SICM: A Supervised-based Identification and Classification Model for Chinese Jargons Using Feature Adapter Enhanced BERT

Abstract. In recent years, cybercriminals in darknet markets are becoming increasingly rampant and they usually converse in jargons to avoid surveillance. These jargons distort the original meaning of innocent-looking words, which makes it difficult for network regulators to understand the true meaning of dialogues content. Existing studies mainly explored jargon detection based on unsupervised methods. However, those solutions generally face significant challenges in setting appropriate jargon evaluation thresholds. To the best of our knowledge, we are the first to propose a **S**upervised-based jargon **I**dentification and **C**lassification **M**odel (SICM). Specifically, we transform jargon detection into a sequence labeling problem. Firstly, we construct and publish the first labeled dataset of Chinese jargons, which divides jargons into seven categories. Then, in order to better represent the unique characteristics of Chinese jargon and facilitate more effective feature fusion, we innovatively propose a Chinese jargon identification and classification model based on Feature Adapter enhanced BERT, which uses attention mechanism to integrate phonetic, glyph and lexical features into the lower layers of BERT. In addition, we also design a global attention layer, which allows the model to grasp the global information of the sentences. The experimental results demonstrate that our model outperforms existing state-of-the-art jargon detection methods, with an F1-score of 91.99%. This study provides a brand-new research idea for the Chinese jargon detection in the darknet marketplace.

Keywords: Chinese jargons · Bert adapter · Supervised learning · Sequence labeling · Feature fusion.

1 Introduction

Background. Due to the anonymity of the dark web, cybercriminals in darknet markets are becoming increasingly rampant [8]. They usually conduct illegal transactions in underground markets, such as spreading gambling information, leaking personal private data through hacking, selling firearms and drugs in regulated areas. These activities seriously endanger the normal order of cyberspace and real society.

However, because of the obscurity of information in the darknet underground markets [8], it is a significant challenge for its regulation. Cybercriminals often use jargons to disguise transactions and avoid surveillance. Jargons distort the original meaning of innocent-looking words. It is not easy to discover the harmful

implications of jargons, which makes cybercrime monitoring and investigation very difficult. For example, ‘‘猪肉” (pork) is a word that is common in real life, but it may refer to ‘‘冰毒” (methamphetamine) in the underground market; ‘‘轨道料” (rail material) actually refers to the bank card information obtained by modifying the POS (Point of Sale) machine. Due to these characteristics of jargons, it is quite difficult for network regulators to grasp what the cybercriminals are saying. The research work on automatic jargons identification and classification is of significant importance for network regulators to better understand the various types of cybercrime.

There have been several studies using unsupervised methods to detect jargons [24, 19, 21, 16, 17, 7, 26, 25]. The main idea of their works is to compare the similarity of word embeddings of the same word in different corpora. Usually, for a specific word, its word embeddings are generated not only in a normal corpus, but also in a malicious corpus in which words may have hidden harmful meanings. Then, the similarity of the same word's two different embeddings will be calculated. The value of the similarity is the key indicator to determine whether the word is a jargon [21, 16, 17, 7]. Besides, there are also some unsupervised methods that use BERT-based MLM (Masked Language Model) to predict jargons [26, 25].

Challenges. All the existing research works of jargon detection use unsupervised methods, and they generally face the following significant challenges.

Firstly, in their jargon identification algorithms, it is challenging to set the appropriate threshold for determining the jargons. Existing unsupervised approaches set a threshold for the result of similarity calculation [21, 16, 17, 7] or the result predicted by the Masked Language Model [26, 25] to obtain the jargon candidates. Nevertheless, the threshold value is usually set subjectively based on experience, and meanwhile the results of jargon prediction are extremely sensitive to the threshold value. If the threshold value is not set properly, it can easily cause normal words to be incorrectly recognized as jargons. Although this problem can be avoided by using the supervised approach, there is no precedent research on jargon identification based on supervised methods, to the best of our knowledge.

Secondly, for Chinese jargon detection, previous methods also have some other problems: most of them only consider the textual information and do not extract the linguistic features of Chinese jargons [24, 19, 17, 7]. Actually, there are many Chinese jargons created based on the similarity of character pronunciation or morphology. For instance, members of gambling websites use the word ‘‘菠菜” (spinach) instead of ‘‘博彩” (Gambling), which are completely different words but very similar in their Chinese pronunciations. Another example is that people often use the word ‘‘果” (fruit) instead of ‘‘裸” (nude) when it comes to pornography. The two words are pronounced differently but have similar Chinese character roots. Therefore, we believe that using an appropriate way to introduce information about the similarity of pronunciation and morphology between different Chinese characters can be of great help in Chinese jargon detection.

