Information Processing and Management 59 (2022) 103033

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Identification of Chinese dark jargons in Telegram underground markets using context-oriented and linguistic features

Yiwei Hou, Hailin Wang, Haizhou Wang *

School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Keywords:
Jargons identification
Information security
Feature engineering
Word embedding
Transfer learning
Vectors projection

ABSTRACT

When cybercriminals communicate with their customers in underground markets, they tend to use secure and customizable instant messaging (IM) software, i.e. Telegram. It is a popular IM software with over 700 million monthly active users (MAU) up to June 2022. In recent years, more and more dark jargons (i.e. an innocent-looking replacement of sensitive terms) appear frequently on Telegram. Therefore, jargons identification is one of the most significant research perspectives to track online underground markets and cybercrimes. This paper proposes a novel Chinese Jargons Identification Framework (CJI-Framework) to identify dark jargons. Firstly, we collect chat history from Telegram groups that are related to the underground market and construct the corpus ***TUMCC*** (Telegram Underground Market Chinese Corpus), which is the first Chinese corpus in jargons identification research 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 identify Chinese dark jargons from commonly-used words. Based on these features, we then run a statistical outlier detection to decide whether a word is a jargon. Furthermore, we employ a word vector projection method and a transfer learning method to improve the effect of the framework. Experimental results show that CJI-Framework achieves a remarkable performance with an F1-score of 89.66%. After adaptation for English, it performs better than state-of-the-art English jargons identification method as well. Our built corpus and code have been publicly released to facilitate the reproduction and extension of our work.

1. Introduction

1.1. Cybercrimes, illegal transactions, and online underground markets

With the development of the Internet, cybercrimes are becoming more and more rampant around the world in recent years (Wegberg et al., 2020). There are several kinds of cybercrimes, such as online fraud (Dou et al., 2020; Pastrana, Hutchings, Thomas, & Tapiador, 2019), malicious hacking activities (Huang & Ban, 2020; Pastrana, Hutchings, Caines, & Buttery, 2018; Samtani, Zhu, & Chen, 2020; Zhao et al., 2021), drug trafficking (Zhang, Fan, Song et al., 2019; Zhang et al., 2020), and illegal transactions (Fan et al., 2020; Kumar et al., 2020; Portnoff et al., 2017; Zhang, Fan, Ye, Zhao & Shi, 2019). To gain economic benefits, cybercriminals make illegal transactions in online underground markets hosted through various channels, such as the Dark Web, online social networks (OSNs), and instant messaging (IM) software. Contents of transactions are diversified, for example,

* Corresponding author.
E-mail address: whzh.nc@scu.edu.cn (H. Wang).

https://doi.org/10.1016/j.ipm.2022.103033
Received 1 June 2021; Received in revised form 4 May 2022; Accepted 13 July 2022
Available online 29 July 2022

spreading obscene videos, distributing hacking tools, reselling leaked personal privacy data, and selling guns and drugs in districts where they are regulated or even banned. They are illegal and have disturbed the social order in the real world.

1.2. Communication platforms for cybercriminals and their customers

There are several channels for cybercriminals and their customers to keep in touch with each other. According to Lusthaus (2019), communication channels that are used by these cybercriminals can be divided into four levels: (1) the top layer, which are the most open forums and marketplaces, e.g. Dark Web; (2) the middle layer of more closely vetted forums; (3) the bottom layer of even smaller and more closed groupings; (4) the molten core, which is centered on the offline organization of cybercrimes. Specifically, public social networks and messaging software also have a dark side of being used as a platform for cybercrimes (Alassad, Spann, & Agarwal, 2021). As one of the most popular communication channels on the top layer, Telegram is an open-source, cross-platform IM software. It is known for its safety and openness (Nobari, Reshadatmand, & Neshati, 2017). (1) Safety. It can provide three forms of end-to-end encrypted communication: one-to-one, group-based, and channel-based. In addition, the loose service policies of Telegram allow users to send illegal messages without being monitored or censored. (2) Openness. Individuals can use 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 June 2022, Telegram reported reaching 700 million monthly active users (MAU),[2] which means that it has become one of the most popular IM software globally. 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 Telegram underground markets.

1.3. Dark jargons in Telegram underground markets

To understand how cybercrimes are carried out and what strategies cybercriminals use to avoid supervision, cybersecurity researchers can analyze the chat history related to their online transactions and advertisements. Nevertheless, chat history in underground markets is usually written carefully and contains lots of “dark jargons”. Dark jargons (we call them “jargons” for short in this paper) are also known as unusual usage of standard words, which have no semantic relevance to their common usage. For instance, in Chinese underground markets, “飞行员 (pilot)” implies “吸毒者 (drug addicts)”, “天窗 (dormer)” implies “静脉注射毒品 (the vein that drugs are injected into)”, “里面 (inside)” implies “监狱 (jail)”, and “童子军 (boy scout)” implies a certain kind of drug. As can be seen from these examples, jargons are innocent-looking, and their adoption can provide strong protection against illegal transactions in online underground markets, leaving no trace for outsiders to comprehend exactly what they are talking about. In Telegram underground markets, jargons usually indicate product names, team roles or trading methods. Thus, cybercriminals can construct a barrier to text understanding and conceal important information. Only experienced buyers can understand these messages, which are unintelligible to outsiders. Therefore, jargons identification is highly valuable in investigating Telegram underground markets and monitoring potential cybercrimes.

However, most of the existing research only focuses on English jargons. Besides, they are mainly aimed at Dark Web forums and social networks (e.g. Twitter). The approaches proposed by previous research cannot be used directly for the jargons identification for Telegram due to differences in language, platform, contexts, and so on. Thus, we propose a novel framework to automatically identify jargons in Chinese underground markets for Telegram.

