Hindawi
Security and Communication Networks
Volume 2022, Article ID 4656837, 9 pages
https://doi.org/10.1155/2022/4656837

Research Article

KGDetector: Detecting Chinese Sensitive Information via Knowledge Graph-Enhanced BERT

Kai Cong,[1] **Tao Li**,[2] **Beibei Li**,[2] **Zhan Gao**,[1] **Yanbin Xu**,[1] **and Fei Gao**[2]

[1]*College of Computer Science, Sichuan University, Chengdu 610065, China*
[2]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Beibei Li; libeibei@scu.edu.cn

Received 16 January 2022; Accepted 29 March 2022; Published 19 May 2022

Academic Editor: Hao Peng

The Bidirectional Encoder Representations from Transformers (BERT) technique has been widely used in detecting Chinese sensitive information. However, existing BERT-based frameworks usually fail to emphasize key entities in the texts that contribute significantly to knowledge inference. To meet this gap, we propose a BERT and knowledge graph-based novel framework to detect Chinese sensitive information (named KGDetector). Specifically, we first train a pretrained knowledge graph-based Chinese entity embedding model to characterize entities in the Chinese textual inputs. Finally, we propose an effective framework KGDetector to detect Chinese sensitive information, which employs the knowledge graph-based embedding model and the CNN classification model. Extensive experiments on our crafted Chinese sensitive information dataset demonstrate that KGDetector can effectively detect Chinese sensitive information, outperforming existing baseline frameworks.

1. Introduction

With the development of information technology, many Chinese online social applications and platforms have emerged, such as Weibo, Tieba, and Bilibili. These social platforms have provided a place for residents to publish their information and become part of people's daily life. People of different ages, careers, countries, and ideologies share and exchange information with each other on these social platforms. As a result, these social platforms have been filled with different information and become the center for information publishing. Some researchers and journalists also claim that social platforms have replaced the status of traditional media, thus making them the most important way for each person to get information. The immediacy of these online social platforms is the key for them to replace the status of the traditional media.

However, to enable immediacy, these social platforms lack an effective review mechanism to avoid the spread of some sensitive information. A certain lack of review mechanisms has attracted some criminals and rumor mongers. Those hostile people spread illegal information which contains sensitive information about politics, terrorism, and pornography on these social platforms. Furthermore, people on these social platforms may be misguided by this information easily, especially young people who could not effectively figure out the truth. Therefore, those sensitive ones may bring about horrible results.

Indeed, there have been many existing approaches developed by former researchers and developers to detect these Chinese sensitive information. To detect sensitive information on the Internet, it is not feasible to use human resources to filter such huge information because of growing labor costs. As a result, some automatic methods have emerged. Today's online social platforms in China rely on two steps to stop the spread of sensitive information. First, it uses a sensitive information filter to stop the users' sensitive information from being published. Then, if some contents with some sensitive information are still published with the lack of a filter, the managers of the platforms will rely on reports from other benign users to withdraw the sensitive information that has already been spread out. Though the sensitive information can be withdrawn in Step 2, it is already spread out. The current filter method in Step 1 mainly

relies on keyword matching, which requires building a keyword dictionary by a human [1,2]. Building a keyword dictionary by a human is hard to cover all the keywords, which may bring about a high false-negative rate, while the keyword matching itself will mislabel some benign content as sensitive, resulting in a high false-positive rate [3,4]. Furthermore, the keyword matching-based detection is also very easy to bypass [5–7]. Hence, many approaches using BERT for classification are studied. Ding et al. built a corpus to train the detection model and applied the BERT model in this detection problem [8]. Though contextualized information could be well extracted by BERT, we argue that many significant entities are not well emphasized as they should be. In the meantime, the knowledge graphs have been used in many research areas of natural language processing. The knowledge graphs can help the researchers identify the different entities and uncover the relationship between different entities very well. Thus, knowledge graphs can be a good tool for us to identify the sensitive entity in the Chinese text. To this end, we propose to detect Chinese based on both BERT and knowledge graph embeddings, which could make use of pretrained knowledge graphs to generate embeddings for named entities. Our main contributions are threefold:

(i) First, we introduce knowledge graph to enrich the input of the classifier. An entity embedding model based on knowledge graph is trained for characterizing entities in the textual inputs, which can significantly improve the model performance.

(ii) Second, we propose an effective framework, named KGDetector, to detect Chinese sensitive information, which employs knowledge graph-based entity embedding model and a convolutional neural network (CNN)-based model to classify the encoded intermediate information.

(iii) Third, we build a Chinese sensitive information dataset based on Chinese Wikipedia, and extensive experiments on this dataset demonstrate that our proposed KGDetector framework outperforms typical frameworks on Chinese sensitive information detection tasks.

2. Related Works

In this section, we introduce the related work about the classification of sensitive information and BERT for text classification.