Contributions. To address the above challenges, we propose the first supervised-based Chinese jargon detection model (SICM[1]) that can identify and classify Chinese jargons automatically. Firstly, we construct and publish the first labeled dataset of Chinese jargons called CJC[2], which divides jargons into seven categories, including 'Drug', 'Gambling', 'Pornography', 'Violence', 'Fraud', 'Hacking' and 'Others'. Then, we transform jargon detection into a sequence labeling problem and innovatively propose a Chinese jargon identification and classification model based on Feature Adapter enhanced BERT, which uses attention mechanism to fuse phonetic, glyph and lexical features. In addition, we also design a global attention layer, which allows the model to grasp the global information of the sentences.

The main contributions of this paper are summarized as follows:

- We construct and publish the first labeled dataset of Chinese jargon (CJC[2]), to the best of our knowledge. The data comes from Chinese underground forums on the darknet where cybercriminal activities are active. Our dataset contains 33,668 sentences, including 19,675 sentences containing jargons and 13,993 normal sentences. A total of 45,079 jargons were labeled, including 1,796 unduplicated jargons.
- We design and publish the first Chinese jargon identification and classification model based on a supervised approach (SICM[1]), to the best of our knowledge. We transform the jargon detection into a sequence labeling problem. For an input sentence, the location of the jargons in the sentence and the crime category they belong to can be identified.
- We extract brand-new features based on the unique characteristics of Chinese jargons. Considering the characteristics of pronunciation and morphology of Chinese jargons, we extract phonetic features and glyph features for each Chinese character. Meanwhile, lexical features are added to the model in order to learn the boundary information related to jargons. The experimental results show that these features can significantly improve the effect of jargon detection.
- We innovatively designed Feature Adapter enhanced BERT, which can integrate various jargon features into the lower layers of BERT [3] directly by modifying the structure of it. Based on an attention mechanism, the Feature Adapter fuses phonetic, glyph, and lexical features between two Transformer layers of BERT. Experimentally, this feature fusion approach is more effective than traditional method.

2 Related Work

Until now, jargon detection is a relatively new research area. Existing researches mainly focus on Chinese [24, 19, 17, 7], English [21, 26, 25] and Japanese [16].

[1] https://github.com/Y1fl/SICM
[2] https://github.com/Y1fl/CJC

In 2016, Zhao et al. [24] proposed the first Chinese jargon detection method. The authors used Word2Vec model to generate word embeddings and clustered jargons in underground market QQ groups by LDA (Latent Dirichlet Allocation) model. In another study on the detection of Chinese jargons by Yang et al. [19], the authors investigated the underground business promoted through blackhat SEO (Search Engine Optimization), and then built KDES (Keyword Detection and Expansion System) to find Chinese jargons in Baidu search engine. These two methods achieve the clustering of jargons, but do not take into account the differences between words when used as jargons and when used as normal semantics.

In subsequent studies [21, 16, 17, 7], researchers tried to introduce the cross-corpus information into the model to detect jargons. Yuan et al. [21] proposed a new technique called Cantreader, which is able to recognize and understand the English jargons in darknet marketplace. The authors modified the Word2Vec model to perform model training in dark corpora containing jargons and benign corpora in normal contexts simultaneously. Similarly, Takuro et al. [16] used Word2Vec for each of the two corpora and detected Japanese jargons in Twitter. Wang et al. [17] designed the CJI-Framework to identify Chinese jargons in Telegram by extracting seven novel features. Ke et al. [7] constructed a word-based pre-training language model called DC-BERT to generate high-quality contextual word embeddings for the corpus of darknet Chinese forums, and then perform cross-corpora jargon detection by computing the cosine similarity. Zhu et al. [26] took a different idea and used the Masked Language Model to self-supervisedly analyze the context to detect English euphemism and its hidden meanings. In addition, Zhu et al. [25] used a similar method which focuses on English euphemistic phrases detection.

In conclusion, existing studies have some common limitations: (1) Most of them obtain jargon candidates by setting a threshold. However, the threshold value is usually sensitive, and if the threshold value is not set properly, it can easily cause normal words to be incorrectly identified as jargons. (2) They did not consider the pronunciation and morphology characteristics of Chinese jargons.

