A Novel Framework of Identifying Chinese Jargons for Telegram Underground Markets

Hailin Wang, Yiwei Hou, Haizhou Wang*
School of Cyber Science and Engineering, Sichuan University
Chengdu (610207), P.R.China
$\{hiramwhl; houyiwei\}$ $@stu.scu.edu.cn; whzh.nc@scu.edu.cn$

2021 International Conference on Computer Communications and Networks (ICCCN) | 978-1-6654-1278-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCN52240.2021.9522221

***Abstract*—As one of the most popular instant messaging (IM) software, Telegram has reached 500 million monthly active users (MAU) up to January 2021. Nevertheless, the characteristics of safety and openness have also made it a popular platform for transactions in underground markets. Moreover, cybercriminals usually use jargons instead of sensitive terms when they communicate in Telegram groups. Nevertheless, jargons identification relies on time-consuming and lagging manual work currently. To solve this problem, this paper proposes a Chinese Jargons Identification Framework (CJI-Framework) to identify jargons automatically. Firstly, we collect chat history from targeted Telegram groups to build a corpus called TUMCC, which is the first Chinese corpus in jargons identification field. Secondly, we extract seven brand-new features which can be classified into three categories: Vectors-based Features (VF), Lexical analysis-based Features (LF), and Dictionary analysis-based Features (DF), to distinguish between Chinese jargons and commonly-used words. Furthermore, we use a word vectors projection method and a transfer learning method to improve the quality of word vectors generated from the corpora. In our experiments, the CJI-Framework reaches a remarkable jargons identification performance with an F1-score of 89.66%. This work provides a method of identifying Chinese jargons for Telegram underground markets effectively and will be helpful for cybercrime investigation. It can also be helpful to jargons identification related to other similar communication platforms and languages.**

***Index Terms*—Security and privacy, Chinese jargons, Telegram, Word vectors, Transfer learning**

I. Introduction

With the development of the Internet, cybercrimes are becoming more and more rampant around the world in recent years [1]. Cybercriminals make illegal transactions in underground markets, such as spreading obscene videos, distributing hacking tools, reselling leaked personal privacy data, and selling guns and drugs in districts under regulation, which are illegal and have disturbed the social order in the real world. As one of the most popular communication methods, Telegram is an open-source, cross-platform instant messaging (IM) software. It is known for its safety and openness [2]. 1) Safety. It can provide three forms of end-to-end encrypted communication: one-to-one, group-based, and channel-based. Besides, the loose service policies of Telegram allow users to send illegal messages without being monitored or censored. 2) Openness. Individuals can utilize the official API [1] to develop robots, which helps take full advantage of the software and make it more customizable. These two outstanding characteristics have been attracting an enormous number of users. In January 2021, Telegram reported reaching 500 million MAU[2], which means that it has become one of the most popular global IM software. Nevertheless, Telegram is convenient not only for common users but also for cybercriminals. They can use this software to spread illegal information and make transactions in underground markets.

A. Jargons in the Underground Markets

To understand how cybercrimes are carried out, and what strategies cybercriminals formulate to protect themselves, cybersecurity researchers can analyze the chat history related to their transactions and advertisements. Nevertheless, chat history in underground markets is usually written carefully and contains lots of jargons[3]. Jargons are also known as unusual usage of words, which have no semantic relevance with their common usage. For instance, in Chinese underground markets, "飞行员 (pilot)" refers to "吸毒者 (drug addicts)". As can be seen from this example, jargons are innocent-looking, and their utilization can provide strong protection for illegal transactions in underground markets. By using jargons, cybercriminals can construct a barrier for text understanding and conceal important information. In Telegram underground markets, jargons usually indicate product names, team roles, or trading methods. Therefore, the automatic identification of jargons is highly significant in understanding Telegram underground markets effectively and monitoring various cybercrimes.

Many cybersecurity researchers have carried out studies on jargons identification [3]–[7]. However, most of the existing research focuses on English jargons, while there is no research based on Chinese jargons. Besides, these studies are mainly aimed at dark web forums and social network (e.g. Twitter), while studies based on IM software such as Telegram is few. The approaches proposed by previous work cannot be used for the jargons identification for Telegram directly, due to differences in language, platform, contexts, and so on. Thus we propose a new method to automatically identify Chinese jargons for Telegram underground markets.

*Corresponding author
ORCID(s): 0000-0003-1197-5906 (H. Wang)

[1] Telegram API. https://core.telegram.org/.

[2] Telegram Announcement. https://t.me/durov/147, 2021.

[3] Jihadists and Vault 7: What it Means for the Rest of Us. https://www.flashpoint-intel.com/blog/terrorism/jihadists-vault-7/, 2017.

978-1-6654-1278-0/21/$31.00 ©2021 IEEE

B. Challenges

At present, research on the identification of Chinese jargons for Telegram underground markets mainly face the following three challenges:

1) The first challenge is that there is no high-quality corpus for Chinese jargons identification experiments currently. It is difficult to develop a dedicated crawler to collect chat history since the challenge of finding out active Telegram groups related to transactions in underground markets. In addition, sample labeling requires a lot of experience and time, which is error-prone and tends to limit the scale of the corpus.

2) The second challenge is the lack of effective features for Chinese jargons identification. There have been studies introducing some features of English jargons [4], [7], but there are no features of Chinese jargons that have been proposed in previous research work. English-based features cannot be applied to the identification of Chinese jargons directly because of the language difference. Thus it is necessary to extract more suitable features to achieve better identification performance.

3) The third challenge is the lack of a valid Chinese jargons identification framework. English-based frameworks introduced in previous research cannot be applied in the Chinese context directly. Moreover, the characteristics of Telegram are different from dark web forums or the social network. As a result, a complete framework of identifying Chinese jargons for Telegram underground markets is still inadequate.