2.1. Sensitive Information Detection. As information spreads all over the world, the detection of sensitive information has become a more and more important research topic. In 2015, Berardi et al. identified classified text using the sensitive information keyword matching technique [9]. Because the keyword dictionary was created artificially, subjective influence might have an impact on the classification accuracy. To detect sensitive information, [10] used a recursive neural network in 2017. By studying the syntax and grammatical structure of the text, this approach uncovered sensitive information in text documents. The sensitivity values of the semantic parts of the text structure were assessed. Furthermore, to capture the intricacy of recognizing sensitive information, they created a sensitive phrase recursive neural network in 2018 [11]. In the same year, Xu et al. introduced a new topic tracking algorithm, which monitored sensitive words during a period of time [12]. The tracking algorithm's initial step was to calculate the weight of sensitive words over a set period of time and identify the top 10 sensitive terms. The second stage was to choose the top three sensitive terms out of a total of ten sensitive words to track. By using high-frequency words as characteristics of the text obtained by TF-IDF, [13] increased the detection accuracy in 2018. The TF-IDF model was used to classify confidential information. In 2019, Xu et al. proposed a new method to use TextCNN in the task of sensitive information detection [14]. It could keep the accuracy of detection, while lowering the training construction time of the detection model. As a result, it could achieve efficient and accurate detection. In the same year, Wang et al. presented a sensitive information classification model based on BERT-CNN [15]. In 2020, Lin et al. proposed a reliable method to extract the characteristics of the data more comprehensively to obtain better detection results [16]. The framework was based on the structure of BiLSTM and CNN. A convolutional neural network was used to extract local features effectively, and a BiLSTM network was built to extract global features of unstructured documents. In the same year, [17] designed a new framework to detect sensitive information via network traffic restore solution. In 2020, to safeguard personal data privacy, [18] used a BERT-based sequence labeling algorithm to discover and delete sensitive data in Spanish clinical literature. In 2021, Ding et al. built a corpus to train the detection model and applied the BERT model in this detection problem [8]. With BERT applied in this problem, more popular NLP methods were implemented in their framework to optimize the accuracy as well. In the same year, Gan et al. designed a scalable multichannel architecture of convolutional neural network and bidirectional long short-term memory (CNN-BiLSTM) model to detect sensitive information in Chinese text [19]. The attention mechanism was introduced to enhance the performance of the model.

2.2. BERT-Based Text Classification. In 2019, Sun et al. conducted abundant experiments to evaluate different fine-tuning techniques of BERT on text classification task [20]. A general solution for BERT fine-tuning was proposed to obtain new state-of-the-art results on eight widely studied text classification datasets. In the same year, Chen et al. presented a joint intent classification and slot filling model based on BERT, which addressed the poor generalization capability of traditional NLU models [21]. This joint model outperformed the BERT model on intent classification accuracy, slot filling F1, and sentence-level semantic frame accuracy. In 2019, Ostendorff et al. built a deep neural language model based on BERT [22]. The model combined text representations of metadata with knowledge graph embeddings of author information. The model achieved

better performance on the classification task compared with the standard BERT framework. In 2020, Jose and others proposed the VGCN-BERT model based on the combination of BERT and a Vocabulary Graph Convolutional Network (VGCN). VGCN-Bert combines local information and global information to build together a final representation for classification. In the same year, Munikar et al. utilized BERT to solve the fine-grained sentiment classification task [23]. They also showed that transfer learning is successful in natural language processing. In 2020, Su et al. presented a deep learning method named Bidirectional Encoder Representations based on BERT to classify genetic mutations using the text evidence from an annotated database [24]. In 2021, [25]designed a BERT-enhanced text graph neural network (BEGNN) model. They created a text graph for each document based on the co-occurrence connection of terms and used GNN to extract text features. Furthermore, BERT was utilized to extract semantic characteristics. In the same year, Zhang et al. proposed a multilayer self-attention model combined with BERT to cope with aspect category and word attention at different granularities [26].

3. The KGDetector Framework

In this section, we elaborate on the KGDetector framework, including the overview, text encoder, and classifier.

3.1. Overview. The hard-coded limit of BERT tokens is 512, but the number of Chinese characters in an entire Chinese Wikipedia entry usually exceeds this upper boundary. Hence, in KGDetector, for each entry, we consider its title and abstract as the input information (in Section 4.1, we find that the average length of the title and abstract is not greater than 512). As shown in Figure 1, given an input text, we first encode it with our crafted encoder, which is composed of a fine-tuned BERT unit and two units for knowledge graph embedding. Then, with encoded information, we further design a CNN model to classify it and derive an output as the inferred label.

3.2. Text Encoder. The text encoder aims to obtain encoded intermediate information that could effectively represent valuable knowledge of the inputted text. To this end, apart from encoding the text input with a fine-tuned BERT, we extract named entities from abstract text and embed them according to knowledge graph embeddings to derive more representative intermediate information.

3.2.1. BERT-Based Textual Information Encoder. We utilize the BERT to acquire contextualized representations from original text consisting of the entry tile and corresponding abstract. Though the initial pretrained BERT[1] provided by Google supports multilingual embeddings including Chinese, compared with multilingual BERT, Chinese BERT could achieve much better performance on Chinese NLP tasks [27]. To derive better performance, we take the BERT trained on Chinese Wikipedia containing both simplified and traditional Chinese text[2]. Unless otherwise mentioned, we utilize the Chinese BERT by default. Specifically, there are 12 layers in the BERT, where each layer contains 768 dimensions. Hence, we could derive an intermediate semantic representation $\mathbf{R}_{\text{BERT}}$ as

$$\begin{aligned} \mathcal{X} &= [CLS]\mathbf{t}_1\mathbf{t}_2\ldots\mathbf{t}_N, \\ \mathbf{R}_{\text{BERT}} &= \text{BERT}(\mathcal{X}), \end{aligned} \tag{1}$$

where $\mathcal{X}$ is the textual input and $\mathbf{R}_{\text{BERT}}$ is a 768-dimensional vector.