1.4. Contributions and organizations

As for the above research objectives, we design a novel framework, namely Chinese Jargons Identification Framework (CJI-Framework), to identify jargons in Chinese online underground markets. It is composed of four modules: *Corpus Preparation Module*, *Lexical analysis-based Features (LF) Extraction Module*, *Vectors-based Features (VF) Extraction Module*, and *Dictionary analysis-based Features (DF) Extraction Module*. Firstly, the *Corpus Preparation Module* collects the chat history of Telegram groups related to underground markets, and performs text-cleaning tasks such as word segmentation along with removing punctuation and stop words, and finally, the Telegram Underground Market Chinese Corpus (***TUMCC***) is constructed. We also construct the Oral Chinese Corpus (***OCC***) and the Interpretative Chinese Corpus (***ICC***). Secondly, the *Lexical analysis-based Features (LF) Extraction Module* records the *occurrence count* of a word, the *context count* of a word, and the *word class* (noun, preposition, adjective, etc.), which are contained in the Telegram Corpus Dictionary (TCD) to compute the features $\beta_{context}$, β_{ratio}, and $\beta_{wordclass}$. Then, in the *Vectors-based Features (VF) Extraction Module*, we generate word vectors of every word in corpora by employing a transfer learning method and a vector projection method, and then compute vectors-related features α_{simiTO}, α_{simiOI}, and $\alpha_{distance}$. Finally, in the *Dictionary analysis-based Features (DF) Extraction Module*, we compute the feature $\gamma_{dictsimi}$ from queries of OpenHowNet,[3] which is a sememe-based lexical knowledge base. Based on these seven features, we run a statistical outlier detection (Gupta, Gao, Aggarwal, & Han, 2013) to decide whether a word is a jargon.

The main contributions of our study are summarized below:

[1] Telegram API. https://core.telegram.org/, accessed on 2022-07-25.

[2] Telegram Announcement. https://telegram.org/blog/700-million-and-premium, accessed on 2022-07-25.

[3] OpenHowNet. https://github.com/thunlp/OpenHowNet, accessed on 2022-07-25.

- We construct Telegram Underground Market Chinese Corpus, namely ***TUMCC***, which has been released on GitHub.[4] It contains 3,863 sentences with 100,000 Chinese characters, from 3,139 individuals in 12 chat groups. 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 show 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 the features proposed by us can help identify Chinese jargons more accurately. Among them, the VF contributes the most, followed by the LF and the DF.
- We propose a novel jargons identification framework, namely the CJI-Framework (Chinese Jargons Identification Framework), and have released the source code.[5] The CJI-Framework generates word vectors from ***TUMCC*** and applies a transfer learning method to improve the quality of vectors. It can effectively capture and make use of the context of jargons to improve the performance of jargons identification. Experimental results show that, in Telegram underground markets, 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%. Moreover, CJI-Framework also improves the F1-score by 4.79% than state-of-the-art method in English jargons identification.

The structure of this paper is organized as follows. In Section 2, we state the research objectives of our work. Section 3 presents the literature review on relevant research fields. The design and implementation of the CJI-Framework are elaborated in Section 4. We introduce the datasets and evaluate the framework through experiments in Section 5. Furthermore, in Section 6, we highlight the theoretical and practical implications of our research. Eventually, Section 7 summarizes the research and discusses future work.

We note that a shorter conference version of this paper appeared in Wang, Hou, and Wang (2021). Our initial work did not address the problem of English adaptation. This paper addresses this issue, improves the framework design and implementation, and provides additional experimental results from new perspectives.

2. Research objectives

At present, studies on the identification of Chinese dark jargons in Telegram underground markets still face challenges. The main objective of this paper is to introduce a complete framework to integrate features to identify Chinese dark jargons in online underground markets. The followings are three problem statements.

2.1. How to collect materials and build a high-quality corpus for Chinese jargons identification work

It is difficult to develop a dedicated crawler to collect chat history due to the challenge of finding 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. There is currently no high-quality corpus in this research field to the best of our knowledge. Therefore, a well-prepared corpus for Chinese jargons identification should be constructed.

2.2. Which features are effective for Chinese jargons identification and how to extract them

There have been studies introducing some features of English jargons (Yang et al., 2017; Yuan, Lu, Liao, & Wang, 2018), 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 for Chinese to achieve better identification performance.

2.3. How to design a valid Chinese jargons identification framework and evaluate the design

English-based frameworks introduced in previous research cannot be applied in the Chinese context directly. Moreover, the characteristics of IM software such as Telegram are different from those of dark web forums or social networks (Hada, Sei, Tahara, & Ohsuga, 2020). As a result, a complete framework of identifying Chinese jargons for Telegram underground markets is still extremely scarce. Hence, designing and implementing a novel jargons identification framework for Chinese in Telegram is of great value.

3. Literature review

In this section, we review the research on tracking online underground markets and cybercrimes, and highlight works on jargons identification. Many cybercriminals make use of the Internet as an important way to exchange information and make illegal transactions, thus how to track online underground markets and cybercrimes has become an important research field. Strategies for cybercrimes reduction and prevention are essential to keep the social order and reduce the cost of crime to society. Studies of how to track online underground markets and cybercrimes can be classified into three categories: analysis of member relationships, analysis of communication channels, and analysis of communication content.

[4] Telegram Underground Market Chinese Corpus (TUMCC). https://github.com/yiyepianzhounc/TUMCC, accessed on 2022-07-25.

[5] Chinese Jargons Identification Framework (CJI-Framework) code. https://github.com/yiyepianzhounc/CJI-Framework, accessed on 2022-07-25.

