Chinese Spam Detection based on Prompt Tuning

Yan Zhang
College of Computer Science
Inner Mongolia University
Hohhot, China
zy09230@163.com

Chunyan An*
College of Computer Science
Inner Mongolia University
Hohhot, China
ann@imu.edu.cnd

***Abstract*—Spam has plagued Internet users for a long time, and it is of great significance to design an efficient spam detection method. In recent years, spam detection methods based on fine-tuning pre-trained language models (PLM) have achieved great success. The approach is to fine-tune a pre-trained language model on a large dataset to adapt it to the downstream spam detection task. However, the objective of the initial training phase of PLM is inconsistent with the objective of downstream tasks, which results in the downstream tasks cannot fully utilize the latent knowledge in PLM. In this paper, we use Prompt Tuning and PLM to identify Chinese spam by constructing additional prompt templates, converting the email classification task into a fill-in-the-blank task, and then getting the email classification results according to the filling content on the prompt templates. This process is very similar to the process of initial training of PLM, which can more fully utilize the rich knowledge in PLM. We use prompt tuning to train the model on public datasets. Through experiments, we found that the accuracy score of the proposed model on trec06 datasets can reach 0.996, and the F1 score can reach 0.994, which is better than the comparison model. In terms of model convergence speed, the proposed model only needs less than 200 training steps to converge, which is faster than the comparison model.**

Keywords-Chinese spam detection; deep learning; prompt tuning; ERNIE

I. INTRODUCTION

The number of Internet users in China has grown rapidly in the past 10 years due to the booming of traditional Internet and mobile Internet. According to "The 44th China Statistical Report on Internet Development", as of June 2019, China's number of Internet users has reached 854 million [1]. This rapid development also facilitates the dissemination of information, including social media, email, WeChat platforms, and other applications. At the same time, these channels also attract malicious users to spread spam, threatening people's property safety. The goal of spam detection technology is to filter out spam before it causes damage to users.

Current mainstream spam detection methods are based on machine learning (ML) and deep learning (DL) techniques. The characteristics of emails are learned through sample data to classify emails. Models commonly used for spam detection include support vector machines (SVM) [2], convolutional neural networks (CNN) [3], recurrent neural networks (RNN) [4], etc. How to obtain accurate text features has a huge impact on model performance, and the pre-trained language models (PLM) that have appeared in recent years have effectively solved this problem. PLM has achieved good results in multiple NLP tasks, and the effect of email detection models can be significantly improved by PLM [5]. The approach is to fine-tune PLM through the task target dataset to make it suitable for downstream tasks. However, an important problem is that the target task of the initial training of PLM is the filling-in-the-blank task, and the downstream task is the classification task, which will cause the model to fail to fully utilize the knowledge in the PLM.

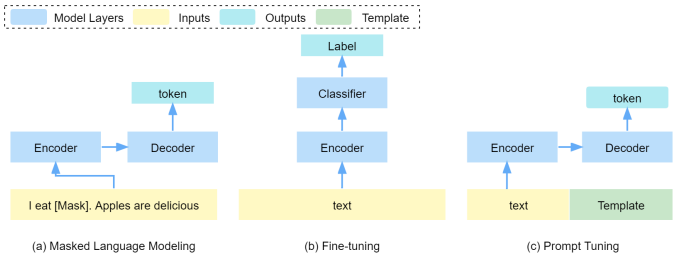


Figure 1. Pre-trained language model training process (a), and two paradigms of fine-tuning (b) and prompt tuning (c).

Now, a new paradigm called prompt tuning [6] has achieved satisfactory results in tasks such as news classification. It does not directly classify through the features of the text but designs prompt templates to convert downstream tasks into similar to the initial training of PLM, let PLM directly complete the task of filling in the blanks, making more efficient use of the rich knowledge in PLM, as shown in Figure 1. However, spam classification is somewhat different from news classification: in order to evade detection, spam will use various methods to disguise as normal emails, and even express friendly feelings, while normal emails may contain some negative or even offensive Emotion, which brings greater challenges for PLM to fill in the blanks in the prompt templates.