C. Contributions and Organization

As for the above challenges, we design a novel framework, namely Chinese Jargons Identification Framework (CJI-Framework), to identify Chinese jargons. It is composed of four modules: *Data Collection and Preprocessing*, *Lexical Analysis and Word Counting*, *Vector Generating and Optimization*, and *Feature Selection and jargons identification*. Firstly, the *Data Collection and Preprocessing* module collects chat history of Telegram underground markets and performs text-cleaning tasks. Secondly, the *Lexical Analysis and Word Counting* module records the occurrence count of a word, the context count of a word, and the word class (noun, preposition, adjective, etc.). Then, in the *Vector Generating and Optimization* module, we generate word vectors of every word in corpora with a transfer learning method and a vector projection method. Finally, in the *Feature Selection and jargons identification* module, we compute values of seven features of each word, then run a statistical outlier detection [8] to decide whether a word is a jargon.

The main contributions of our study are summarized as below:

- We construct a Telegram Underground Market Chinese Corpus, namely TUMCC, which has been released on GitHub[4]. To the best of our knowledge, the TUMCC is the first Chinese corpus containing the chat history of Telegram groups related to transactions in underground markets. Meanwhile, we propose an approach to generate high-quality word vectors based on a relatively small scale of chat history, and experimental results indicate that it improves the performance of jargons identification.
- We extract seven brand-new features of Chinese jargons. These features can be classified into three categories: Vectors-based Features (VF), Lexical analysis-based Features (LF), and Dictionary analysis-based Features (DF). Experimental results show that these seven features proposed by us can help identify Chinese jargons accurately.
- We propose a novel jargons identification framework, namely CJI-Framework (Chinese Jargons Identification Framework), and have released the source code of the framework[5]. It can effectively capture and make use of the context of jargons to improve the performance of jargons identification. Experimental results show that, in the Telegram underground markets, the CJI-Framework performs well in identifying Chinese jargons, with an accuracy rate of 87.50%, a precision rate of 92.86% , a recall rate of 86.67%, and an F1-score of 89.66%.

The structure of this paper is organized as follows. In Section II, we introduce related works and achievements in relevant research fields. The design and implementation of the CJI-Framework are elaborated in Section III. Furthermore, we evaluate this framework through experiments in Section IV. Eventually, Section V summarizes our work.

II. Related Work

In this section, we introduce studies related to jargons identification for Telegram. There isn't much research in this exact field, while most of them fall into studies on underground market and economy, or covert communication approaches such as Telegram. The related work of underground markets, communication approaches, and jargons identification is introduced separately below.

A. Underground Market and Economy

Many criminals make use of the Internet as an important way to publish and exchange information, thus how to monitor cybercrimes has become an important research field. Moshchuk et al. [9] collect information such as files and links to investigate cybercrimes. Pastranaet et al. [10] noticed that cybercriminals always make use of underground forums to communicate with each other, thus they designed a bot to crawl their information in underground sites and build up a dataset to analyze their actions. Broadhurst et al. [11] summarized categories of illegal transactions in online underground markets. They conducted a comprehensive analysis of the methods to commit cybercrimes. As a result, all of these works proposed ways to understand the dark web and underground market.

[4] Telegram Underground Market Chinese Corpus (TUMCC). https://github.com/HiramWHL/TUMCC.

[5] Chinese Jargons Identification Framework (CJI-Framework) code. https://github.com/m1-llie/CJI-Framework.

B. Covert Communication Approaches

To understand the underground market better, we must know more about their communication platforms. Interactive platforms in the deep web and dark web, for example, IM software and online forums, have been studied for years. These interactive channels have facilitated the transmission of sensitive information in underground markets.

Sutikno et al. [12] compared the communication security, synchronization, backup, and other functions of IM software such as WhatsApp, Viber, and Telegram, and eventually concluded that unconditional security is the most notable characteristic of Telegram. Nobari et al. [2] analyzed the structural and topical aspects of messages published in Telegram, then extracted the mention graph and page rank of their data. Shehabat et al. [13] studied the role of encrypted social media in cybercrimes and found that Twitter and Facebook were strictly regulated, while Telegram was the final choice. From all of the research above, we can conclude that Telegram is the most ideal choice compared with other communication platforms.

C. Content Analysis and Jargons Identification

Even we know where cybercrimes are happening and how to collect their chat history, it is also necessary to understand their communication. To protect themselves, chat history in underground markets often contains a lot of jargons. As for the identification of jargons in underground markets, Yang et al. [4] proposed a method of using search engines to analyze illegal products and services. However, since jargons are often commonly-used words, there is always severe competition when they are put into search engines. To address this problem, Hada et al. [5] proposed a Japanese-based jargons identification approach. They calculated the word similarity between a certain word and prepared already-known jargons. If they are close enough, this word can be determined as a jargon. Aoki et al. [6] identified jargons by Vectors-based Features, and proposed a standard calculation method to distinguish between commonly-used words and jargons. Yuan et al. [7] proposed the SCM model for English jargons identification. They modified the Word2Vec to train two corpora simultaneously. The SCM model can generate a pair of comparable word vectors at a time. Comparing the two sets of word vectors, the semantics of each word can be compared, and then jargons can be identified.

Still, the research above has some shortages. Most of them depend on the quality of word vectors heavily. However, high-quality word vectors need a large-scale dataset to train from, which is hard to satisfy. Besides, some of their works even need a well-maintained jargon list, which is difficult to apply widely. Furthermore, none of their work can be adapted to identify Chinese jargons for Telegram because of the lack of features.

III. Framework Design and Implementation

To identify Chinese jargons more effectively, we propose the CJI-Framework, as shown in Figure 1. There are four modules in this framework: *Data Collection and Preprocessing*, *Lexical Analysis and Word Counting*, *Vector Generating and Optimization*, and *Feature Selection and jargons identification*.The four modules of the CJI-Framework are introduced in detail below.