3 Methodology

In this section, we give a detailed introduction of the proposed SICM model, which is shown in Fig. 1.

3.1 Character Feature Extraction Module

Phonetic Feature Extraction. Inspired by the 'Trans-pinyin' system proposed by Li et al. [9], we use a similar approach to extract the phonetic features of Chinese characters. The approach of 'Trans-pinyin' is to combine Chinese Pinyin with the IPA (International Phonetic Alphabet) system[3]. In the IPA sys-

[3] https://www.internationalphoneticassociation.org

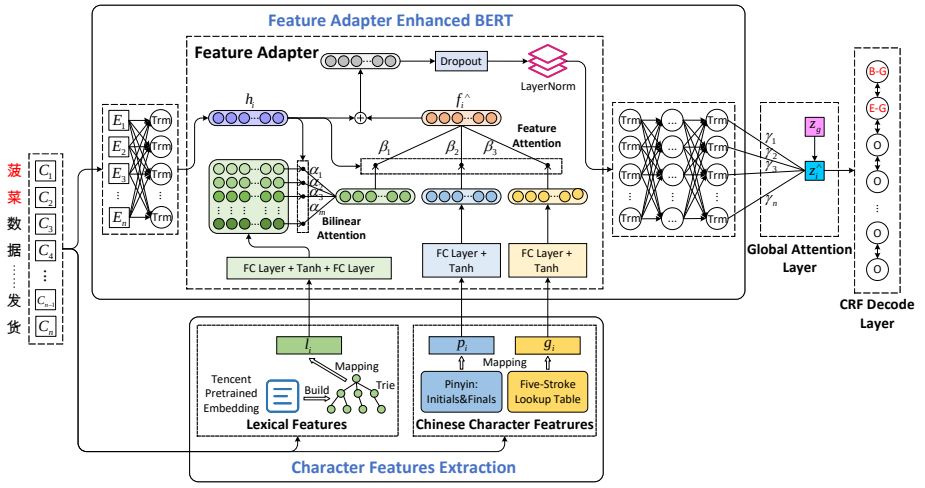

Fig. 1. Model Structure of SICM

tem, syllables with similar pronunciation can be well represented, which helps to describe the similarity of pronunciation between different Chinese characters.

Specifically, for each Chinese character in the input sequence, we first use the 'pypinyin' library[4] to get its Pinyin representation. Then, we divide the Pinyin representation into two parts: the initial and the final, and map them to the corresponding representations in the IPA system respectively, according to the conversion table[5]. Therefore, each part will be represented by a one-hot encoding. There is also a one-dimensional phonetic weight for each character. If two initials are similar in pronunciation, they will be represented as the same one-hot encoding with different phonetic weights. At last, we concatenate two one-hot encodings of the character and its phonetic weight to obtain its phonetic embedding representation.

Glyph Feature Extraction. We use the Five-stroke (or Wubi) encoding[6] of Chinese characters to generate glyph features, so that characters with similar morphology have similar vector representations. Five-stroke input method is a Chinese character input method, which encodes Chinese characters based on strokes and glyphs, and is a typical shape-based input method. The Five-stroke encoding creates mappings between different Chinese character roots and 25 English letters, with each Chinese character represented by no more than five English letters. For each character, we represent each bit of the Five-stroke encoding with a 25-dimensional one-hot encoding, and then concatenate them together to get the final glyph embedding.

[4] https://github.com/mozillazg/python-pinyin
[5] https://github.com/untunt/PhonoCollection/blob/master/Standard%20Chinese.md
[6] https://en.wikipedia.org/wiki/Wubi_method

Lexical Feature Extraction. Unlike English, Chinese words are not separated by spaces and lack distinct lexical boundaries, which makes Chinese sequence labeling a challenging task. There have been some works attempting to integrate lexical information into Chinese sequence labeling model to learn the boundary information related to Chinese words [23, 12, 10, 4, 11]. To allow our model to learn the boundary information related to Chinese jargons, we use a popular pre-trained embedding corpus from Tencent AI Lab[7], which is pre-trained on large-scale high-quality data using directional skip-gram model [14] and contains over 12 million words and phrases. More importantly, the corpus has advantages in coverage and freshness, including a large number of domain-specific words.