3.2.2. Knowledge Graph-Based Entity Embedding. Though contextualized information of the inputted text could be well represented by $\mathbf{R}$, we argue that some valuable entities may not be emphasized. In particular, we aim to use the information of named entities in the abstract as auxiliaries to construct a more representative intermediate vector and enhance more effective sensitive information detection. For instance, if Bruce Lee[3]'s name appears in the abstract, then the article corresponding to this abstract should be more inclined to introduce the content of kung fu.

To this end, we train an entity embedding model on a Chinese knowledge graph CN-DBpedia[4] to represent named entities in the abstract with their entity embeddings. Following previous work [28], we set the knowledge graph-based entity embeddings to be 200-dimensional. Specifically, given training dataset with triplets as

$$D = \{(a, r, b)\}, \tag{2}$$

which is composed triplets with entities a and b from entities set E, and relation r is from relations set R. Given an embedding function $\mathscr{E}(\cdot)$, we embed the entities and relations as $\mathscr{E}(a)$, $\mathscr{E}(b)$, and $\mathscr{E}(r)$. We denote such embeddings as $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{r}$. After randomly initializing the embedding function $\mathscr{E}(\cdot)$, we optimize an equation to derive an effective embedding model

$$\min_{\mathscr{E}\in\mathbb{R}^d} \sum_{(a,r,b)\in D} \sum_{(\underline{a},r,\underline{b})\in \underline{D}} \text{ReLU}\left(\eta + \|\mathbf{a}+\mathbf{r}-\mathbf{b}\| - \|\underline{\mathbf{a}}+\mathbf{r}-\underline{\mathbf{b}}\|\right), \tag{3}$$

where

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

and $\|\cdot\|$ represents the ℓ_2 norm. It is worth noting that η is a positive margin (we set it to be 2 in our experiments), and $\underline{D}$ is a corrupt dataset with unmatched triplets:

$$\underline{D} = \{(\underline{a}, r, b) | \underline{a} \in E\} \cup \{(a, r, \underline{b}) | \underline{b} \in E\}. \tag{5}$$

Finding optimal solutions $\mathscr{E}^*$ for (3), we take $\mathscr{E}^*$ as embedding models for entities in the abstract. Formally, given a list of entities $\{e_1, e_2, \ldots, e_M\}$, we embed them to $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$ based on $\mathscr{E}^*$. The number of entities in different abstracts may not be the same, whereas the sizes of vectors fed into the classifier should be consistent. Considering that the entities yield a sparse distribution in the

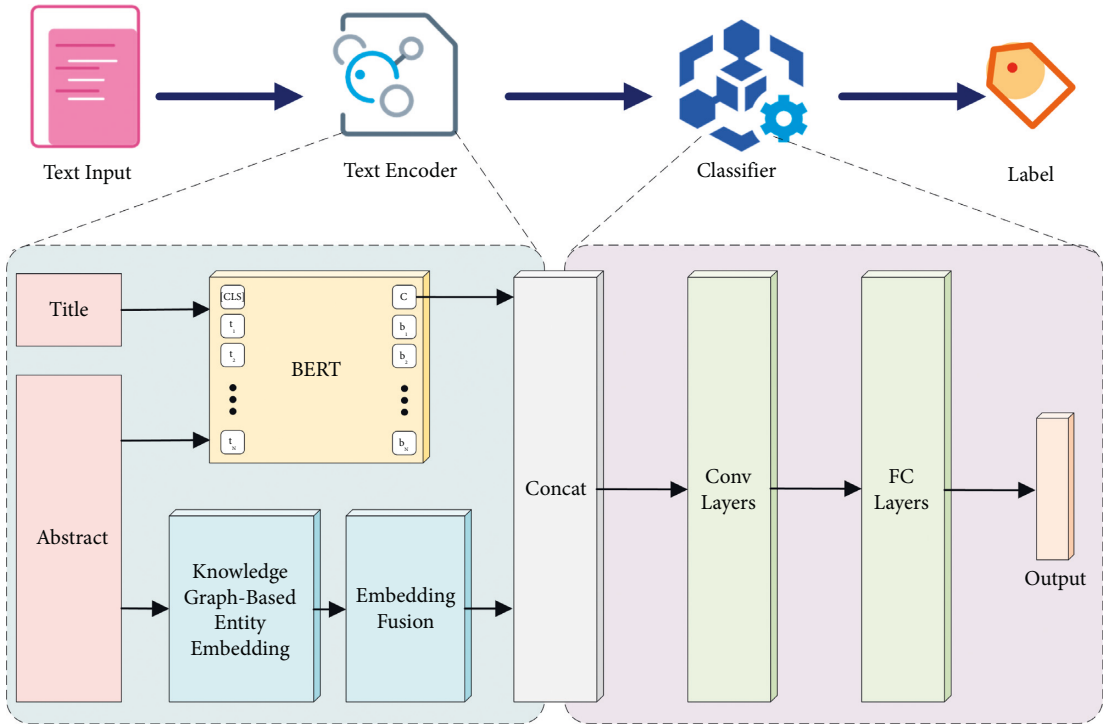

Figure 1: The workflow of the KGDetector framework.

embedding space, we compute an embedding $\mathbf{e}$ to represent each embedding in $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$ as

$$\mathbf{e} = \sum_{i=1}^{M} \left(\frac{\mathbf{e}_i}{M}\right). \tag{6}$$

Then, we concatenate the embedding $\mathbf{e}$ and the semantic representation $\mathbf{R}_{\text{BERT}}$ as the intermediate vector $\mathbf{I}$ that will be fed into the classifier.