The contributions of this paper are summarized as follows: We design a Chinese spam detection model based on prompt tuning and ERNIE [7]. We construct prompt templates as additional information to help the ERNIE model achieve spam detection. We conduct experiments on the public Chinese spam dataset, and the experiments show that our model outperforms existing models in terms of accuracy score, F1 score, and convergence speed.

*Corresponding Author
DOI reference number: 10.18293/SEKE2022-120

II. Related Work

Over the years, many methods have been proposed to detect spam based on the content of the email. These methods fall into three main categories: supervised, semi-supervised[8], and unsupervised[9] methods. Supervised methods treat mail detection as a classification task, and supervised methods generally show better performance compared to other methods[10].

Among the various methods of supervised learning, methods based on deep learning perform better. [11] tested many supervised learning algorithms such as NB, SVM, KNN, etc., which were used individually or in combination to detect spam. [12] found that ML methods are inefficient in the case of high-dimensional data and different spammers, and it becomes crucial to explore DL methods with effective feature selection mechanisms. With the development of deep learning, many scholars began to use DL technology to automatically learn features. In NLP, deep learning methods are mainly based on a distributed representation of each word, also known as word embedding [13]. [14] implemented a Semantic Convolutional Neural Network (SCNN) model that maps word vectors using an NLP technique called Word2Vec. [5] shows that different methods of obtaining word vectors have a greater impact on feature extraction, and using PLM to obtain word vectors is the best way.

The common approach in existing works is to extract features using encoders of pre-trained language models and then use classification algorithms to identify spam. Spam will deceive the detection model by carefully designing the input content, which requires the detection model to obtain sufficiently accurate email features. [5] used BERT Encoder to obtain email features and tested a variety of classification algorithms. In addition, they simulated the camouflage methods commonly used in spam such as synonym replacement to test the performance of the model. The results show that the BERT-based detection model can effectively resist this camouflage strategy. [15] used the M-BERT pre-trained language model, which can encode multiple languages to achieve multilingual mixed spam classification.

The spam detection model designed in this paper is aimed at Chinese spam, and we select the ERNIE model to extract the text features of the email. The ERNIE model has achieved excellent results in some Chinese natural language processing tasks and demonstrated strong knowledge reasoning ability in the cloze test. ERNIE uses multi-layer Transformer [16] as basic encoder. The Transformer can capture the contextual information for each token in the sentence via self-attention, and generates a sequence of contextual embeddings. In the training stage, they randomly mask 15 percents of basic language units, and using other basic units in the sentence as inputs, and train a transformer to predict the mask units [7].

In past studies, the standard paradigm for using pre-trained language models to handle downstream tasks is the fine-tuning paradigm [17], which focuses on designing training objectives in the training phase to adapt pre-trained language models to downstream tasks. Today, this paradigm is hopefully replaced by a paradigm called prompt tuning, which reformulates downstream tasks with the help of textual cues to look more like the tasks solved during the original PLM training [6]. Some studies have applied this paradigm in text classification tasks, such as [18] formulating task-specific prompt templates according to logical rules, outperforming baseline models in many-class text classification. [19] showed that training models with prompt templates on different tasks or different amounts of data can improve the prediction performance.

Although the use of the prompt-tuning paradigm in some classification tasks can improve the model effect, since spam emails are disguised as normal emails using various methods, there is no relevant research to show that this paradigm is still suitable for spam detection models. In this paper, a spam detection model based on the ERNIE and prompt tuning is designed for Chinese spam. The prompt template is manually set, and PLM completes the task of filling in the template. Then the model classifies the email according to the content filled in the template by PLM.

