the M.Eng. degree in the School of Electronic and Communication Engineering from the Shandong University, Weihai, China, in 2015. She is currently with Weihai Beiyang Electrical Group co. Ltd, Weihai, China. Her current research interests include fault diagnosis, artificial intelligence, and computer visions.