3.3. Classifier. Recall that the output of the text encoder $\mathbf{I}$ is composed of representation $\mathbf{R}_{\text{BERT}}$ from BERT (768 dimensions) and vector $\mathbf{e}$ (200 dimensions). As shown in Figure 2, we train a convolutional neural network (CNN) to classify concatenated embeddings. In KGDetector, we, respectively, employ three convolutional layers and two fully connected (FC) layers to comprehensively extract information of embeddings and make generalized classification. Furthermore, we deploy an activation function for the 2-dimensional output. During model training, the softmax function is used. During inference time, the softmax function is replaced by the one-hot encoding.

4. Experiments

4.1. Experimental Settings. We conduct our experiments on a computer with Windows 11, Intel(R) Core(TM) i7-9750F CPU 2.60 GHz, and an NVIDIA GeForce RTX 3090 GPU. The deep learning models are implemented using PyTorch[5].

We evaluate the KGDetector on Chinese Wikipedia[6]. The original texts are traditional Chinese; to simplify, we first utilize OpenCC[7] to convert them into simplified Chinese and perform an additional text cleaning step to keep only Chinese words. Each entry (i.e., page) of Wikipedia mainly consists of four parts, i.e., title, abstract, article, and links. We build the groundtruth based on the article while making prediction based solely on the title along with the abstract. Specifically, we first collect open-source sensitive keywords lists[8][9] as sensitive identifier. Then, we filter the article of each entry with the sensitive keywords lists to identify a set of benign articles, which have no sensitive keywords, and a set of potentially sensitive articles, which have at least one of the sensitive keywords. We further manually check the potentially sensitive entries to confirm their sensitiveness. We split the dataset into training set, validation set, and test set with a ratio of 7 : 2:1. Table 1 shows the number of samples of each set. Table 2 presents the parameter settings of the KGDetector. Table 3 displays the average length, i.e., number of words, of title and abstract of the selected Wikipedia entries.

In this work, we compare the KGDetector with baseline, i.e., filter title and abstract directly using sensitive words lists, and state-of-the-art studies on Chinese sensitive information detection [8,15,19]. Ding et al. built a corpus to train the detection model and applied the BERT model in this detection problem [8]. Wang et al. presented a sensitive information classification model based on BERT-CNN [15]. Gan et al. designed a scalable multichannel architecture of convolutional neural network and bidirectional long short-term memory (CNN-BiLSTM) model with the attention mechanism to detect sensitive information in Chinese text [19]. Furthermore, we also compare the performance of the KGDetector under different data sources: title, abstract, title with abstract, first N words in the article, and last N words in the article. Further, we also compare it with popular text classification models, i.e., TextCNN and BiLSTM.

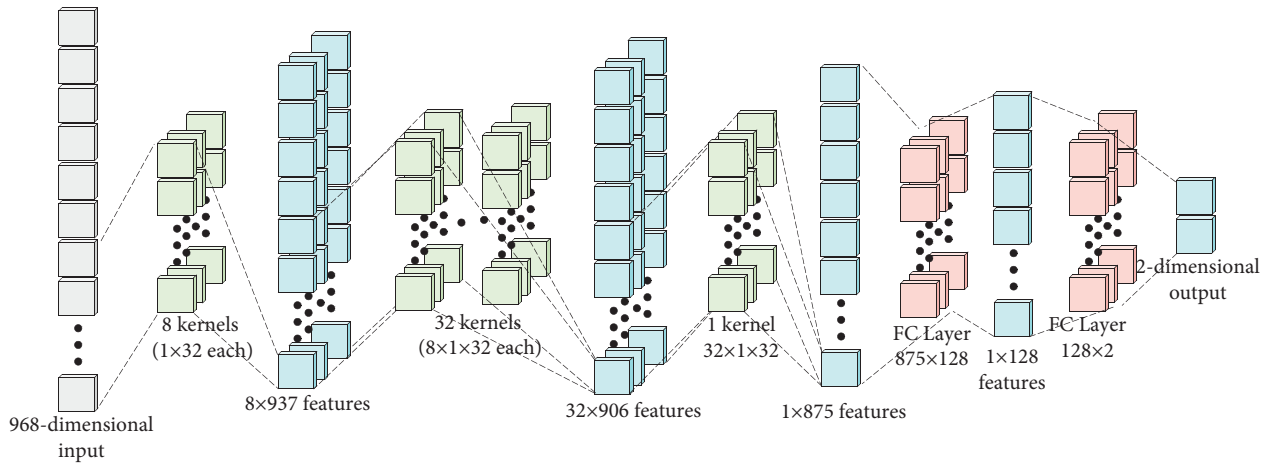

Figure 2: Illustration of the classifier.

Table 1: Number of samples of the dataset.

Type	Training set	Validation set	Test set
Benign entries	2940	420	847
Sensitive entries	2940	420	847

Table 2: Parameter settings of the KGDetector.

Batch size	Learning rate	No. of training epochs
64	0.001	50
Dropout rate	Optimizer	Activation function
0.5	SGD	ReLu

Table 3: Length information of the Wikipedia entries.