III. Prompt Tuning Method

An example of email detection with prompt tuning is shown in Figure 2. Our model consists of the following parts:

- Prompt Addition: Design a suitable prompt template for the task. The content of the template is a sentence describing the email message, and it has a [Mask] placeholder. This template is spliced with the original text and used as the input of the Encoder.
- Fill in the Prompt template: The input text is encoded and decoded by PLM, and the filling content of the [Mask] position is obtained.
- Answer Mapping: Mapping the filled content Y of the [Mask] position with the answer space Z, resulting in an effective predictive model.

A. Prompt Addition

A classification task can be denoted as $T = \{X, Y\}$, where X is the instance set and Y is the class set. For each $x \in X$, there is a unique label $y \in Y$.

Prompt Addition needs to design a prompt template T() suitable for the task first, and then map each text x to $x_{prompt} = T(x)$, x_{prompt} is obtained by splicing the prompt template and the email text. The prompt template converts the original task into a fill-in-the-blank task by adding additional prompt information and at least one [Mask] placeholder. We set T() = 'This is a [Mask][Mask] email' and x maps to x_{prompt} = 'This is a [Mask][Mask] mail. x '. In addition, we set a character set V, $v \in V$ to fill the prompt template.

B. Fill in the Prompt template

Calculate the conditional probability $p([MASK] = v| x_{prompt})$ of the [Mask] position padding character using the encoder-decoder network structure from PLM. First add the token embedding, position embedding, and segment embedding of x_{prompt} as the input of the encoder.

Encoder uses multi-layer Transformer structure to capture the context of each character through self-attention to encode the

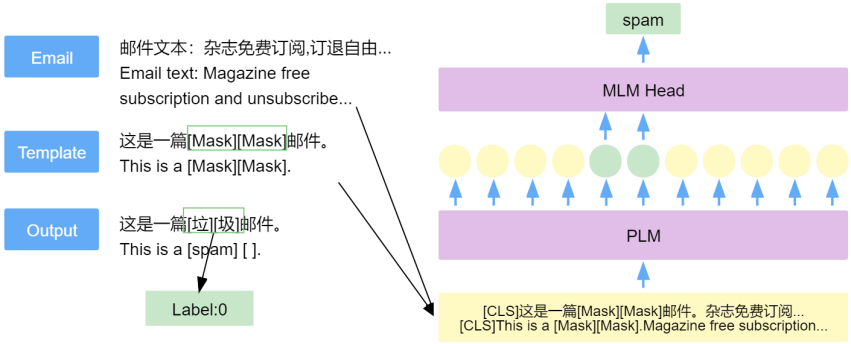


Figure 2. An example of implementing email detection using prompt tuning.

input content. The Multi-Head Attention in the Decoder decodes according to the output from the Encoder, and outputs the hidden vector $h_{[Mask]}$ of [Mask]. Calculate the dot product of $h_{[Mask]}$ and Emb(v), Emb(v) is the embedding of the token v. Then, the scores of all characters in the model dictionary are calculated by the softmax function, and the character with the largest score is obtained as the predicted character. To prevent the model from predicting characters that exceed our answer space, we restrict the output to the set V.

$$p([Mask] = v|x_{prompt}) = \begin{cases} \text{Softmax}\left(h_{[Mask]} \cdot \text{Emb}(v)\right) & v \in V \\ 0 & v \notin V \end{cases} \quad (1)$$

C. Answer Mapping

At this stage, the calculated predicted characters need to be mapped to the category labels of the emails. Set a mapping $\emptyset$ between the mail category y and the predicted character v. With this mapping function, we can map the predicted category label to the predicted character at the [Mask] position:

$$p(y|x) = p([Mask] = \emptyset(y)|x_{prompt}) \quad (2)$$

For example, we can map the label of normal mail to 'good' and the label of spam to 'bad'. When the pre-trained language model fills in 'good' or 'bad' on the prompt template, we can know whether the email x is spam.