We use the method proposed by Liu et al. [11] to obtain lexical features. Specifically, a Trie is first constructed based on the pre-trained lexicon **D**. Then, for an input sentence sequence $X = \{x_1, x_2, x_3, ..., x_n\}$, all its character subsequences are traversed and matched with the Trie, such that each character x_i is given a word list $W_i = \left\{w_i^1, w_i^2, w_i^3, ..., w_i^m\right\}$. The m indicates the maximum number of words matched by a character. If the number of words is less than m, the remaining positions will be filled with '<PAD>'. Finally, based on W_i, we get the Character lexical feature $l_i = \left\{v_i^1, v_i^2, v_i^3, ..., v_i^m\right\}$, which is a set of word embeddings. The v_i^j is the word embedding of w_i^j and is obtained by the following equation:

$$v_i^j = \mathbf{E}\left(w_i^j\right) \tag{1}$$

where **E** denotes the lookup table of the pre-trained lexicon **D**. We treat the lexical feature l_i as a m-by-d_w matrix, where d_w denotes the dimension of the word embedding in **D**.

3.2 Feature Adapter Enhanced BERT Module

In this paper, we use the Chinese BERT pre-trained model [2] to extract text features. For the same word, BERT can generate different vectors according to the specific context, which effectively solves the problem of polysemy. For example, the Chinese jargon "马" (horse) can be either "欺骗行为中的受害者" (victim of fraud) in fraud or "木马病毒" (Trojan horse) in hacking.

The traditional feature fusion is usually performed after the output of BERT model. However, such approaches do not take full advantage of the powerful representation capability of BERT. To make our feature fusion more effective, we modify the structure of BERT. In 2019, Houlsby et al. [5] proposed the mechanism of BERT Adapter, and some works have applied it to the field of NLP (Natural Language Processing) [11, 1, 18, 13]. The BERT Adapter is added between Transformer layers of BERT and is designed to learn task-specific parameters for downstream tasks.

Inspired by studies [11] and [18], we design the Feature Adapter to integrate Chinese character features (including phonetic features and glyph features) and

[7] https://ai.tencent.com/ailab/nlp/en/embedding.html

lexical features into the lower layers of BERT directly. Specifically, we place the Feature Adapter between the first and second Transformer layers of BERT.

Feature Adapter accepts four inputs, which are text features, phonetic features, glyph features and lexical features of characters. For the i-th character x_i in the input sentence sequence $X = \{x_1, x_2, x_3, ..., x_n\}$, the input of Feature Adapter can be represented as (h_i, p_i, g_i, l_i), where $h_i \in \mathbb{R}^{d_h}$ denotes the character vector output by the previous Transformer layer; and $p_i \in \mathbb{R}^{d_p}$, $g_i \in \mathbb{R}^{d_g}$ denote the phonetic features and glyph features obtained by the Chinese character feature extractor, respectively; and $l_i \in \mathbb{R}^{m \times d_w}$ denotes the lexical features. To align the different feature representations, we apply the following transformations to the phonetic features and glyph features:

$$\hat{p_i} = \tanh\left(\mathbf{W}_p p_i + \mathbf{b}_p\right) \tag{2}$$

$$\hat{g_i} = \tanh\left(\mathbf{W}_g g_i + \mathbf{b}_g\right) \tag{3}$$

where $\mathbf{W}_p$ is a d_h-by-d_p matrix, $\mathbf{W}_g$ is a d_h-by-d_g matrix, $\mathbf{b}_p$ and $\mathbf{b}_g$ are bias. They are all learnable parameters.

The lexical feature $l_i = \left\{v_i^1, v_i^2, v_i^3, ..., v_i^m\right\}$ of each character is an m-by-d_w matrix. Here, we use the method of [11] to further process the lexical features. First, a nonlinear transformation is applied to l_i that maps the lexical features to the same dimensions as the text features:

$$l_i^{'} = \mathbf{W}_2\left(\tanh\left(\mathbf{W}_1 l_i^T + \mathbf{b}_1\right)\right) + \mathbf{b}_2 \tag{4}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ are learnable parameters, $\mathbf{W}_1$ is a d_h-by-d_w matrix, $\mathbf{W}_2$ is a d_h-by-d_h matrix, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias. In our approach, each character is matched to up to m words. However, not all of these words are useful. To be able to allow the model to select the most appropriate words and thus learn the correct boundary information, a bilinear attention mechanism is used here to calculate the weights of different words under the same character:

$$\alpha_i = \text{softmax}\left(h_i \mathbf{W}_{att} l_i^{'}\right) \tag{5}$$

$$\hat{l_i} = \sum_{j=1}^{m} \alpha_i^j v_i^j \tag{6}$$

where $\mathbf{W}_{att}$ is a d_h-by-d_h matrix, which is a learnable parameter and represents the weight matrix of the bilinear attention mechanism; α_i is the attention weight distribution representing the relevance between the current character x_i and the words it matches (i.e. words in W_i). Finally, we obtain the final lexical embedding $\hat{l_i}$ by attention-weighted summation.