Parts	Min	Avg	Max	95% quantile
Title	1	5	53	12
Abstract	6	174	822	258
Article	74	2766	50791	30980

To evaluate the performance of the KGDetector, we consider four metrics, namely, (1) accuracy: the correct proportion of the classification results; (2) precision: the proportion of entries classified as sensitive that are indeed sensitive; (3) recall: the proportion of sensitive entries that are classified as sensitive; and (4) F1-score: the weighted average of precision and recall. They are defined in equations (7)–(10), where TP stands for true positive, TN stands for true negative, FP stands for false positive, and FN stands for false negative.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{7}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{10}$$

4.2. Performance Evaluation

4.2.1. Comparison with Different State-of-the-Art Studies. We compare our KGDetector with the state-of-the-art schemes proposed by other researchers. If a framework achieves better performance than other frameworks in all four metrics: accuracy, precision, recall, and F1-score, this framework outperforms other frameworks. As shown in Figure 3, the evaluation comparison proves that the KGDetector outperforms other state-of-the-art studies, respectively, in all the four metrics.

4.2.2. Comparison under Different Data Sources. Furthermore, different data sources: title, abstract, title with abstract, first N words in the article, and last N words in the article, are considered in our experiment as well. As shown in Figure 4, the model using the title text with abstract text as the data source outperforms other models with different data sources. The model using title as the data source is the weakest model among all the models. This indicates that the information contained in the title text is not rich enough to help the model classify. In practice, the author of the sensitive article will also try not to add the sensitive information in the title which may cause his sensitive article to be detected easily. The models with the first N words or the last N words also perform badly in detecting sensitive information. It is claimed that the first N words or the last N words cannot effectively summarize all the information contained in the article because some sensitive information may not be in the first N words or the last N words. However, we can see that the performance of the model using abstract as the data source can nearly reach the performance of the model using the title with abstract as data sources. It can be concluded that the text in the abstract contains the most information about the article as the author will try to summarize the whole article in the abstract firmly.

4.2.3. Comparison under Different Sizes of Training Dataset. When we train a sensitive information detection model, the size of training dataset matters. If a framework needs less

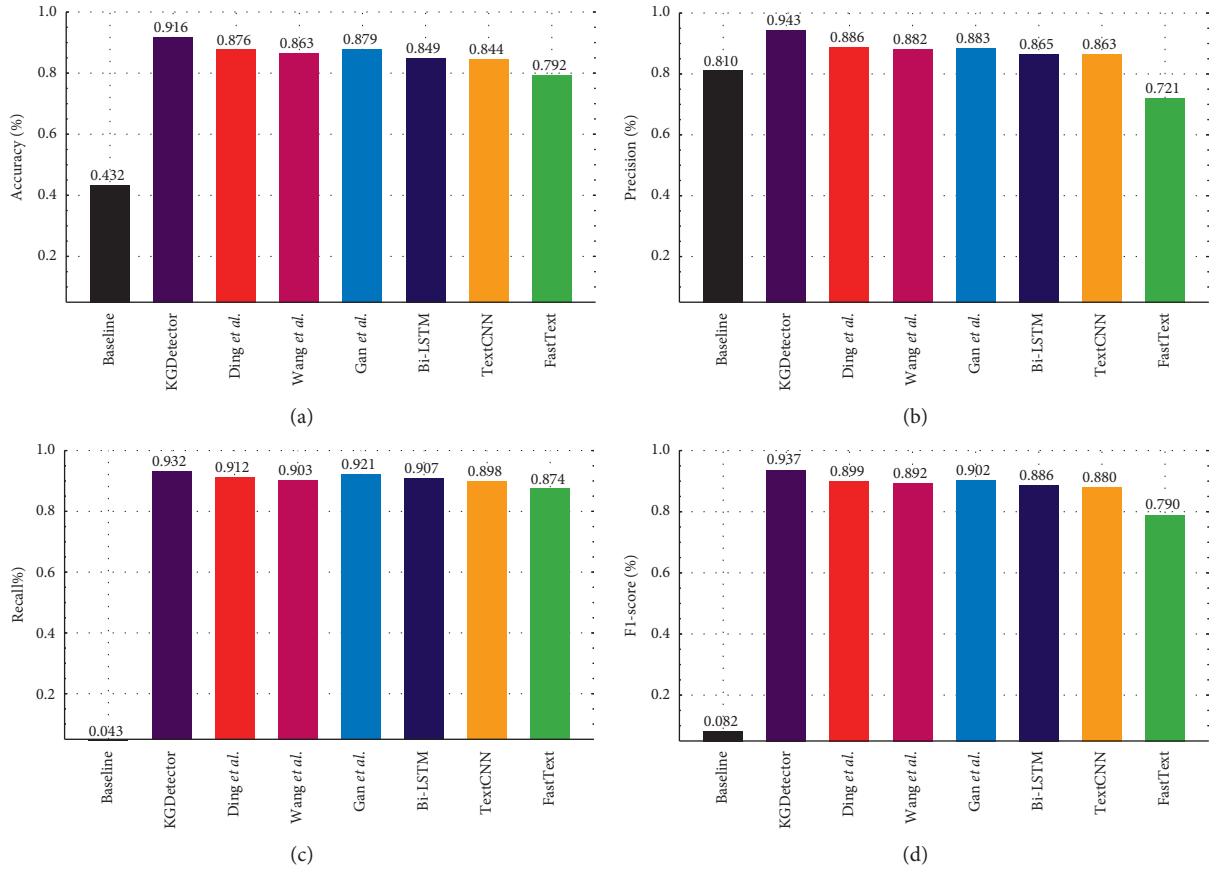

Figure 3: Performance comparison (a) Accuracy. (b) Precision. (c) Recall. (d) F1-score.