3.1. Analysis of memberships in online underground markets

Tayebi, Ester, Glässer, and Brantingham (2014) used a social network analysis (SNA) method to conduct cybercrime analysis, proposing a framework for co-offense prediction. Morgia, Mei, Raponi, and Stefa (2018) developed a way to uncover the geographical distribution of groups of dark web market visitors into time zones. They used the time of all posts in the dark web forums to build profiles of the visiting crowds. Then they uncovered the geographical origin of the dark web crowd by matching the crowd profile to users from known regions on regular web platforms. Fan et al. (2020) introduced an attributed heterogeneous information network (AHIN) to simulate complex relations among entities in underground markets. Based on the constructed AHIN, they further constructed a heterogeneous GNN model (mHGNN) to propagate and aggregate the information, and finally conducted illicit traded product identification. Kumar et al. (2020) built a multi-view unsupervised framework (eDarkFind) which can take advantage of domain-specific knowledge, to detect Sybil accounts. Their model combined features such as substance features, stylometric features, location, and domain-specific contextual features to give a vendor-level multi-view embedding.

3.2. Analysis of communication channels in online underground markets

To understand online underground markets better, one of the research perspectives is to know more about their communication channels. Interactive platforms in the deep web and dark web, for example, IM software and online forums, have been studied for years. These communication channels have facilitated the exchange of sensitive information and illegal transactions in online underground markets. Sutikno, Handayani, Stiawan, Riyadi, and Subroto (2016) compared 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. (2017) analyzed the structural and topical aspects of messages published in Telegram, then extracted the mention graph and page rank of their data. Hoseini et al. (2020) studied several platforms' ecosystems, including Telegram, WhatsApp, and Discord, to understand how public groups on these platforms differ in characteristics and usage.

3.3. Analysis of communication content in online underground markets

Pastrana, Thomas, Hutchings, and Clayton (2018) 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. Lee et al. (2019) proposed a framework called MFScope to collect cryptocurrency addresses in online underground markets. They could use the data to classify usages of cryptocurrencys to identify trades of illicit goods, and trace cryptocurrency money flows. Their work can be helpful in revealing black money operations in online underground markets. Haasio, Harviainen, and Savolainen (2020) examined contextual features of 9300 dis-normative messages from a Finnish dark website, and contributed to the understanding of drug seeking and buying behavior.

Specifically, there is a sub-direction to analyze the communication content at the word level. There are several works focusing on the identification of dark jargons (abnormal usage of normal words) (Aoki, Sasano, Takamura, & Okumura, 2017; Hada et al., 2020; Yang et al., 2017; Yuan et al., 2018), neologisms (newly created terms) (Dasgupta et al., 2020; Li, Cheng, Huang, Chen, & Niu, 2021; Zhao, Zhang, Xing, Li, & Chen, 2016), blended words (multiple words joined together) (Farrell, Araque, Fernandez, & Alani, 2020; Maddela, Xu, & Preoţiuc-Pietro, 2019), and keywords extraction (Nasar, Jaffry, & Malik, 2019). As for the identification of jargons in online underground markets, Yang et al. (2017) proposed a method of using search engines to analyze illegal products and services. Nevertheless, since jargons are often innocent-looking, there is always severe competition when they are put into search engines. Aoki et al. (2017) identified jargons by vectors-based features, and proposed a standard calculation method to distinguish commonly-used words and jargons. Yuan et al. (2018) proposed the SCM model for English jargons identification. They modified Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) 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, which helps identify jargons. Hada et al. (2020) 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.

Still, the current research on jargons identification has some shortages. Most of them depend on the quality of word vectors heavily, while high-quality word vectors need a large-scale dataset to train from, which is hard to satisfy. Moreover, some of their works need a well-maintained jargon list, which is difficult to apply widely. Furthermore, none of the existing work has focused on achieving a good performance to identify Chinese jargons. Thus, we propose a novel framework of identifying Chinese jargons, with transfer learning being applied to generate word vectors for dark jargons. Our method does not rely on a ground-truth jargon list.