A. Data Collection and Preprocessing

TABLE I
Overview of OCC and ICC.

Corpus (Statistical objects)	Sources	Numbers
OCC (Sentences)	Weibo	4,435,959
	Tieba	13,580,419
	Douban	500,000
OCC (Chinese Characters)	Weibo	151,795,728
	Tieba	523,268,741
	Douban	157,121,585
ICC (Sentences)	Wikipedia	369,870
ICC (Chinese Characters)	Wikipedia	162,988,659

The CJI-Framework uses three corpora: self-built TUMCC (Telegram Underground Market Chinese Corpus), Oral Chinese Corpus (OCC), and Interpretative Chinese Corpus (ICC). The OCC consists of public Weibo, Tieba, and Douban corpus[6]; the ICC is from the public Chinese Wikipedia dataset[7]. Thus, the OCC represents oral Chinese materials while the ICC represents formal usage of Chinese. Details are shown in Table I.

To the best of our knowledge, there is no high-quality corpus for Chinese jargons identification currently, so we build TUMCC by ourselves. Firstly, we select 12 Telegram groups that are active in transactions in underground markets (selling guns, drugs, etc.). Then we develop a dedicated crawler to collect the chat history from these groups. The crawler worked from August to September 2020, collecting chat history from January 2017 to August 2020. Through this process, jargons being used recently in Chinese underground markets can be gathered. A total of 28,749 sentences, including 804,971 characters, from 19,821 Telegram users were collected.

Then, we carry out three steps to clean and formalize the raw texts. First of all, information such as username and online time was removed through text cleaning. Secondly, jargon labeling was carried out manually by two researchers, and when objections occurred, a third-party arbitration would step in. Finally, punctuations and stop words were removed. Text segmentation was done by Jieba[8], i.e. a famous Chinese word segmentation tool. As a result, the TUMCC is built, which contains 3,863 sentences (a total of 100,000 characters) from 3,139 Telegram users.

[6] Chinese Word Vectors. https://github.com/Embedding/Chinese-Word-Vector.

[7] Chinese Wikipedia Dataset. https://dumps.wikimedia.org/zhwiki/latest/ .

[8] Jieba word segmentation. https://github.com/fxsjy/jieba.

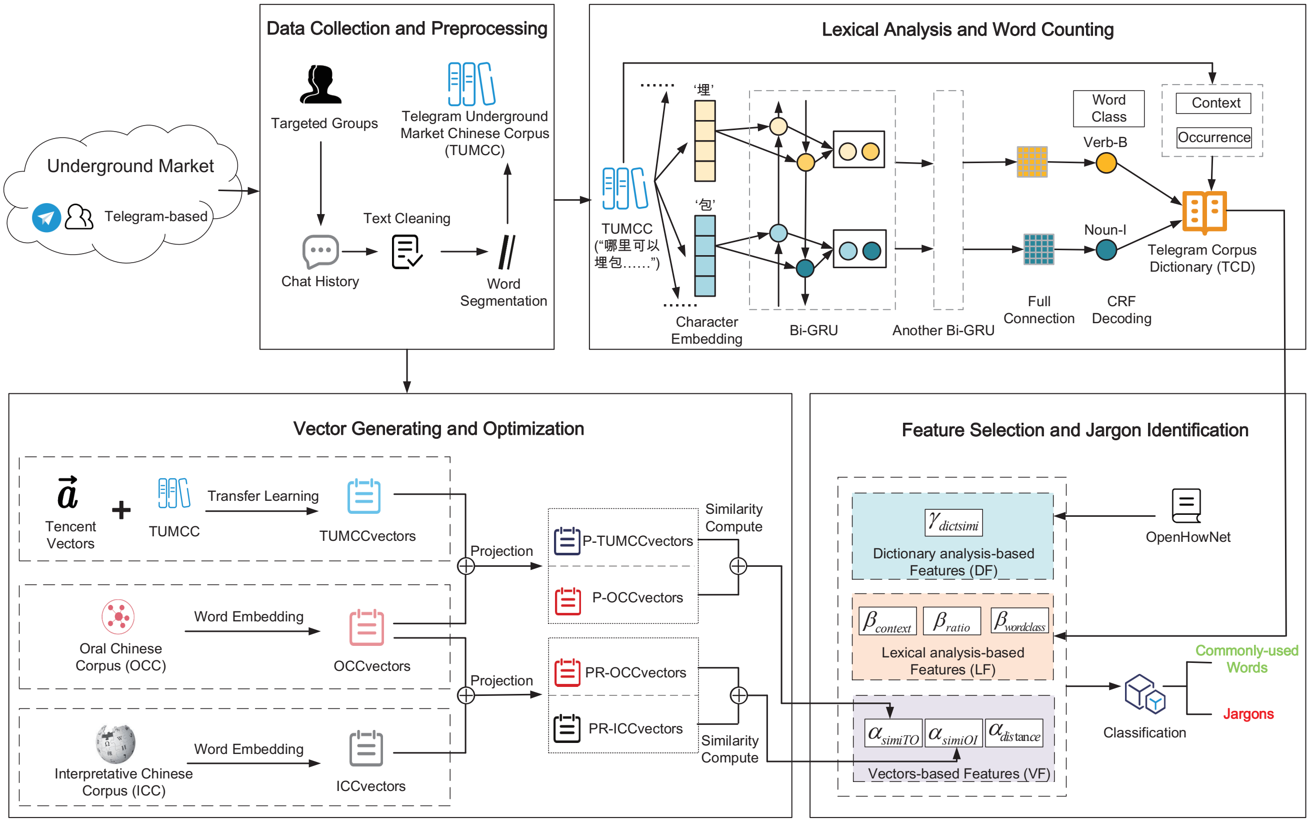

Fig. 1. The proposed Chinese jargons identification framework, CJI-Framework.