As mentioned above, the transformed phonetic embedding $\hat{p_i}$, glyph embedding $\hat{g_i}$ and lexical embedding $\hat{l_i}$ are all d_h-dimensional vectors. Let $f_i = \left\{\hat{p_i}, \hat{g_i}, \hat{l_i}\right\}$, representing the 3-by-$h$ feature matrix of character x_i. Different jargons have different sensitivities to various features, so we design the following attention-based feature fusion strategy:

$$h_i^{'} = \tanh\left(\mathbf{W}_h h_i + \mathbf{b}_h\right) \tag{7}$$

$$\beta_t = \frac{\exp\left(h_i^{'}(f_i^t)^T\right)}{\sum_{k=1}^{3}\exp\left(h_i^{'}(f_i^k)^T\right)} \tag{8}$$

$$\hat{f}_i = \sum_{t=1}^{3}\beta_t f_i^t \tag{9}$$

where $\mathbf{W}_h$ is a d_h-by-d_h matrix and f_i^t denotes the t-th row in the feature matrix f_i. The β_t denotes the attention weights of different features. Then, we add the final character features $\hat{f}_i$ with h_i to obtain the character vector fused with the character features:

$$\hat{h_i} = h_i + \hat{f}_i \tag{10}$$

Finally, $\hat{h_i}$ is output as Feature Adapter after a dropout layer and layer normalization, and input to the next Transformer layer.

3.3 Global Attention Layer

In the Chinese jargon corpus, it is usually possible to recognize the category of crime present in the sentence by observing the contextual background of the whole sentence. Sentences with the same jargon categories usually tend to have similar contexts, while two sentences with different jargon types may have significant contextual distinction. Therefore, we believe that introducing global information of sentences for the model is beneficial to the identification and classification of jargons.

It should be noted that words in a sentence contribute differently to the global information. Referring to [22], we designed a global attention layer to introduce weighted global contextual information for each character in the sentence sequence. We denote the output of the last Transformer layer by $Z = \{z_1, z_2, z_3, ..., z_n\}$, and then calculate the attention score using the following:

$$z_g = \text{avg}\left\{z_1, z_2, z_3, ..., z_n\right\} \tag{11}$$

$$e_i = \mathbf{V}^T \tanh\left(\mathbf{W}_g z_g + \mathbf{W}_z z_i\right) \tag{12}$$

$$\gamma_i = \frac{\exp\left(e_i\right)}{\sum_{k=1}^{n}\exp\left(e_k\right)} \tag{13}$$

where z_g denotes the global representation of the whole sentence; $\mathbf{V} \in \mathbb{R}^{d_h}$, $\mathbf{W}_g \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_z \in \mathbb{R}^{d_h \times d_h}$ are all trainable parameters; γ_i denotes the attention score of the i-th character in the sentence sequence, reflecting the importance of that character in the global information of the whole sentence. Then, the sentence representation is generated by weighted summation:

$$s = \sum_{i=1}^{n}\gamma_i z_i \tag{14}$$

Finally, the global attention layer combines the generated sentence representation with the output of the last Transformer layer as the input to the following CRF (Conditional Random Field) decoding layer:

$$\hat{z}_i = z_i + s \tag{15}$$

4 Experiments

4.1 Dataset Construction

First, we adopt the raw data of the darknet which has been published by Ke et al. [7]. The data comes from multiple popular Chinese darknet websites[8], covering most types of cybercrimes. Then, In order to balance the amount of data in different crime categories, we supplement data from additional Chinese darknet forums[9]. All these raw data are not annotated.