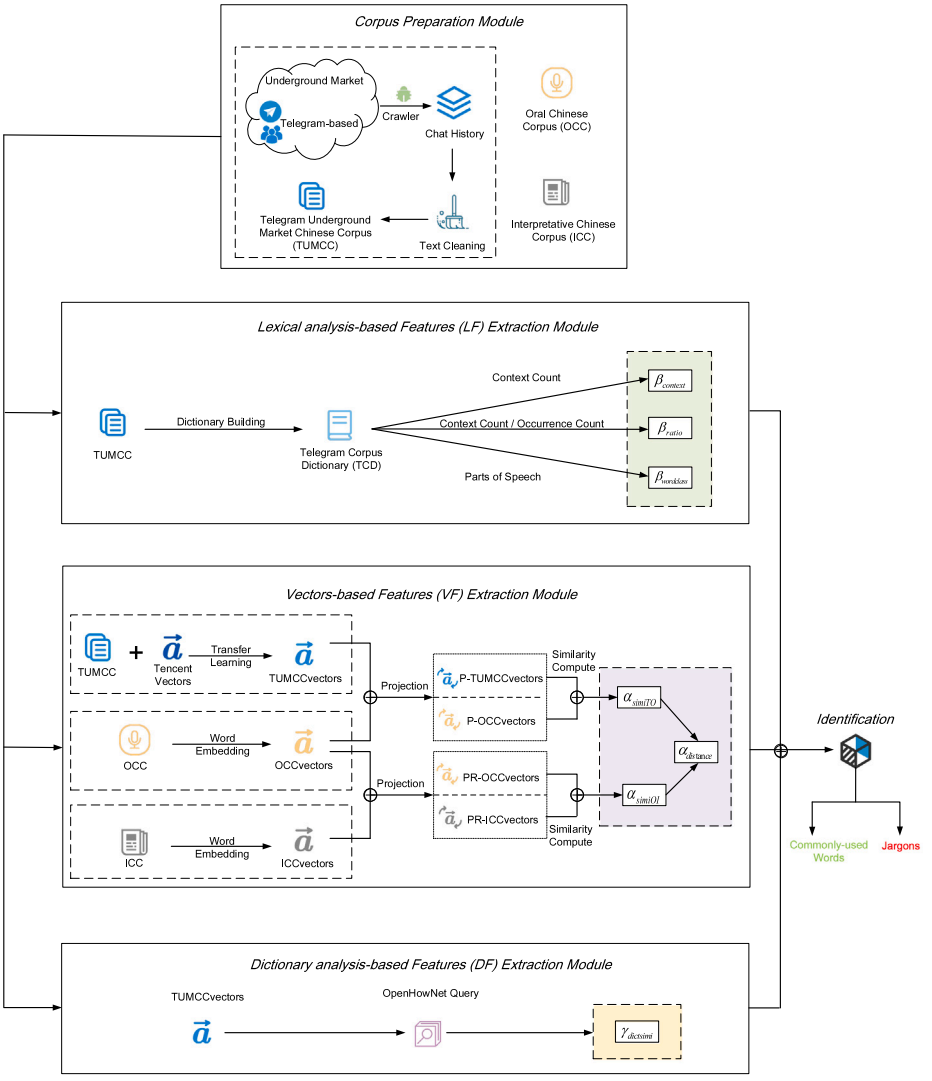


Fig. 1. The proposed Chinese Jargons Identification Framework, CJI-Framework.

4. Framework design and implementation

To identify Chinese jargons more effectively, we propose the CJI-Framework, as shown in Fig. 1. There are four modules in this framework: *Corpus Preparation Module*, *Lexical analysis-based Features (LF) Extraction Module*, *Vectors-based Features (VF) Extraction Module*, and *Dictionary analysis-based Features (DF) Extraction Module*. Firstly, we collect the corpus from Telegram underground markets and construct the ***TUMCC***. Secondly, we extract seven brand-new features that can be classified into three categories: the LF, VF, and DF. Finally, we run statistical outlier detection to decide whether a word is a jargon.

(1) ***Corpus Preparation Module.*** In this module, we collect chat history from 12 Telegram groups and construct a Chinese corpus for the following jargons identification work. Firstly, we develop a dedicated crawler to collect the chat history. Secondly, we filter and clean these texts, then label the dataset manually. Finally, the text is segmented by a tokenizer, along with punctuation and stop words being removed. Then the ***TUMCC*** is built, with a scale of 100,000 Chinese characters. What is more, we also construct the ***OCC*** of 832,186,054 characters, and the ***ICC*** of 162,988,659 characters, which are needed when computing the Vectors-based Features (VF), i.e. the features α_{simiTO}, α_{simiOI} and $\alpha_{distance}$.

(2) ***Lexical analysis-based Features (LF) Extraction Module.*** In this module, we construct the TCD which is needed when computing the Lexical analysis-based Features (LF). The TCD contains all words that appear in the ***TUMCC***, as well as the *context count* of a word, the *occurrence count* of a word, and its *word class*. The *context count* is corresponding to the feature $\beta_{context}$; we make use of the *context count* and the *occurrence count* to compute the feature β_{ratio}; the *word class* is related to the feature $\beta_{wordclass}$.

(3) ***Vectors-based Features (VF) Extraction Module.*** Cybercriminals usually use innocent-looking jargons instead of sensitive terms. The context around a word can usually reflect its meaning (Sasano & Korhonen, 2020). Thus, we can identify a jargon

Table 1
Overview of the proposed 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 count of context conditions (the *context count*)	β_{context}
	The ratio of the *context count* and the *occurrence count*	β_{ratio}
	Parts of speech (the *word class*)	$\beta_{\text{wordclass}}$
Dictionary analysis-based Features (DF)	The dictionary analysis results based on OpenHowNet	γ_{dictsimi}

Table 2
Overview of the targeted 12 telegram groups. (Members were counted on Nov.10th, 2020).

Group name	Overall group members	Members being collected	Sentences being collected	Characters being collected
Asiaweedy	8,325	1,286	2,011	54,271
RiotMarket	1,172	461	679	18,319
shegroup	50,003	5,211	7,750	216,921
awllc888	15,030	2,580	3,742	108,511
xddos10	3,277	615	899	25,082
MaXianNo99999	2,794	712	781	21,586
shanhaidanbao	1,068	396	1,803	46,116
wenjiandai	5,016	1,107	1,606	44,801
maiqiangwang	3,128	503	574	16,066
II0009	2,887	619	904	25,176
cntor	31,894	5,639	7,189	205,359
MaXianNo1	2,591	692	811	22,763
TOTAL	118,860	19,821	28,749	804,971

by finding out the difference between contexts. To capture the contextual difference of a word, it is necessary to convert natural language into word vectors that can be directly processed by a computer. Based on three corpora built in the first module, we generate three sets of word vectors: TUMCCvectors generated by the ***TUMCC***, OCCvectors generated by the ***OCC***, and ICCvectors generated by the ***ICC***. The outputs of this module are the features α_{simiTO}, α_{simiOI}, and $\alpha_{distance}$.

(4) ***Dictionary analysis-based Features (DF) Extraction Module.*** Besides word vectors, word sense disambiguation technology is another perspective of jargons identification. We carry the word sense disambiguation step based on the OpenHowNet dictionary to compute the feature $\gamma_{dictsimi}$.

The overview of the seven features is shown in Table 1. And finally, we run a statistical outlier detection (Gupta et al., 2013) to determine the thresholds of each feature. Only a word that meets the threshold of all features will be identified as a jargon by our framework. The four modules of the CJI-Framework and the final determination of dark jargons are introduced in detail below.

4.1. Corpus Preparation Module

Firstly, we collect chat history materials from targeted Telegram groups. Detailed information of the 12 selected Telegram groups is shown in Table 2.

To the best of our knowledge, there is no high-quality corpus for Chinese jargons identification currently, so we build the ***TUMCC*** by ourselves. Firstly, we select 12 Telegram groups that are active in online underground market transactions (selling guns, drugs, etc.). Then, we develop a dedicated crawler to collect 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 online underground markets can be gathered. A total of 28,749 sentences, including 804,971 Chinese characters, from 19,821 Telegram users were collected.