B. Lexical Analysis and Word Counting

The TUMCC has already been built in Section III-A above. Next, we further build the TCD (Telegram Corpus Dictionary) which contains words in TUMCC and their properties: their occurrence count, their context count, and their word class (noun, preposition, adjective, etc.).

Since the generation of word vectors relies on the analysis of different contexts [14], a word must appear many times in the corpus. To filter out these words, we regard how many times a word has appeared as their occurrence count. Nevertheless, if a word always appears in the same contexts, we also cannot generate its word vectors precisely. Thus we use a “window mechanism” to evaluate the diversity of a word’s contexts as the context count.

Besides, to obtain the word class of all words, we perform lexical analysis in TUMCC. Using the Bi-GRU-CRF model [15], we can analyze the composition of sentences and the word class of each word, then get the LF (Lexical analysis-based Features) results.

As shown in Figure 1, firstly, through the embedding step, we generate a vectorized representation of each sentence in the TUMCC. For one sentence, namely a sequence of characters $\{c_1, c_2, ..., c_T\}$, each character c_i will be mapped to a vector $e(c_i)$. Vectors are the input of the following processes. Secondly, after the embedding step, we build the Bi-GRU Layer. A Bi-directional GRU (Bi-GRU) is formed by a reversed GRU combined with a forward GRU, and their results will be concatenated as output. The deep network formed from multiple Bi-GRUs can represent some functions and model varying-length dependencies, thus can achieve a better result [15]; in particular, we stack two Bi-GRUs to form the Bi-GRU Layer. Furthermore, the Full-connection Layer converts the output of the Bi-GRU Layer into an L-dimensional vector (L is the number of all possibilities of the word class). Finally, the CRF Layer decodes the final sequence.

In particular, in our CJI-Framework, the CRF Layer connected behind Bi-GRUs has the function of constraint decoding. The $h = \{h_1, h_2, ..., h_T\}$ is a sequence representation generated by the second Bi-GRU, $y = \{y_1, y_2, ..., y_T\}$ is a label sequence, and $P(y|h)$ is a conditional probability learned by the CRF Layer. $P(y|h; t, s)$ is defined by the probabilistic model of linear-chain CRF:

$$P(y \mid h; t, s) = \frac{\prod_{i=1}^{T} \psi_i \left(y_{i-1}, y_i, h\right)}{\sum_{y' \in \gamma(h)} \prod_{i=1}^{T} \psi_i \left(y'_{i-1}, y'_i, h\right)} \quad (1)$$

where

$$\psi \left(y_{i-1}, y_i, h\right) = \exp\left(\sum_{i=1}^{T} t\left(y_{i-1}, y_i, h\right) + s\left(y_i, h\right)\right) \quad (2)$$

and the $\gamma(h)$ represents all possible tag sequences. The t refers to the transition probabilities, and the s denotes the output of the linear function.

As we have mentioned above, the CRF Layer will find a value y that maximizes the family of conditional probability $P(y \mid h; t, s)$ during the process of decoding. As a result, the word class of each word in input sentences can be finally determined. This feature, namely $\beta_{\text{wordclass}}$, will be further discussed in Section III-D.

C. Vector Generating and Optimization

Since jargons are unusual usage of words, their meanings are quite different between the TUMCC and general public Chinese corpora. The meaning of a word can be inferred by its context [16]. Therefore, we can identify a jargon by finding out the cross-corpus difference of a word's context. Nevertheless, a word is also used differently between formal situations and oral situations. Therefore, we classify general public Chinese corpora into two categories when collecting data, namely OCC and ICC, which have demonstrated in Table I.

We employ a widely-used language model, namely word vectors, to compute the contextual differences of a word in different corpora. The generation of word vectors can make full use of a word's context [17]. Therefore, if the similarity of two word vectors is high, their corresponding words will have similar contexts. To capture this feature, we convert TUMCC, OCC, and ICC into corresponding sets of word vectors: TUMCCvectors, OCCvectors, and ICCvectors. When comparing cross-corpus semantics, we need to compare a word between TUMCC and OCC by calculating the cosine similarity, namely $Sim_{TUMCC-OCC}$. If the value is lower than the average similarity value, it indicates that the usage of the word is inconsistent between underground markets and oral situations, and it may be a jargon. Similarly, we can compute the $Sim_{OCC-ICC}$ between the OCC and the ICC.

1) Vectors Generation Based on the Transfer Learning Method: The effective training of word vectors must be based on a large-scale corpus. Nevertheless, due to the concealment of transactions in underground markets, it is often tough for cybersecurity researchers to find enough materials to build a large-scale and high-quality corpus to train word vectors [18]. Besides, labeling jargons is hardly carried out on a large corpus because of time and energy limitations. To reduce the reliance on the scale of labeled TUMCC, we use the transfer learning method to build the TUMCCvectors based on a selected pre-trained vectors.

We construct a character-level VCDM (Variational Contextual Definition Modeler) [19] to do transfer learning. The VCDM consists of three parts: *Encoders*, *Neural Definition Inferer*, and *Variational Definition Modeler*. The higher level LSTM can capture context-based features such as topics, while the lower level LSTM can learn grammatical-level information such as the word class. Through model extraction and refitting, the sentence structure is scanned and the internal state is updated to generate TUMCCvectors based on the current sentence. The generated TUMCCvectors can represent the new usage of words in the TUMCC.

To sum up, we complete the construction of TUMCCvectors, OCCvectors, and ICCvectors in the *Vector Generating and Optimization* module. In the subsequent process, these vectors will be used to compute the VF.