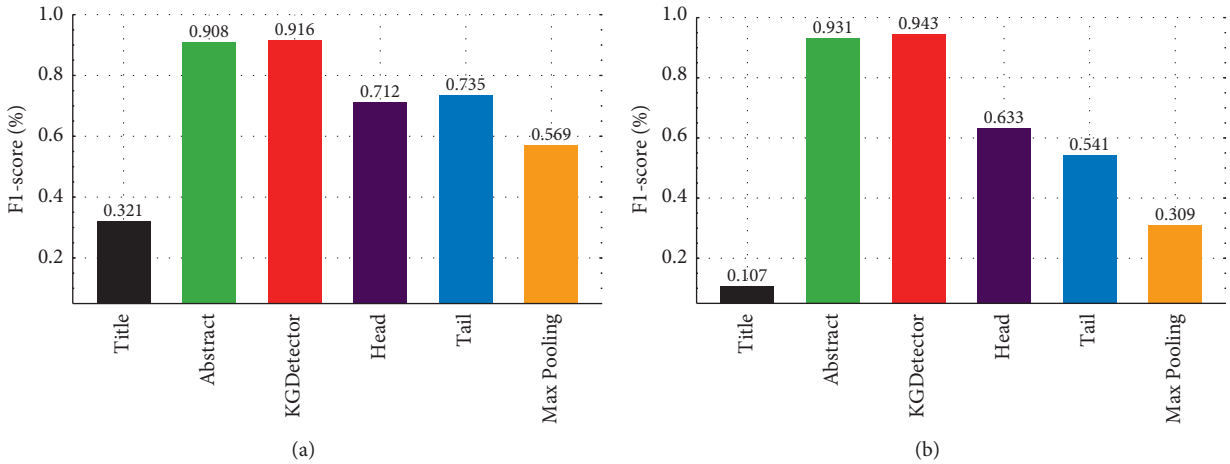

Figure 4: Continued.

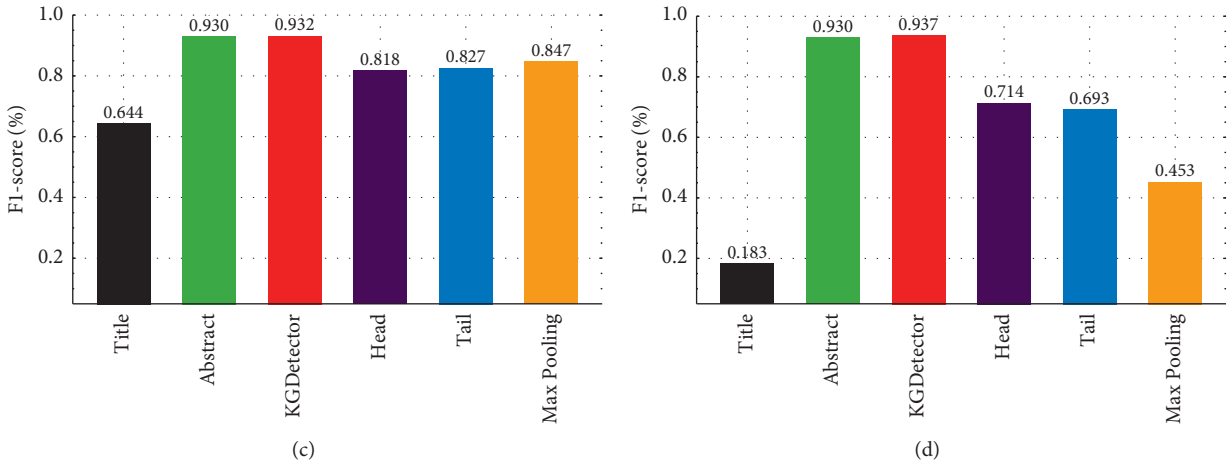

Figure 4: Comparison of different data sources (a) Accuracy. (b) Precision. (c) Recall. (d) F1-score.

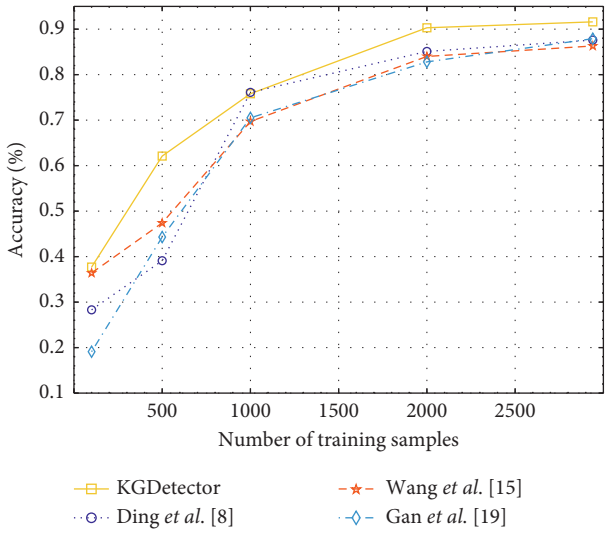

Figure 5: Accuracy with varying sizes of training dataset.