We refer to the existing classification rules of Chinese jargons [7] and fine tune it, classifying jargons into seven categories: 'Drugs', 'Gambling', 'Pornography', 'Violence', 'Fraud', 'Hacking' and 'Others', which covers most types of cybercrime in darknet markets. It is worth noting that our dataset is quite different from those in previous works [24, 17, 7] on Chinese jargons, as we did not perform word segmentation on it but used the BIOES tagging scheme to label the corpus character by character, which can preserve the original information of texts. In the end, our dataset contains 33,668 sentences, including 19,675 sentences containing jargons and 13,993 normal sentences. A total of 45,079 jargons were labeled, including 1,796 unduplicated jargons. The specific information is shown in Table 1.

Table 1. Dataset Statistics

Category	#Sentences	#Jargons	#Unduplicated jargons	Example of jargons
Drug	1,936	3,920	225	叶子 (leaf), 猪肉 (pork)
Gambling	1,831	3,713	239	菠菜 (spinach), 搏彩 (fight color)
Pornography	4,132	10,560	362	果体 (fruit body), 狼友 (wolf friend)
Violence	1,456	3,469	203	气狗 (air dog), 秃鹰 (bald eagle)
Fraud	4,284	10,320	381	轨道料 (rail material), 裸条 (bare strip)
Hacking	4,138	8,147	341	蜜獾 (Mellivora capensis), 轰炸机 (bomber)
Others	2,998	4,860	47	梯子 (ladder), 洋葱 (onion)

4.2 Experiment Settings

In our work, all experiments are conducted on a workstation equipped with Intel(R) Core(TM) i9-10900 CPU and NVIDIA GeForce RTX 3070 GPU with

[8] https://github.com/KL4MVP/Chinese-Jargon-Detection/tree/master/dataset
[9] https://github.com/Y1fl/CJC

a memory size of 64 GB. All experimental models are built by the deep learning framework Pytorch v1.7.1.

We randomly divide our dataset into training, validation and test sets in the ratio of 6:2:2. In the partitioning process, we ensure that the ratio between different categories of jargons is approximately the same.

4.3 Sequence Labeling Baseline Model Comparison Experiment

To demonstrate the effectiveness of the proposed model SICM, we tested the performance of the model against seven baseline methods on our dataset CJC. These baselines are all sequence labeling models because no supervised jargon detection method has been proposed before. In particular, BERT-FeatureLSTM-CRF is a traditional feature fusion method, which concatenates the output of the last Transformer layer with the phonetic embedding, the glyph embedding and the weighted lexical embedding, and then uses BiLSTM as the fusion layer. This baseline can be used as a comparative experiment for feature fusion methods. In this experiment, we use *Precision*, *Recall* and *F1-score* as evaluation metrics. The results are shown in Table 2.

Table 2. Results for Sequence Labeling Baseline Model Comparison Experiment

Models	Precision	Recall	F1-score
BiLSTM-CRF [6]	87.91	87.42	87.66
IDCNN-CRF [15]	87.44	87.19	87.31
LatticeLSTM [23]	89.82	88.03	88.92
BERT-CRF	89.68	90.43	90.06
BERT-BiLSTM-CRF	90.05	90.87	90.46
LEBERT [11]	90.43	90.93	90.68
BERT-FeatureLSTM-CRF	90.28	90.99	90.64
SICM (Our Model)	**91.73**	**92.25**	**91.99**

Among all eight models, our model SICM shows the best results in each metric. The results of BiLSTM-CRF and IDCNN-CRF are closer, while LatticeLSTM performs better than them. This is because LatticeLSTM introduces lexical information into the model. The results of BERT-CRF and BERT-BiLSTM-CRF are significantly better than the first three models because they use the pre-trained model BERT. The results of LEBERT outperformed the first five baselines, probably because it uses a more advanced lexicon feature extraction method and is able to inject it into the lower layers of BERT, thus allowing the model to better capture the boundary information between words. In addition, SICM outperforms BERT-FeatureLSTM-CRF because the use of Feature Adapter for feature fusion takes full advantage of the powerful representation capabilities of BERT.

Experimental results show that our model SICM outperforms various advanced sequence labeling models for jargon detection, and our feature fusion approach is more effective than the traditional method.

4.4 Unsupervised Methods Comparison Experiment

In this experiment, we apply four existing unsupervised jargon detection methods to our dataset CJC and compare their results with our model SICM.