2) Semantic comparison based on unsupervised projection: To compute the $Sim_{TUMCC-OCC}$, we have to compare vectors that are trained from different corpora. The training process includes steps that introduce strong randomnesses, such as weight parameter initialization when the GloVe is applied to generate word vectors. Even if the same corpus and parameters are used to generate word vectors sets twice, the outcome is still quite distinct. Therefore, if the word vector generation model is used directly to train two independent vectors sets, the outcome is not comparable. Experimental results in Section IV-A proves our hypothesis. As all corpora share concepts that are grounded in the real world [20], vector spaces of separately trained vectors are usually similar, and words with similar meanings are always close in the space [21]. To solve the problem that word vectors generated independently cannot be compared directly, we construct a transfer matrix to project these vectors into one shared space, where they can be comparable. We can use transfer matrix W to map the source vector x into the destination vector z:

$$z = W^* x \tag{3}$$

The gradient descent algorithm is applied to construct this transfer matrix. Suppose there are a set of word pairs and vector representation associated with them, we can define them as $\{x_i, z_i\}_{i=1}^n$. The $x_i \in \mathbb{R}^{d_1}$ is the word vector representation of the word i. The $z_i \in \mathbb{R}^{d_2}$ is another word vector representation of word i. The loss function can be defined as follows, using the distance between the corresponding words in two corpora:

$$\min_W \sum_{i=1}^{n} \|W^* x_i - z_i\|^2 \tag{4}$$

We still need a seed library containing word-pairs between the two corpora to compute the value of the loss function above. Therefore, we adopt a reinforcement learning method [22] used in the machine translation field to avoid building the library manually. Specifically, we build the seed library through four steps. Firstly, we carry out word vectors preprocessing step to orthogonalize word vectors. Secondly, we build a preliminary seed library based on multi-dimensional comparison of each word in the vectors set. Furthermore, we apply reinforcement learning to optimize its quality repeatedly. Finally, symmetric re-weighting [23] is applied to further improve the quality of the seed library. Through the steps above, we can build an effective seed library that can be used in the construction of the transfer matrix.

After the previous steps, we improve the comparability of two separately trained word vector sets greatly. We classify TUMCCvectors, OCCvectors and ICCvectors described in Section III-C1 into two comparing groups: TUMCCvectors

TABLE II
OVERVIEW OF OUR JARGONS IDENTIFICATION FEATURES.

Category	Feature Description	Feature Symbol
Vectors-based Features (VF)	Cosine similarity of P-TUMCCvectors and P-OCCvectors	α_{simiTO}
	Cosine similarity of PR-OCCvectors and PR-ICCvectors	α_{simiOI}
	The absolute value of the difference between α_{simiTO} and α_{simiOI}	α_{distance}
Lexical analysis-based Features (LF)	The context count	β_{context}
	The ratio of context count and occurrence count	β_{ratio}
	Word class (noun, preposition, adjective, etc.)	$\beta_{\text{wordclass}}$
Dictionary analysis-based Features (DF)	The dictionary analysis results based on OpenHowNet [24]	γ_{dictsimi}

and OCCvectors, along with OCCvectors and ICCvectors. After projection, there are two pairs of comparable word vectors being generated: P-TUMCCvectors and P-OCCvectors, along with PR-OCCvectors and PR-ICCvectors. These two pairs will be used in the calculation of seven features in Section III-D.

D. Feature Selection and Jargon Identification

The final module of the CJI-Framework is to determine whether a word ought to be classified as a jargon. Seven features have been extracted based on the following: two comparable word vectors pairs mentioned above, the TCD, and dictionary analysis based on OpenHowNet [24]. These features are shown in Table II.

Details of feature calculation are as follows.

1) Vectors-based Features, VF: Jargons is a replacement of commonly-used words in underground markets, using for product names, team roles, or trading methods. Such substitutions often change the context of commonly-used words greatly, while word vectors generation tools, such as GloVe [9], can use exactly the context information to generate vectors. According to the theory proposed by Yuan et al. [7], the vector similarity of a jargon between P-TUMCCvectors and P-OCCvectors will be remarkably lower than that of a commonly-used word, which can be a feature to characterize changes in the context. Cosine similarity is seen as the standard of similarity comparison. Its calculation is as below:

$$\cos\theta = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}|\cdot|\vec{y}|} = \frac{x_1*y_1+x_2*y_2+\ldots+x_n*y_n}{\sqrt{x_1^2+x_2^2+\ldots+x_n^2}*\sqrt{y_1^2+y_2^2+\ldots+y_n^2}} \quad (5)$$

Nevertheless, we cannot determine jargons through a low similarity between P-TUMCCvectors and P-OCCvectors directly, because the usage of a word may be also quite different between formal situations and oral situations. For example, "粉红" means "pink" in formal situations, while in oral situations it mostly means "patriotism". If OCC contains a certain percentage of formal situations, such as Wikipedia and official news, false positives will exist (the similarity of the certain word such as the word "粉红" mentioned above will be low between TUMCC and OCC, thus it will be misjudged as a jargon). Thus we introduce PR-OCCvectors to compare with PR-ICCvectors. *Assumption 1) the meaning of a word between OCC and ICC is very similar, that is, $Sim_{OCC-ICC}$ (the feature α_{simiOI}) is bigger than the average value; Assumption 2) the meaning of the word between TUMCC and OCC is very different, that is, $Sim_{TUMCC-OCC}$ (the feature α_{simiTO}) is smaller than the average value; Assumption 3) the word have different meaning in two comparing groups, that is $\|\text{sim}_{TUMCC-OCC} - \text{Sim}_{OCC-ICC}\|$ (the feature $\alpha_{\ distance}$) is larger than average.* If one word meets the three assumptions above, then it will be defined as a jargon by VF. To conclude, the usage of a jargon should be similar between formal situations and oral situations, while it is quite distinct between oral situations and Telegram underground markets, thus we can use α_{simiOI}, α_{simiTO}, and $\alpha_{\text{ distance}}$ to compose VF which can be a group of features of jargons identification.