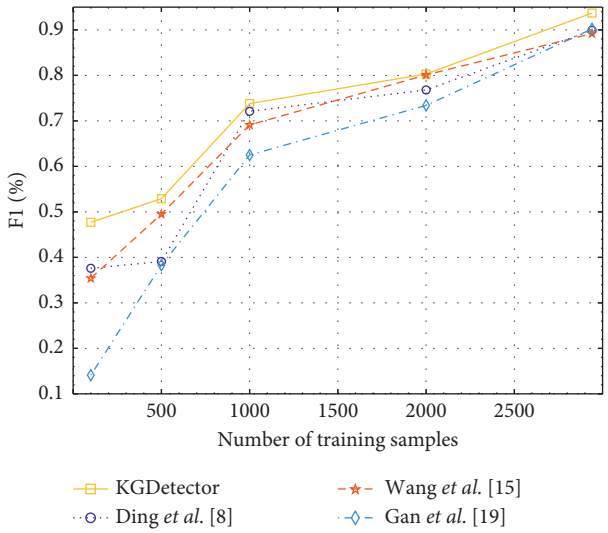

Figure 6: F1-score with varying sizes of training dataset.

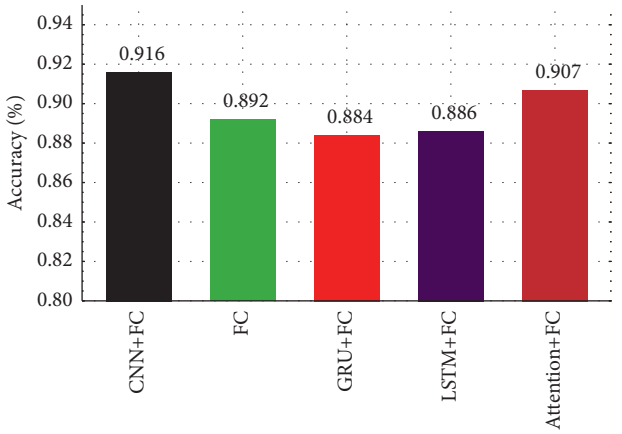

Figure 7: Accuracy of different classifiers.

training data to achieve a certain performance or achieves higher accuracy with a fixed size of training dataset, it outperforms other frameworks. As shown in Figures 5 and 6, compared with Wang et al.*et al*, Ding et al.*et al*, and Gan et al.*et al*, KGDetector costs less training data to achieve a certain accuracy. For instance, to receive an accuracy greater than 0.90, KGDetector needs about 2000 data samples, while others may require more than 2500 data samples. Besides, given a fixed size of training data, KGDetector always reaches the highest accuracy among all schemes. For instance, with 500 training data samples, the accuracy of KGDetector is about 0.62, while that for other frameworks is smaller than 0.50.

4.2.4. Comparison with Different Classifiers. We further consider using different classifiers in our framework. Four types of classifiers, convolution layer + fully connected layer (CNN + FC), fully connected layer (FC), gated recurrent unit + fully connected layer (GRU + FC), and attention + fully connected layer (attention + FC), are considered. As can be seen from Figures 7 and 8, our evaluation results show that CNN + FC is the best classifier among the four.

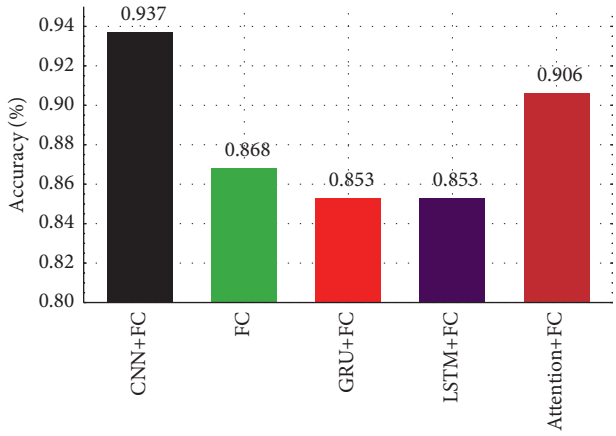

Figure 8: F1-score of different classifiers.

5. Conclusion

In this paper, we have proposed a novel framework, named KGDetector, to detect Chinese sensitive information based on knowledge graph-enhanced BERT. Specifically, we trained a pretrained knowledge graph-based entity embedding model to generate knowledge graph embeddings, which can enrich the input of the classifier. Then, an effective framework KGDetector, which employs the knowledge graph-based embedding model and the CNN classification model, was designed to detect Chinese sensitive information. Extensive experiments on our crafted Chinese sensitive information dataset demonstrate that the proposed KGDetector outperforms existing frameworks in terms of accuracy, precision, recall, and F1-score in detecting Chinese sensitive information. Our future work will extend the framework to multiple languages.

Data Availability

The dataset of this work is constructed by publicly accessible wikipedia data in https://dumps.wikimedia.org/zhwiki/2021.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by the National Key Research and Development Program of China (no. 2020YFB1805400), in part by the National Natural Science Foundation of China (no. U19A2068), and the Sichuan Science and Technology Program (no. 2022YFG0193).

References

[1] K.-W. Fu, C.-H. Chan, and M. Chau, "Assessing censorship on microblogs in China: discriminatory keyword analysis and the real-name registration policy," *IEEE Internet Computing*, vol. 17, no. 3, pp. 42–50, 2013.

[2] Y. Kou, B. Semaan, and B. Nardi, "A confucian look at internet censorship in China," in *Human-Computer Interaction (INTERACT)*, R. Bernhaupt, G. Dalvi, A. Joshi, D. K. Balkrishan, J. O'Neill, and M. Winckler, Eds., Springer International Publishing, Berlin, Germany, pp. 377–398, 2017.