Secondly, we carry out three steps to clean and formalize the raw texts to build the ***TUMCC***. 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, punctuation and stop words were removed, and text segmentation was done by Jieba,[6] 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.

4.2. Lexical analysis-based Features (LF) Extraction Module

As shown in Fig. 1 above, we build the TCD (Telegram Corpus Dictionary) which contains words in the ***TUMCC*** and their properties: their *occurrence count*, their *context count*, and their *word class*. The TCD is used to compute three features of the LF.

[6] Jieba word segmentation. https://github.com/fxsjy/jieba, accessed on 2022-07-25.

(1) ***The features*** $\beta_{context}$ ***and*** β_{ratio}. Since the generation of word vectors relies on the analysis of different contexts (Spinde et al., 2021), a word should appear many times in the corpus. To filter out these words, we regard how many times a word has appeared as its *occurrence count*. Nevertheless, if a word always appears in the same context, we 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*. The *context count* is represented as the feature $\beta_{context}$. Besides, we consider the ratio of *context count* and *occurrence count* as a feature called β_{ratio}.

(2) ***The feature*** $\beta_{wordclass}$. We have found that in Chinese online 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)", "冰毒 (meth)" is called "冰糖 (rock candy)", 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, get the feature $\beta_{wordclass}$, and only keep nouns and verbs in our jargon list. To obtain the *word class* of all words (seen as the feature $\beta_{wordclass}$), we perform lexical analysis on the ***TUMCC***. To be more specific, experimental results in Section 5.6 lead us to apply the BaiduLAC tool[7] in the CJI-Framework.

4.3. Vectors-based Features (VF) Extraction Module

4.3.1. Implementation overview

Since jargons are unusual usages of words, their meanings are quite different between the ***TUMCC*** and other *general public Chinese corpora*. The meaning of a word can be inferred from its context (Levy & Goldberg, 2014; Zheng, Cai, Chen, & de Rijke, 2020). 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 the *general public Chinese corpora* into two categories when collecting data, namely the ***OCC*** (Oral Chinese Corpus) and the ***ICC*** (Interpretative Chinese Corpus), whose details are demonstrated in Table 3.

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 (Levy & Goldberg, 2014). Therefore, if the similarity of two word vectors is high, their corresponding words will have similar contexts, and vice versa. To capture this similarity value, we convert three corpora, ***TUMCC***, ***OCC***, and ***ICC***, into corresponding sets of word vectors: TUMCCvectors, OCCvectors, and ICCvectors. When comparing cross-corpus semantics, cosine similarity is seen as a common metric (Xia, Zhang, & Li, 2015). Its calculation is as below:

$$\cos\theta = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}|\cdot|\vec{y}|} = \frac{x_1 * y_1 + x_2 * y_2 + \cdots + x_n * y_n}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} * \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}} \tag{1}$$

We compare a word between the ***TUMCC*** and the ***OCC*** by calculating the cosine similarity, namely $Sim_{TUMCC-OCC}$. According to the theory proposed by Yuan et al. (2018), this value of a jargon should be remarkably lower than that of a commonly-used word, which can be a feature to characterize changes in the context. So if the value is lower than the similarity value of normal words, it indicates that the usage of the word is inconsistent between underground markets and oral situations, and it may be a jargon. Nevertheless, a single comparison is not enough, because the daily usage of certain words is extremely diverse, and the usage of a word may be quite different between formal situations and oral situations. For example, "粉丝" means "一种用绿豆、红薯淀粉等做成的丝状食品 (a kind of shredded food made of mung beans, sweet potato starch, etc.)" in formal situations, while in oral situations it implies "追星族, 狂热爱好者 (a fanatic lover of something)". If the ***OCC*** contains a certain percentage of formal situations, such as Wikipedia and official news, false positives will exist (because the similarity of a certain word, such as the word "粉丝" mentioned above, will be low between the ***TUMCC*** and the ***OCC***, thus it will be misjudged as a jargon). Therefore, the second comparison is to compute word similarities between the ***OCC*** and the ***ICC***, namely $Sim_{OCC-ICC}$. If the cosine similarity is lower than the similarity value of normal words, it indicates that the usage of a certain word is inconsistent between the oral situation and the formal situation. Such words will not be considered as jargons by our framework.

Specifically, the assumptions are: *(1) the meaning of a word between the* ***OCC*** *and the* ***ICC*** *is very similar, that is,* $Sim_{OCC-ICC}$ *(the feature* α_{simiOI}*) is bigger than the value of other words; (2) the meaning of this word between the* ***TUMCC*** *and the* ***OCC*** *is very different, that is,* $Sim_{TUMCC-OCC}$ *(the feature* α_{simiTO}*) is smaller than the value of other words; (3) this word have different meanings in two comparing groups, that is,* $\|sim_{TUMCC-OCC} - Sim_{OCC-ICC}\|$ *(the feature* $\alpha_{distance}$*) is larger than the average.* If one word meets three assumptions above, then it will be defined as a jargon by the 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. As a result, the VF category consists of three features, α_{simiOI}, α_{simiTO} and $\alpha_{distance}$.

When implementing the above two parts of comparisons, we have to solve the following two problems. The first challenge is related to the generation of TUMCCvectors; details of vectors generation can be seen in Section 4.3.2. The second challenge is related to vectors comparison; details of semantic comparison can be seen in Section 4.3.3.

[7] BaiduLac. https://github.com/baidu/lac, accessed on 2022-07-25.

Table 3
Overview of the Oral Chinese Corpus (*OCC*) and the Interpretative Chinese Corpus (*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

4.3.2. Vectors generation based on word embedding and transfer learning

(1) ***Generate the OCCvectors and ICCvectors***. The effective training of word vectors should be based on a large-scale corpus (Peters, Neumann, Zettlemoyer & Yih, 2018). Both the ***OCC*** and the ***ICC*** have reached a million-level scale, so GloVe, i.e. a word embedding model, can be utilized to directly generate the OCCvectors and the ICCvectors. It can use a numeral vector to represent the meaning of a word (Artetxe, Labaka, & Agirre, 2018). These vectors can capture semantic features, such as similarity and analogy. In this module, we employ GloVe to transform the ***OCC*** and the ***ICC*** into vectors sets: the OCCvectors and the ICCvectors.