Finally, according to the dataset D, the cross-entropy loss function is used to calculate the error between the correct answer and the predicted value, and all the model parameters are updated.

$$Loss = -\frac{1}{|X|}\sum_{x \in X} \log p([Mask] = \emptyset(y_x)|T(x)) \quad (3)$$

IV. EXPERIMENTS

We conduct experiments on Chinese datasets to demonstrate the effectiveness of our model in the task of email detection.

A. Datasets and Experimental Settings

We conduct experiments on public Chinese spam datasets: Trec06[20] and microblogPCU[21].

- Trec06： one of the largest and most widely used datasets for Spam Detection， including Chinese and English emails.
- microblogPCU: Data sourced from Weibo, including email text and information about these spammers.

TABLE I. STATISTICS OF DIFFERENT DATASETS

Dataset	*SPAM*	*HAM*	*Total*
Trec06	42853	21766	64619
microblogPCU	2194	3007	5201

More details of these datasets are shown in Table I shows. For all the above datasets, we use F1 scores and accuracy scores as the main metric for evaluation.

B. Experimental settings

We set the length of each text to 250 characters, excess characters are deleted, and insufficient characters are padded with 0 when converted into word embeddings. our model is optimized with Adam using the learning rate of 1e - 5 on ERNIE-1.0, with a linear warmup for the first 10% steps. For all datasets, we train our model for 20 epochs with the batch size 64. The best model checkpoint is selected based on the performance on the development set.

C. baseline models

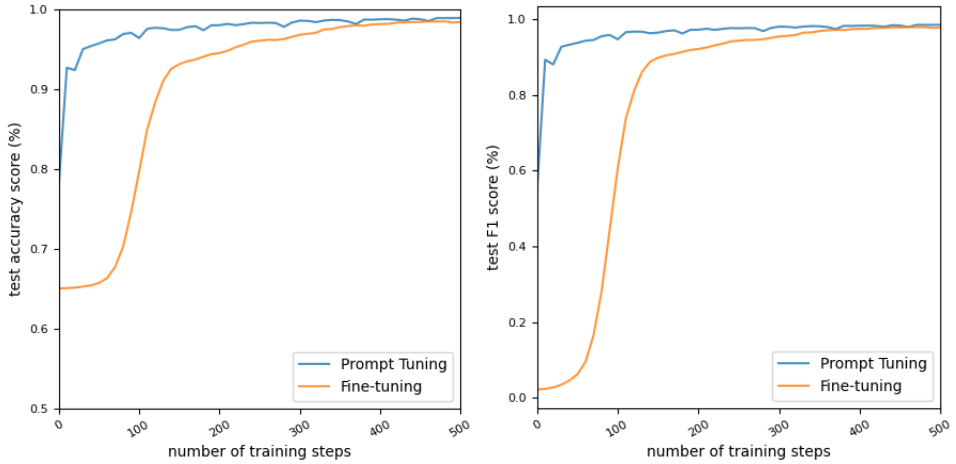
we compare our model with several typical models for text classification, including:

- Learning models from scratch: for text classification, the typical approach is learning neural models from scratch. We choose SVM [2] , Naive Bayes ,DPCNN [3], TextRNN [4], TextRCNN [22] as baselines, these models perform well in text classification tasks.
- Fine-tuning pre-trained models: PLM performs well in various NLP tasks, and many works use fine-tuning PLM for text classification. [15] showed that using PLMs is effective for spam detection, we implemented fine-tuning on ERNIE.

In addition, due to the strong semantic expression ability of PLM, we improved the deep learning techniques mentioned above by using PLM as an embedding layer, implementing ERNIE+RCNN, ERNIE+TextRNN, and ERNIE+DPCNN.

TABLE II. COMPARATIVE EXPERIMENTAL RESULTS OF OUR MODEL AND BASELINE MODELS ON DIFFERENT DATASETS.