2) Lexical analysis-based Features, LF: If the context diversity of a word is below a given threshold, current word vector generation algorithms will not be able to generate its vectors effectively [14]. We introduce a "window mechanism" in the calculation to evaluate the contexts diversity of a word. The word itself and k words around it are taken into consideration at the same time. That is, to be counted as a new context, k characters around the word (i.e. its context) cannot be the same. We summarize the counting result as the feature β_{context}. Referring to the research of Yan et al. [7], we set the k value to 5. Furthermore, we also regard the ratio of the context count and the occurrence count as the feature β_{ratio}.

We have found that in Chinese underground markets, jargons are usually used by cybercriminals to represent their product names, team roles, or trading methods, etc. In this case, jargons are mostly used as nouns or verbs in sentences. For example, "大麻" (marijuana) is called "叶子" (leaf), and one certain trading method is called "埋包" (burying). In the CJI-Framework, lexical analysis is applied to identify the word class and filter out ones that are not related to jargons (such as prepositions). We will filter out these words and only keep nouns and verbs in our jargon list.

3) Dictionary analysis-based Features, DF: Word sense disambiguation technology [25] can find out words with abnormal meanings which are not contained in the dictionary, which is another perspective of jargons identification besides

[9] Glove. https://nlp.stanford.edu/projects/glove/.

word vectors. Therefore, we carry word sense disambiguation step based on OpenHowNet dictionary [24]. We combine the OpenHowNet with a similarity comparison method based on sememes [26]. Thus we can further confirm whether the word has a significantly different meaning in TUMCC.

The calculation method of γ_{dictsimi} in DF we proposed is as follows. Suppose we need to calculate γ_{dictsimi} value of word A. Firstly, we need to calculate the cosine similarity of the word A and all other words B_i in TUMCCvectors one by one, and then sort them in descending order, taking top k words $\{B_i \mid 1 \leq i \leq k\}$ as synonyms of the word A. Next, the OpenHowNet dictionary is applied to check the similarities between the word A and $\{B_i \mid 1 \leq i \leq k\}$. After that, we calculate the average similarity to obtain the value γ_{dictsimi}. If a word is used commonly, the average similarity mentioned above should be a value higher than the average, and vice versa. Therefore, this feature is also helpful to identify jargons from commonly-used words. We have undertaken an experiment to find out an ideal value of k, and finally find that the relatively most appropriate one is $k = 20$, so 20 is selected for subsequent experiments.

IV. Experiment

Our experiments were undertaken on a server with Intel (R) Xeon (R) Gold 6130 CPU, 128G memory, and Tesla V100 GPU, containing 32G video memory. All experiments have been repeated ten times to get the final average result.

Four metrics are used for effect evaluation, including Accuracy, Precision, Recall, and F1-score. The confusion matrix is employed to introduce these metrics. True Positive (TP) is the number of jargons that are accurately identified, False Positive (FP) is the number of commonly-used words that are mistakenly regarded as jargons, False Negative (FN) is the number of jargons that are mistakenly regarded as commonly-used words, and True Negative (TN) is the number of commonly-used ones that are correctly classified. Positive and negative samples are categorized by two researchers independently. When objections occurred, a third-party arbitration would step in to ensure the accuracy of classification.

Then Accuracy, Precision, Recall, and F1-score can be computed as follows:

$$Accuracy = \frac{|TP + TN|}{|TP + FP + FN + TN|} \tag{6}$$

$$Precision = \frac{|TP|}{|TP + FP|} \tag{7}$$

$$Recall = \frac{|TP|}{|TP + FN|} \tag{8}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

Through the following experiments, we have evaluated the implementation and performance of the CJI-Framework.

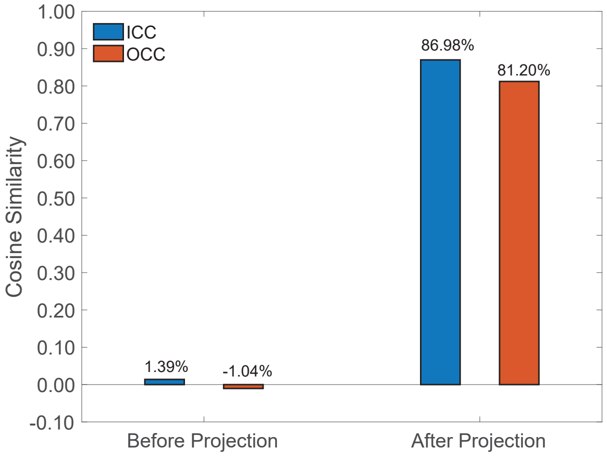

Fig. 2. Comparison results of corresponding vectors cosine similarity before and after projection.

A. Evaluation of the Necessity of Word Vectors Projection

In this experiment, we use two corpora, OCC and ICC, to generate their word vectors twice. To verify the validity of projection, we compute comparison results of corresponding vectors cosine similarity before and after projection (elaborated in Section III-C2). The experimental results in Figure 2 show that, for each corpus, the average cosine similarity of corresponding vectors before projection is low, with -0.0104 for OCC and 0.0139 for ICC. Because of the randomness of initial parameters such as weight parameter initialization in Glove, if vectors are generated twice for each corpus, vectors of the same word should be quite different, thus their similarity value will be low. In another group, the similarity of projected vectors is 0.8120 and 0.8698 correspondingly. The value becomes significantly higher, which indicates that the projection step can eliminate the influence of training randomness. This shows that the projection step is necessary and effective to control variables, which makes the vectors that generated independently be comparable.

B. Evaluation of Feature Effectiveness

We extract seven brand-new features and they are of three categories: Vectors-based Features (VF), Lexical analysis-based Features (LF), and Dictionary analysis-based Features (DF). To verify the validity of the proposed features, we conduct a feature ablation experiment with the TUMCC. That is, each time we remove a category of features, and then jargons identification work is done to explore the contribution of remaining subsets. The subsets can be elaborated by the following set-difference function:

$$F \backslash F' = \{x \mid x \in F \wedge x \notin F'\} \tag{10}$$

where the F is all features of three categories, the F' is a subset of the F with a particular category, and the x is all user data of a feature.