(2) ***Generate the TUMCCvectors***. At least a million-level corpus is required to build high-quality word vectors (Peters, Neumann, Zettlemoyer & Yih, 2018). Nevertheless, there are two reasons which lead to a limited scale of corpus: (1) The concealment of transactions in online underground markets often makes it tough for cybersecurity researchers to find enough materials to build a large-scale and high-quality corpus to train word vectors (Thomas, McCoy, Grier, Kolcz, & Paxson, 2013). (2) Labeling jargons is hardly carried out on a large corpus because of time and energy limitations, so we can only build a limited scale of the corpus after data-labeling. Therefore, to reduce the reliance on the scale of labeled ***TUMCC***, we apply a transfer learning method to build TUMCCvectors based on a pre-trained high-quality word vectors set. We construct a character-level VCDM (Variational Contextual Definition Modeler) (Reid, Marrese-Taylor, & Matsuo, 2020) to do transfer learning. The VCDM consists of three parts: *Encoders*, *Neural Definition Inferer*, and *Variational Definition Modeler*. To be specific, the generation process of new vectors can be formulated as the following generative probabilistic model:

$$p(d||w) = \int_z p(d, z \mid w) d_z = \int_z p(d \mid z, w) p(z \mid w) d_z \tag{2}$$

where the w is the input vectors of transfer learning, and the d is the output vectors. The joint semantics of (w, d) are captured by introducing the latent variable z, and the conditional probability $p(d \mid w)$ evolves into $p(d \mid w, z)$. That is to say, the generation of the d is conditioned on both the w and the z.

The generated TUMCCvectors can represent the new usage of words in the ***TUMCC***. To select the most appropriate pre-trained Chinese word vectors set, we carry out comparison experiments in Section 5.2. Experimental results show that the public Tencent Vectors performs best. So we choose the Tencent Vectors as the basis of transfer learning.

4.3.3. Semantic comparison based on unsupervised word vector projection

To compute the $Sim_{TUMCC-OCC}$ (i.e. the feature α_{simiTO}), as well as the $Sim_{OCC-ICC}$ (i.e. the feature α_{simiOI}), 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 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 (experimental results in Section 5.4 prove this). Therefore, if the word vector generation model is used directly to train two independent vectors sets, the outcome is not comparable. In other words, the TUMCCvectors and the OCCvectors cannot be compared directly. The OCCvectors and the ICCvectors also cannot be compared directly for the same reason. Nevertheless, as all corpora share concepts that are grounded in the real world (Le & Mikolov, 2014), vector spaces of separately trained vectors are usually similar, and words with similar meanings are always close in the space (Liu, Ungar, & Sedoc, 2019). In this case, to solve the problem that word vectors generated independently cannot be compared directly, we apply the transfer matrix method (Zhang, Xiong, & Su, 2018) used in machine translation. We build a synonym dictionary based on reinforcement learning, and further calculate the transfer matrix to implement the projection of vectors. Thus, these vectors are projected into one shared space, where they can be comparable. The transfer matrix W maps 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 is a set of word pairs and vector representation associated with them, and 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 the word i. The loss function can be defined as follows, using the distance between the corresponding words in two corpora:

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

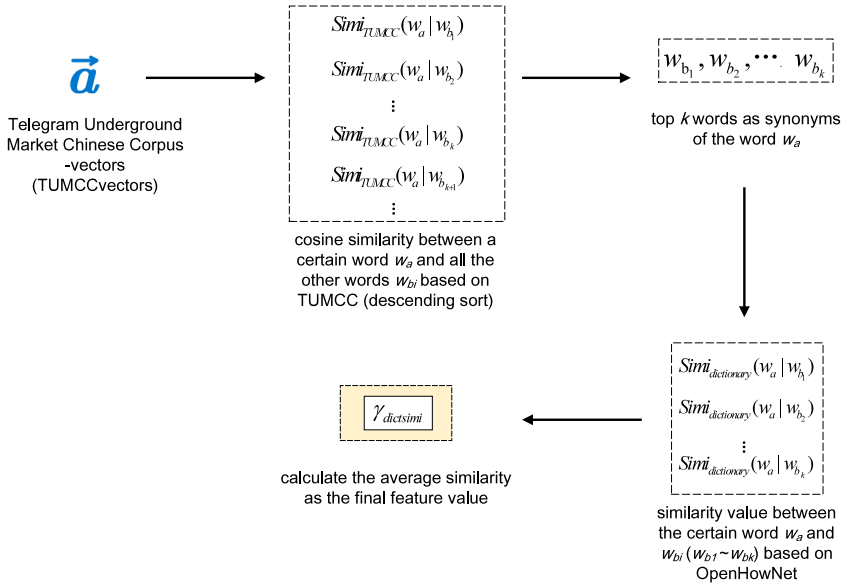


Fig. 2. Overview of the feature extraction using OpenHowNet in the *Dictionary analysis-based Features (DF) Extraction Module*.

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 (Artetxe et al., 2018) 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 the word vectors preprocessing step to orthogonalize word vectors. Secondly, we build a preliminary seed library based on the multi-dimensional comparison of each word in the vectors set. Furthermore, we apply reinforcement learning to optimize its quality repeatedly. Finally, symmetric re-weighting (Tamaazousti, Le Borgne, Hudelot, Seddik, & Tamaazousti, 2020) 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. Note that no special processing is required to exclude dark jargons from the synonym dictionary, for Section 5.5 will show that the existence of jargons in it does not influence the identification performance.