Each unsupervised method generates a jargon list. Considering that these are unsupervised methods, we rank the jargon lists generated by each method and then use Precision at k ($P@k$) as the evaluation metric, which is often used in the field of information retrieval to assess the relevance of search results to a query [20]. Specifically, we define the indicator function as follows:

$$I_{w_0}(w) = \begin{cases} 1, & w \text{ is a jargon;} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The criterion for judging jargons here are those labeled in our dataset CJC. We selected multiple k values (k = 10, 20, 30, 40, 50, 60, 80, 100). In addition, for methods designed to detect English jargons [21, 26], we use an open source Chinese word segmentation tool[10] to preprocess the dataset.

Table 3. Results for Unsupervised Methods Comparison Experiment

Methods	Language	$P@10$	$P@20$	$P@30$	$P@40$	$P@50$	$P@60$	$P@80$	$P@100$
CantReader [21]	English	0.00	0.00	0.07	0.05	0.04	0.07	0.10	0.10
CJI-Framework [17]	Chinese	**0.70**	**0.40**	0.30	0.30	0.28	0.30	**0.33**	**0.33**
DC-BERT [7]	Chinese	0.50	0.35	**0.40**	**0.38**	**0.36**	**0.33**	0.29	0.27
MLM [26]	English	0.10	0.20	0.17	0.18	0.16	0.15	0.13	0.12

The results are shown in Table 3. It can be observed that CJI-Frameworks and DC-BERT, the two detection methods for Chinese jargons, can achieve relatively good results. However, even for the $P@10$ metric, CJI-Frameworks, the best performer among the unsupervised methods, can only reach 0.70, i.e., only 7 of the top 10 most likely words are jargons. In contrast, the *Precision*, *Recall* and *F1-score* of SICM can all reach above 0.90, which achieves better results. CantReader performs poorly on our dataset, whose conclusion is similar to the research work [26]. We infer that this is because the method requires an additional benign corpus and it is difficult for us to guarantee that the selected Chinese corpus[11] is appropriate. MLM also does not work well, the method uses a native BERT model and cannot predict multiple tokens at the same time, so it is not well suited for Chinese jargon detection.

Experiments show that our method is significantly better than existing state-of-the-art unsupervised methods and can better detect jargons in the dark web.

4.5 Ablation Experiment

Our model combines Chinese character features (including phonetic features and glyph features), lexical features, and global information for sentences. We

[10] https://github.com/fxsjy/jieba
[11] https://github.com/brightmart/nlp_chinese_corpus

explore the contribution of each module by ablation experiment. Specifically, model without Chinese character features (SICM\CF), model without lexicon features (SICM\LF), and model without global attention layer (SICM\GA) are evaluated in turn and compared with SICM. The evaluation metrics for this experiment are also *Precision*, *Recall* and *F1-score*.

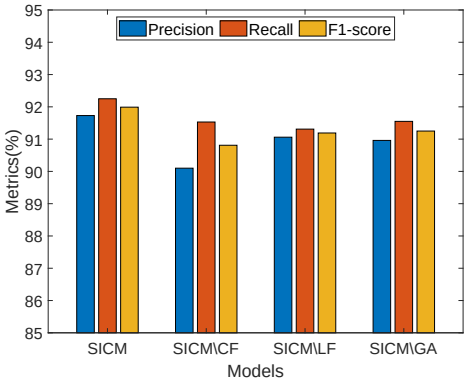

Fig. 2. Results for Ablation Experiment.

The results are shown in Fig. 2. It can be seen that each of the ablation models causes a decrease in the results, which proves that our proposed Chinese character features, lexical features, and global attention layer are all beneficial to the detection of Chinese jargons. Among them, (SICM\CF) performs the worst, which indicates that the effectiveness of Chinese character features is the greatest. This further improves that introducing phonetic features and glyph features can help the model to better understand Chinese jargons.

5 Conclusion

In this paper, we propose the first supervised-based Chinese jargon identification and classification model, to the best of our knowledge. At the beginning, we construct and publish the first labeled dataset of Chinese jargons. Then, we transform jargon detection into a sequence labeling problem and propose our brand-new model. Specifically, in order to better represent the specific characteristics of Chinese jargon and to facilitate more effective feature fusion, we innovatively propose a Chinese jargon identification and classification model based on Feature Adapter enhanced BERT, which uses attention mechanism to integrate phonetic, glyph and lexical features into the lower layers of BERT. Furthermore, we employ a global attention layer to let our model grasp the global information of the sentences. Experiments show that our model outperforms existing state-of-the-art jargon detection methods. This study provides a brand-new research idea for the Chinese jargon detection in the darknet marketplace.