TABLE III
JARGON IDENTIFICATION RESULTS OF DIFFERENT FEATURE SETS.

Features Sets	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1-score** (%)
F	**87.50**	**92.86**	**86.67**	**89.66**
F/DF	87.30	76.47	68.42	72.22
F/LF	65.08	45.45	78.95	57.69
F/VF	55.56	39.53	89.47	54.83

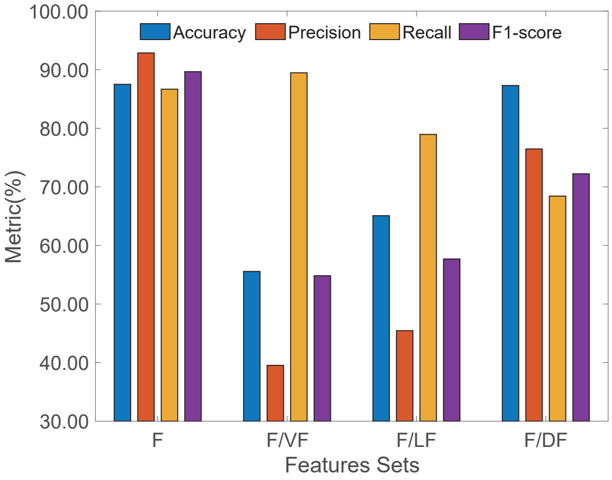

Fig. 3. Jargon identification results of different feature sets.

The experimental results are shown in Table III and Figure 3. The performance of each approach with F, F/DF, F/LF, and F/VF are compared. It demonstrates that when the seven features of the three categories are all taken into consideration at the same time, we get the best result. Excluding any category of features will lead to a decrease in performance. In addition, all approaches perform worst using the feature set of F/VF, which indicates that the validity of Vectors-based Features (VF) is the greatest. To conclude, the validity of feature categories can be sorted from highest to lowest as the VF, the LF, and the DF.

TABLE IV
RESULTS OF DIFFERENT JARGONS IDENTIFICATION APPROACHES.

Approach	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F1-score** (%)
SCM [7]	14.12	12.64	95.45	22.32
SCM+LF+DF	80.79	32.35	50.00	39.28
SCM+LF	79.10	35.29	81.82	49.31
SCM+DF	79.66	36.00	81.82	50.00
CJI-Framework	**87.50**	**92.86**	**86.67**	**89.66**

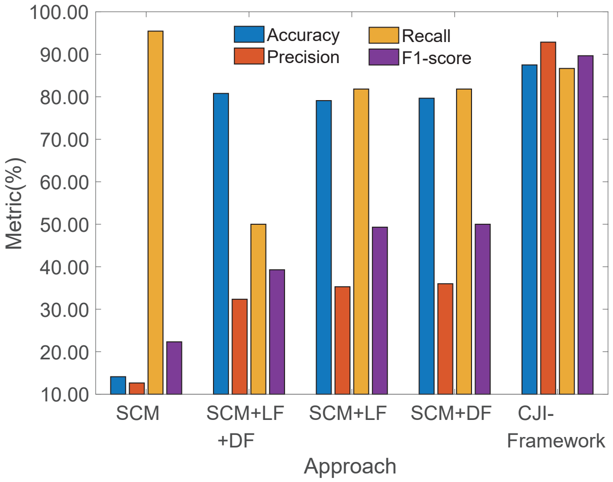

Fig. 4. Results of different jargons identification approaches.

C. Evaluation of the Proposed Framework

The CJI-Framework combines word vector projection and transfer learning parts, as well as the seven features of Chinese jargons. It has achieved ideal results for Telegram groups related to underground markets. Before that, an effective model in the area of jargons identification was the SCM model [7] aimed at English jargons. By improving Word2Vec, the SCM makes it possible to train two comparable word vectors sets at the same time, avoiding the influence of randomness. Nevertheless, it cannot utilize the various features of jargons. To evaluate the CJI-Framework, we use the SCM model and the CJI-Framework separately to carry a Chinese jargons identification task based on the TUMCC. To control the number of variables, we also add LF and DF features to the SCM. The experimental results are shown in Table IV and Figure 4.

It can be seen that the F1-score of the SCM is only 22.32% in the Chinese environment. After combining the $\beta_{\text{wordclass}}$, the performance of the SCM has been effectively improved, with an F1-score of 49.31%. Further, with DF being added, the F1-score reaches 50.00%. When LF and DF are added at the same time, the SCM can reach an accuracy rate of 80.97%, but due to low performance in the metrics of precision rate and recall rate, the final metric, F1-score, is still low (the exact value is 39.28%). To compare, the CJI-Framework reaches an F1-score of 89.66%. This shows that: 1) The DF and the LF are indeed effective and necessary. 2) The SCM is designed for English, so the process and implementation cannot be transferred to Chinese directly. To sum up, the CJI-Framework can achieve better results in Chinese jargons identification.

V. CONCLUSION

This paper proposes a brand-new CJI-Framework to identify Chinese jargons for Telegram underground markets. Specifically, to evaluate our framework, we construct the TUMCC, the first Chinese corpus containing the chat history of Telegram groups related to transactions in underground markets. Moreover, we extract seven features of three categories aimed at Chinese, including the VF, the LF, and the DF, to distinguish between jargons and commonly-used words. Furthermore, applying a transfer learning approach for word vectors generation and a reinforcement learning method for vectors projection, the CJI-Framework reaches better performance for identifying jargons. The experimental results show that this framework is an efficient method for Chinese jargons identification.