As shown in Fig. 1 above, after projection, there are two pairs of comparable word vectors being generated: the TUMCCvectors and the OCCvectors are projected to the P-TUMCCvectors and the P-OCCvectors; the OCCvectors and the ICCvectors are projected to the PR-OCCvectors and the PR-ICCvectors. Then, they can be used to compute the VF.

4.4. Dictionary analysis-based Features (DF) Extraction Module

Word sense disambiguation technology (Raganato, Camacho-Collados, & Navigli, 2017) can help determine the most accurate meaning of a certain word based on the context in which it is found. In this case, we apply it to find out words with abnormal meanings that are not contained in the formal dictionary, which is another perspective at the phase level for jargons identification. Therefore, we carry the word sense disambiguation step based on OpenHowNet dictionary. In particular, we combine the OpenHowNet with a similarity comparison method based on sememes (Niu, Xie, Liu, & Sun, 2017) to find synonyms. Synonyms of a word are related to its real meaning (Qian, Feng, Wen, & Chua, 2021), based on which we can further confirm whether the word has a significantly different meaning in TUMCC and thus is a potential jargon.

The calculation method of the feature $\gamma_{dictsimi}$ in the DF is shown in Fig. 2. Suppose we need to calculate the $\gamma_{dictsimi}$ value of word w_a. Firstly, we need to calculate the cosine similarity between a certain word w_a and all the other words w_{b_i} appearing in TUMCCvectors, and then sort the values in descending order, taking the top k words $w_{b_1} \sim w_{b_k}$ as synonyms of the word w_a. These synonyms suggest the true meaning of word w_a in ***TUMCC***. Next, the cosine similarity values between the word w_a and its each synonym $w_{b_1} \sim w_{b_k}$ are computed again, being based on the OpenHowNet dictionary this time, reflecting how similar these words are in a common environment with no jargon. After that, we then calculate the average similarity to obtain the feature $\gamma_{dictsimi}$. If a word is used commonly, the average similarity mentioned above should be higher than the exact threshold value, and vice versa. Therefore, this feature is also helpful for identifying jargons from commonly-used words. We have undertaken an experiment to determine the ideal value of k, and found out that the relatively most appropriate one is $k = 20$, so 20 is selected for subsequent experiments.

We apply the OpenHowNet dictionary to calculate the similarity of two words. It provides rich information about the hierarchy of sememes. For two words w_{s_1} and w_{s_2} to calculate their similarity based on OpenHowNet, we start from their nodes in the sememe

tree and then find the corresponding words with the same relationship to the root word for similarity calculation. The calculation method is as follows:

$$Simi_{dictionary}\left(w_{s_1} \mid w_{s_2}\right) = \frac{1}{N}\sum_{i=1}^{N} Simi_{rela}\left[\left(w_{s_1}{}'\right)_i \mid \left(w_{s_2}{}'\right)_i\right] \tag{5}$$

where N denotes that there are N corresponding interrelations, and $\sum_{i=1}^{N} Simi_{rela}\left[\left(w'_{s_1}\right)_i \mid \left(w'_{s_2}\right)_i\right]$ is the similarity value of certain interrelation i.

To conclude, three categories of features have been extracted based on the following: the LF based on the TCD, the VF based on two pairs of comparable word vectors, and the DF based on dictionary analysis from the public Chinese dictionary OpenHowNet. Finally, we adopt an outlier detection method to implement the final determination step of jargons identification. Only a word that meets the threshold of all features will be judged as a jargon by our framework. Specifically, the statistical method based on Tukey box plot is applied, which is improved from three standard deviations, i.e. the empirical rule. Define a threshold set $X = \{x_1, x_2, \ldots, x_i, \ldots\}$, in which x_i is an exact threshold for a certain feature. The threshold value is calculated by IQR (Inter Quartile Range), as shown below:

$$x_i = Q3 + k(IQR) \tag{6}$$

or

$$x_i = Q1 - k(IQR) \tag{7}$$

where $Q3$ is the upper quartile, $Q1$ is the lower quartile, $IQR = Q3 - Q1$, and $k \geq 0$. In our research, we use k=1.5 to find thresholds in each module. Whether to use Formula (6) or Formula (7) to calculate the threshold x_i depends on which feature we are working on. To be specific, if the outlier condition of a certain feature is relatively larger in the result set, Formula (6) will be used, otherwise, Formula (7) is responsible.

The effect of the overall framework will be evaluated in Section 5 below.

5. Experiments

In this section, we evaluate the performance of the proposed CJI-Framework for Chinese jargons identification. Experiments were undertaken on a server with an Intel Xeon (R) Gold 6130 CPU with 128 GB memory, and a Tesla V100 GPU with 32 GB video memory. All experiments were repeated ten times independently to obtain the average results.

Four metrics are used for effect evaluation in some of the following experiments, including Accuracy, Precision, Recall, and F1-score:

$$Accuracy = \frac{|TP + TN|}{|TP + FP + FN + TN|} \tag{8}$$

$$Precision = \frac{|TP|}{|TP + FP|} \tag{9}$$

$$Recall = \frac{|TP|}{|TP + FN|} \tag{10}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

where TP (True Positive) is the number of jargons that are accurately identified, FP (False Positive) is the number of commonly-used words that are mistakenly regarded as jargons, FN (False Negative) is the number of jargons that are mistakenly regarded as commonly-used words, and TN (True Negative) 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.

5.1. Datasets

The proposed CJI-Framework depends on three corpora, which are the ***OCC***, the ***ICC***, and the ***TUMCC***. To prepare the ***TUMCC***, we collect chat history from targeted Telegram groups and carry out three steps to clean and formalize the raw texts. Detailed information about the ***TUMCC*** has been introduced in Section 4.1. The ***TUMCC*** contains 3,863 sentences (a total of 100,000 characters) from 3,139 Telegram users.