References

1. Bapna, A., Firat, O.: Simple, scalable adaptation for neural machine translation. In: Proceedings of the 24th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 1538–1548 (2019)
2. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z.: Pre-training with whole word masking for chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing **29**, 3504–3514 (2021)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186 (2019)
4. Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., Si, L.: A neural multi-digraph model for chinese ner with gazetteers. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1462–1467 (2019)
5. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: Proceedings of the 36th International Conference on Machine Learning. pp. 2790–2799 (2019)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. Ke, L., Chen, X., Wang, H.: An unsupervised detection framework for chinese jargons in the darknet. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining. pp. 458–466 (2022)
8. Kovalchuk, O., Masonkova, M., Banakh, S.: The dark web worldwide 2020: Anonymous vs safety. In: Proceedings of the 11th International Conference on Advanced Computer Information Technologies. pp. 526–530 (2021)
9. Li, J., Meng, K.: Mfe-ner: Multi-feature fusion embedding for chinese named entity recognition. arXiv preprint arXiv:2109.07877 (2021)
10. Li, X., Yan, H., Qiu, X., Huang, X.J.: Flat: Chinese ner using flat-lattice transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6836–6842 (2020)
11. Liu, W., Fu, X., Zhang, Y., Xiao, W.: Lexicon enhanced chinese sequence labeling using bert adapter. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. pp. 5847–5858 (2021)
12. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.J.: Simplify the usage of lexicon in chinese ner. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5951–5960 (2020)
13. Pfeiffer, J., Vulić, I., Gurevych, I., Ruder, S.: Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In: Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing. pp. 7654–7673 (2020)
14. Song, Y., Shi, S., Li, J., Zhang, H.: Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In: Proceedings of the 16th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 175–180 (2018)
15. Strubell, E., Verga, P., Belanger, D., McCallum, A.: Fast and accurate entity recognition with iterated dilated convolutions. In: Proceedings of the 22nd Conference on Empirical Methods in Natural Language Processing. pp. 2670–2680 (2017)

16. Takuro, H., Yuichi, S., Tahara, Y., Ohsuga, A.: Codewords detection in microblogs focusing on differences in word use between two corpora. In: Proceedings of the 3rd International Conference on Computing, Electronics & Communications Engineering. pp. 103–108 (2020)
17. Wang, H., Hou, Y., Wang, H.: A novel framework of identifying chinese jargons for telegram underground markets. In: Proceedings of the 30th International Conference on Computer Communications and Networks. pp. 1–9 (2021)
18. Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X.J., Ji, J., Cao, G., Jiang, D., Zhou, M.: K-adapter: Infusing knowledge into pre-trained models with adapters. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. pp. 1405–1418 (2021)
19. Yang, H., Ma, X., Du, K., Li, Z., Duan, H., Su, X., Liu, G., Geng, Z., Wu, J.: How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy. In: Proceedings of the 38th IEEE Symposium on Security and Privacy. pp. 751–769 (2017)
20. Yang, P., Fang, H., Lin, J.: Anserini: Enabling the use of lucene for information retrieval research. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1253–1256 (2017)
21. Yuan, K., Lu, H., Liao, X., Wang, X.: Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces. In: Proceedings of the 27th USENIX Security Symposium. pp. 1027–1041 (2018)
22. Yuan, Y., Zhou, X., Pan, S., Zhu, Q., Song, Z., Guo, L.: A relation-specific attention network for joint entity and relation extraction. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. pp. 4054–4060 (2020)
23. Zhang, Y., Yang, J.: Chinese ner using lattice lstm. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. pp. 1554–1564 (2018)
24. Zhao, K., Zhang, Y., Xing, C., Li, W., Chen, H.: Chinese underground market jargon analysis based on unsupervised learning. In: Proceedings of the 14th IEEE Conference on Intelligence and Security Informatics. pp. 97–102 (2016)
25. Zhu, W., Bhat, S.: Euphemistic phrase detection by masked language model. In: Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing. pp. 163–168 (2021)
26. Zhu, W., Gong, H., Bansal, R., Weinberg, Z., Christin, N., Fanti, G., Bhat, S.: Self-supervised euphemism detection and identification for content moderation. In: Proceedings of the 42nd IEEE Symposium on Security and Privacy. pp. 229–246 (2021)