While the jargons identification technology is developing, cybercriminals are also utilizing new methods to prevent jargons from being identified when they communicate. Therefore, making our framework adaptable to the evolution of communication methods in Telegram underground markets will be done in our further work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61802270, 61802271, 81602935, and 81773548. In addition, this work is also partially supported by The Applied Basic Research Programs of Science and Technology Department of Sichuan Province (No.2019YJ0110) and The Key Research and Development Program of Science and Technology Department of Sichuan Province (No. 2020YFS0575). The authors thank anonymous reviewers for their helpful comments to improve the paper.

References

[1] R. van Wegberg, F. Miedema, U. Akyazi, A. Noroozian, B. Klievink, and M. van Eeten, “Go see a specialist? predicting cybercrime sales on online anonymous markets from vendor and product characteristics,” in *Proceedings of the 29th International World Wide Web Conference (WWW’20)*, Taipei, 2020, pp. 816–826.

[2] A. Dargahi Nobari, N. Reshadatmand, and M. Neshati, “Analysis of telegram, an instant messaging service,” in *Proceedings of the 26th ACM on Conference on Information and Knowledge Management (CIKM’17)*, Singapore, 2017, pp. 2035–2038.

[3] K. Zhao, Y. Zhang, C. Xing, W. Li, and H. Chen, “Chinese underground market jargon analysis based on unsupervised learning,” in *Proceedings of the 14th IEEE Conference on Intelligence and Security Informatics (ISI’16)*, Tucson, USA, 2016, pp. 97–102.

[4] H. Yang, X. Ma, K. Du, Z. Li, H. Duan, X. Su, G. Liu, Z. Geng, and J. Wu, “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 (Oakland S&P’17)*, San Jose, USA, 2017, pp. 751–769.

[5] T. HADA, Y. SEI, Y. TAHARA, and A. OHSUGA, “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 (ICCECE’20)*, Southend, UK, 2020, pp. 103–108.

[6] T. Aoki, R. Sasano, H. Takamura, and M. Okumura, “Distinguishing japanese non-standard usages from standard ones,” in *Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing (EMNLP’17)*, Copenhagen, Denmark, 2017, pp. 2323–2328.

[7] K. Yuan, H. Lu, X. Liao, and X. Wang, “Reading thieves’ cant: Automatically identifying and understanding dark jargons from cybercrime marketplaces,” in *Proceedings of the 27th USENIX Security Symposium (USENIX Security’18)*, Baltimore, USA, 2018, pp. 1027–1041.

[8] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, “Outlier detection for temporal data: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[9] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy, “A crawler-based study of spyware in the web,” in *Proceedings of the 13th Network and Distributed System Security Symposium (NDSS’06)*, San Diego, USA, 2006, pp. 2–2.

[10] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, “Crimebb: Enabling cybercrime research on underground forums at scale,” in *Proceedings of the 27th International World Wide Web Conference (WWW’18)*, Lyon, France, 2018, pp. 1845–1854.

[11] R. Broadhurst, D. Lord, D. Maxim, H. Woodford-Smith, C. Johnston, H. W. Chung, S. Carroll, H. Trivedi, and B. Sabol, “Malware trends on ‘darknet’ crypto-markets: Research review,” *Available at SSRN: http://ssrn.com/abstract=3226758*, 2018.

[12] T. Sutikno, L. Handayani, D. Stiawan, M. A. Riyadi, and I. M. I. Subroto, “Whatsapp, viber and telegram: Which is the best for instant messaging?” *International Journal of Electrical & Computer Engineering*, vol. 6, no. 3, pp. 2088–8708, 2016.

[13] A. Shehabat, T. Mitew, and Y. Alzoubi, “Encrypted jihad: Investigating the role of telegram app in lone wolf attacks in the west,” *Journal of Strategic Security*, vol. 10, no. 3, pp. 27–53, 2017.

[14] R. Sasano and A. Korhonen, “Investigating word-class distributions in word vector spaces,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL’20)*, online, 2020, pp. 3657–3666.

[15] Z. Jiao, S. Sun, and K. Sun, “Chinese lexical analysis with deep bi-gru-crf network,” *arXiv preprint arXiv:1807.01882*, 2018.

[16] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 2177–2185, 2014.

[17] A. Lauscher, G. Glavas, S. P. Ponzetto, and I. Vulic, “A general framework for implicit and explicit debiasing of distributional word vector spaces,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI’20)*, New York, USA, 2020, pp. 8131–8138.

[18] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, “Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse,” in *Proceedings of the 22nd USENIX Security Symposium (USENIX Security’13)*, Washinton D. C. , USA, 2013, pp. 195–210.

[19] M. Reid, E. Marrese-Taylor, and Y. Matsuo, “Vcdm: Leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling,” in *Proceedings of the 17th Conference on Empirical Methods in Natural Language Processing (EMNLP’20)*, Punta Cana, Dominican, 2020, pp. 6331–6344.

[20] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *arXiv preprint arXiv:1309.4168*, 2013.

[21] T. Liu, L. Ungar, and J. Sedoc, “Unsupervised post-processing of word vectors via conceptor negation,” in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI’19)*, Hawaii, USA, 2019, pp. 6778–6785.

[22] M. Artetxe, G. Labaka, and E. Agirre, “A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL’18)*, Melbourne, Australia, 2018, pp. 789–798.

[23] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. E. A. Seddik, and M. Tamaazousti, “Learning more universal representations for transfer-learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2212–2224, 2020.

[24] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, “Openhownet: An open sememe-based lexical knowledge base,” *arXiv preprint arXiv:1901.09957*, 2019.

[25] A. Raganato, J. Camacho-Collados, and R. Navigli, “Word sense disambiguation: A unified evaluation framework and empirical comparison,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL’17)*, Alencia,Spain, 2017, pp. 99–110.

[26] Y. Niu, R. Xie, Z. Liu, and M. Sun, “Improved word representation learning with sememes,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL’17)*, vol. 1, Vancouver, Canada, 2017, pp. 2049–2058.