	Models	Trec06		microblogPCU	
		Accuracy	*F1*	*Accuracy*	*F1*
Learning models from scratch	Naive Bayes	0.9758	0.9734	0.7958	0.7874
	SVM	0.9935	0.9929	0.7861	0.7834
	TextRNN	0.9882	0.9833	0.7418	0.7994
	TextRCNN	0.9943	0.9915	0.8170	0.8393
	DPCNN	0.9948	0.9926	0.8035	0.8277
Fine-tuning pre-trained models	ERNIE+TextRNN	0.9950	0.9928	0.8265	0.8495
	ERNIE+ TextRCNN	0.9953	0.9933	0.8170	0.8398
	ERNIE+ DPCNN	0.9931	0.9900	0.7765	0.8159
	Fine-tuning ERNIE	0.9951	0.9931	0.8324	0.8533
Prompt tuning pre-trained models	prompt tuning ERNIE	**0.9960**	**0.9942**	**0.8423**	**0.8633**



(a) Accuracy scores (b) F1 scores

Figure 3. Changes in accuracy scores (a) and F1 scores (b) with increasing number of training steps.

D. *Experimental results*

The experimental results are shown in Table II, From the table, we can see that:

- Using a pre-trained language model to obtain text features can significantly improve the model performance compared to methods that do not use a pre-trained model.
- For the Fine-tuning paradigm, using a pre-trained language model combined with DPCNN, TextRNN and TextRCNN have a small performance gap compared to just using a pre-trained language model and a fully connected layer for classification.
- Our model uses the prompt tuning paradigm, and its performance is better than the model using the fine-tuning paradigm. The reason for the low F1 scores and accuracy scores on the microblogPCU dataset is that the length of the text is short, and the dataset itself has problems with labeling errors.

In general, the experimental results in Table II show that our model has higher F1 scores and accuracy scores than the baseline models in the Chinese spam detection task. We believe that this is the result of the combined knowledge of the prompt template and the latent knowledge of the PLM.

We also tested the effect of training sets of different sizes on the model, as shown in Figure 4, our model performed well on both small and rich datasets. As shown in Figure 3, training the model through prompt tuning can significantly improve the convergence speed. Our model only requires less than 200 training steps to converge, while training the model with fine-tuning requires 400 training steps. Furthermore, our model had an F1 score of 0.6 before the model started training, while the comparison model only had an F1 score of 0.2. Our experiments show that the knowledge contained in the prompt template plays a positive role in both model convergence and prediction.

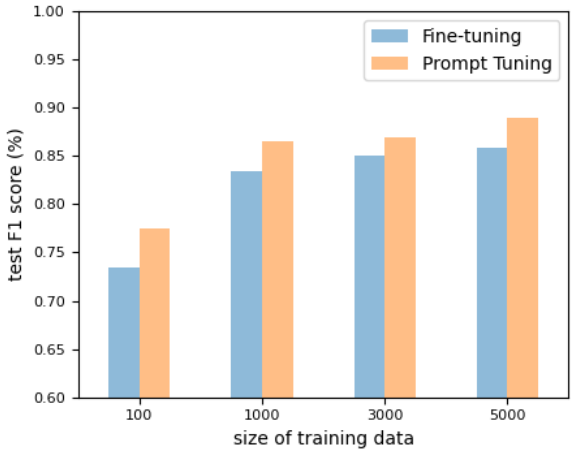


Figure 4. Comparison of F1 scores (%) when training models on different sized training sets with two paradigms.

V. CONCLUSION

In this paper, we design a Chinese spam detection model using the ERNIE pre-trained language model and the prompt tuning paradigm. By designing the prompt template, the mail classification task is converted into a filling-in-the-blank task, and the latent knowledge of the pre-trained language model and the knowledge contained in the prompt template are more fully utilized, which improves the convergence speed and prediction accuracy of the model. The experimental results show that the accuracy score of our model can reach 0.996, the F1 score can reach 0.994, which is better than the comparison model, and the convergence speed of our model is faster. In our work, the design of prompt template relies on the expertise of developers. In the future, we hope to further study the construction method of prompt templates, try to automatically generate templates and design learnable templates.