[3] J. Matthes and M. Kohring, "The content analysis of media frames: toward improving reliability and validity," *Journal of Communication*, vol. 58, no. 2, pp. 258–279, 2008.

[4] D. Nobori and Y. Shinjo, "VPN Gate: a Volunteer-Organized public VPN relay system with blocking resistance for bypassing government censorship firewalls," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 229–241, USENIX Association, Seattle, WA, April 2014.

[5] S.-y. Lee, "Surviving online censorship in China: three satirical tactics and their impact," *The China Quarterly*, vol. 228, pp. 1061–1080, 2016.

[6] Y. Mou, K. Wu, and D. Atkin, "Understanding the use of circumvention tools to bypass online censorship," *New Media & Society*, vol. 18, no. 5, pp. 837–856, 2016.

[7] J. Holowczak and A. Houmansadr, "CacheBrowser: bypassing Chinese censorship without proxies using cached content," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 70–83, Association for Computing Machinery, New York, NY, USA, October 2015.

[8] M. Ding, X. Wang, C. Wu, K. Wang, and X. Yang, "Research on automated detection of sensitive information based on BERT," *Journal of Physics: Conference Series*, vol. 1757, no. 1, Article ID 012088, 2021.

[9] G. Berardi, A. Esuli, C. Macdonald, I. Ounis, and F. Sebastiani, "Semi-automated text classification for sensitivity identification," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1711–1714, CIKM, Melbourne Australia, October 2015.

[10] J. Neerbeky, I. Assentz, and P. Dolog, "TABOO: detecting unstructured sensitive information using recursive neural networks," in *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 1399-1400, IEEE, San Diego, CA, USA, April 2017.

[11] J. Neerbek, I. Assent, and P. Dolog, "Detecting complex sensitive information via phrase structure in recursive neural networks," in *Proceedings of the Pacific-asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 373–385, Springer, Melbourne, VIC, Australia, June 2018.

[12] G. Xu, Z. Yu, and Q. Qi, "Efficient sensitive information classification and topic tracking based on Tibetan web pages," *IEEE Access*, vol. 6, pp. 55643–55652, 2018.

[13] T. Katić and N. Milićević, "Comparing sentiment analysis and document representation methods of amazon reviews," in *Proceedings of the International Symposium on Intelligent Systems and Informatics (SISY)*, September 2018.

[14] G. Xu, C. Qi, H. Yu, S. Xu, C. Zhao, and J. Yuan, "Detecting sensitive information of unstructured text using convolutional neural network," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 474–479, CyberC, Guilin, China, October-2019.

[15] Y. Wang, X. Shen, and Y. Yang, "The classification of Chinese sensitive information based on bert-cnn," in *Proceedings of the International Symposium on Intelligence Computation and Applications (ISICA)*, pp. 269–280, Springer, Berlin, Heidelberg Germany, 2019.

[16] Y. Lin, G. Xu, G. Xu, Y. Chen, and D. Sun, "Sensitive information detection based on convolution neural network and bi-directional lstm," in *Proceedings of the International Conference on Trust, Security and Privacy in Computing and*

Communications, pp. 1614–1621, TrustCom, Guangzhou China, January 2020.

[17] Q. Hong, T. Zheng, L. Wenli, T. Jianwei, and Z. Hongyu, "A sensitive information detection method based on network traffic restore," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 832–836, IEEE, Phuket, Thailand, February 2020.

[18] A. G. Pablos, N. Pérez, and M. Cuadros, "Sensitive data detection and classification in Spanish clinical text: experiments with bert," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pp. 4486–4494, Marseille, France, May 2020.

[19] C. Gan, Q. Feng, and Z. Zhang, "Scalable multi-channel dilated CNN-BiLSTM model with attention mechanism for Chinese textual sentiment analysis," *Future Generation Computer Systems*, vol. 118, pp. 297–309, 2021.

[20] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?, Lecture Notes in Computer Science," in *Proceedings of theChina National Conference on Chinese Computational Linguistics (CCL)*, pp. 194–206, Springer, Changsha, China, 2019.

[21] Q. Chen, Z. Zhuo, and W. Wang, "Bert for Joint Intent Classification and Slot Filling," 2019, https://arxiv.org/abs/1902.10909.

[22] M. Ostendorff, P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, and B. Gipp, "Enriching Bert with Knowledge Graph Embeddings for Document Classification," arXiv preprint arXiv: https://arxiv.org/abs/1909.08402, 2019.

[23] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using bert," *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1, pp. 1–5, 2019.

[24] Y. Su, H. Xiang, H. Xie et al., "Application of bert to enable gene classification based on clinical evidence," *BioMed Research International*, vol. 2020, pp. 1–13, 2020.

[25] Y. Yang and X. Cui, "Bert-enhanced text graph neural network for classification," *Entropy*, vol. 23, no. 11, 1536 pages, 2021, [Online]. Available:.

[26] X. Zhang, X. Song, A. Feng, and Z. Gao, "Multi-self-attention for aspect category detection and biomedical multilabel text classification with bert," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–6, 2021.

[27] Y. Cui, W. Che, T. Liu et al., "Pre-training with Whole Word Masking for Chinese Bert," 2019. arXiv preprint arXiv: https://arxiv.org/abs/1906.08101.

[28] A. Lerer, L. Wu, J. Shen et al., "PyTorch-BigGraph: a large-scale graph embedding system," in *Proceedings of the Conference on Machine Learning and Systems (MLSys)*, Palo Alto, CA, USA, 2019.