VI. ACKNOWLEDGEMENT

The authors wish to thank the Project of Inner Mongolia Science &Technology Plan under Grant No. 2021GG0164, 2020GG0186, Natural Science Foundation of China under Grant No.61862047,61962039,62162046, Inner Mongolia Science and Technology Innovation Team of Cloud Computing and Software Engineering, Inner Mongolia Engineering Lab of Cloud Computing and Service Software and Inner Mongolia Engineering Lab of Big Data Analysis Technology.

REFERENCES

[1] http://www.cac.gov.cn/2019/08/30/c_1124938750. htm.

[2] S. I. Wang and C. D. Manning, “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification,” in The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers, 2012, pp. 90–94, [Online]. Available: https://aclanthology.org/P12-2018/.

[3] R. Johnson and T. Zhang, “Deep Pyramid Convolutional Neural Networks for Text Categorization,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, 2017, pp. 562–570, doi: 10.18653/v1/P17-1052.

[4] P. Liu, X. Qiu, and X. Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning,” in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, 2016, pp. 2873–2879.

[5] S. R. Galeano, “Using BERT Encoding to Tackle the Mad-lib Attack in Spam Detection,” CoRR, vol. abs/2107.0, 2021.

[6] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing,” CoRR, vol. abs/2107.1, 2021, [Online]. Available: https://arxiv.org/abs/2107.13586.

[7] Y. Sun et al., “ERNIE: Enhanced Representation through Knowledge Integration,” CoRR, vol. abs/1904.0, 2019, [Online]. Available: http://arxiv.org/abs/1904.09223.

[8] J. S. Whissell and C. L. A. Clarke, “Clustering for semi-supervised spam filtering,” in The 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, CEAS 2011, Perth, Australia, September 1-2, 2011, Proceedings, 2011, pp. 125–134, doi: 10.1145/2030376.2030391.

[9] Z. Chen and D. Subramanian, “An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter,” CoRR, vol. abs/1804.0, 2018, [Online]. Available: http://arxiv.org/abs/1804.05232.

[10] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,” Heliyon, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[11] R. Narayan, J. K. Rout, and S. K. Jena, “Review Spam Detection Using Opinion Mining,” in Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, 2018, pp. 273–279.

[12] A. Barushka and P. Hajek, “Spam Filtering in Social Networks Using Regularized Deep Neural Networks with Ensemble Learning,” in Artificial Intelligence Applications and Innovations, 2018, pp. 38–49.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013, [Online]. Available: http://arxiv.org/abs/1301.3781.

[14] G. Jain, M. Sharma, and B. Agarwal, “Spam Detection on Social Media Using Semantic Convolutional Neural Network,” Int. J. Knowl. Discov. Bioinform, vol. 8, no. 1, pp. 12–26, 2018.

[15] J. Cao and C. Lai, “A Bilingual Multi-type Spam Detection Model Based on M-BERT,” in IEEE Global Communications Conference, GLOBECOM 2020, Virtual Event, Taiwan, December 7-11, 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9347970.

[16] [A. Vaswani et al., “Attention is All you Need,” in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 2017, pp. 5998–6008, [Online]. Available:https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[17] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained Models for Natural Language Processing: A Survey,” CoRR, vol. abs/2003.0, 2020, [Online]. Available: https://arxiv.org/abs/2003.08271.

[18] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, “PTR: Prompt Tuning with Rules for Text Classification,” 2021, [Online]. Available: http://arxiv.org/abs/2105.11259.

[19] T. Le Scao and A. M. Rush, “How many data points is a prompt worth?,” in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, [Online], 2021, pp. 2627–2636, doi: 10.18653/v1/2021.naacl-main.208.

[20] https://plg.uwaterloo.ca/˜gvcormac/treccorpus06/,2006,[Online].

[21] https://archive.ics.uci.edu/ml/datasets/microblogPCU,2015,[Online].

[22] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent Convolutional Neural Networks for Text Classification,” in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2267–2273.