Besides the self-built ***TUMCC***, the CJI-Framework also uses the public ***OCC*** and the public ***ICC***. The ***OCC*** consists of public Weibo, Tieba, and Douban corpus[8]; the ***ICC*** is from the public Chinese Wikipedia dataset.[9] Thus, the ***OCC*** represents oral Chinese materials while the ***ICC*** represents formal usage of Chinese. An overview of the two corpora is shown in Table 3 above.

[8] Chinese Word Vectors. https://github.com/Embedding/Chinese-Word-Vectors, accessed on 2022-07-25.

[9] Chinese Wikipedia Dataset. https://dumps.wikimedia.org/zhwiki/latest/, accessed on 2022-07-25.

Table 4
Comparison of jargons identification results when different vectors sets are used as the origin of transfer learning.

Vectors Set	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
OCCvectors	29.17	33.33	13.33	19.04
Wikipedia Vectors	45.83	62.50	33.33	43.48
Weibo Vectors	62.50	87.50	46.67	60.87
ICCvectors	58.33	72.73	53.33	61.54
Tencent Vectors	87.50	92.86	86.67	89.66

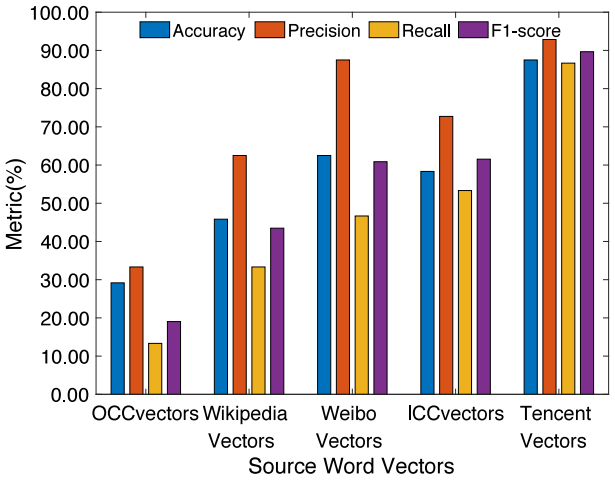


Fig. 3. Comparison of jargons identification results when different vectors sets are used as the origin of transfer learning.

5.2. Evaluation of different transfer learning sources

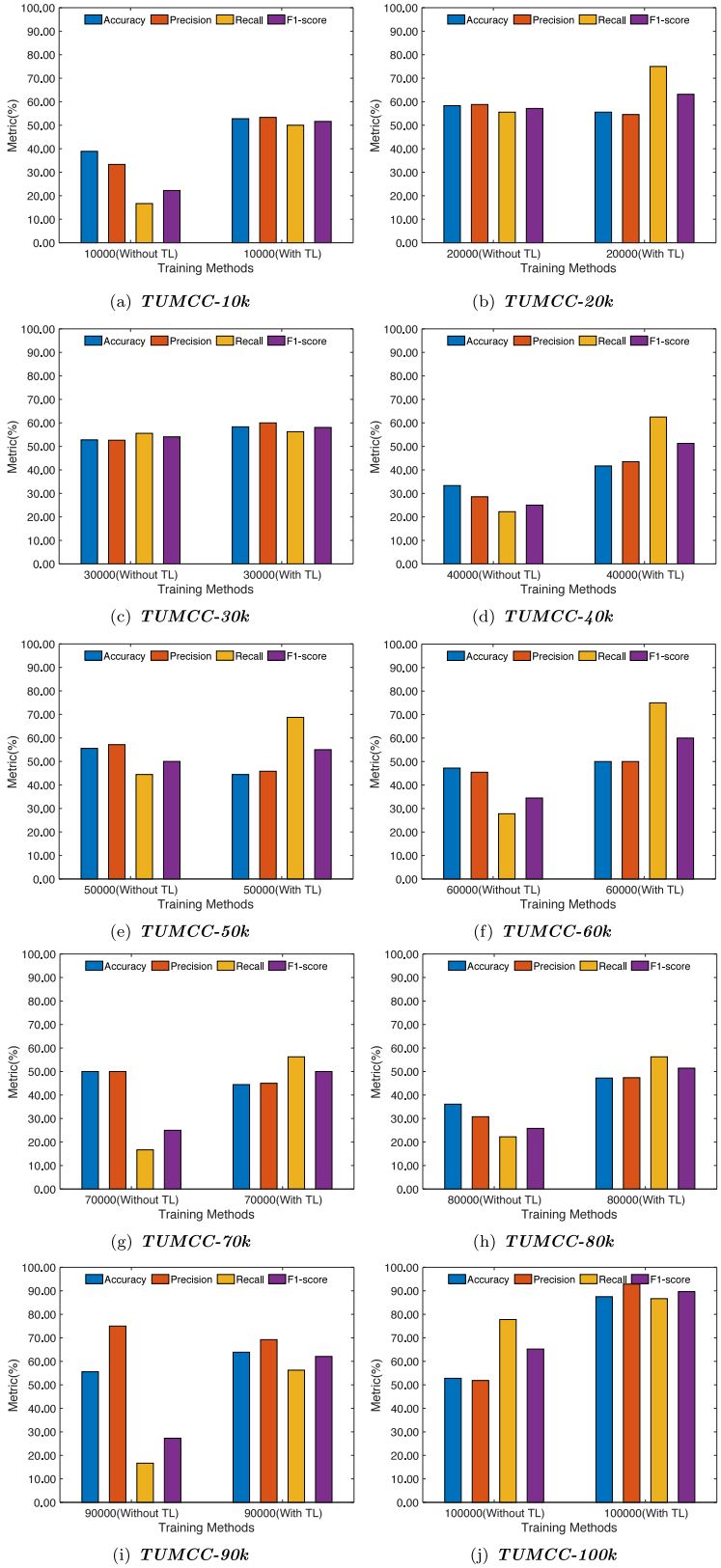
In this section, we evaluate the word vectors set used for transfer learning. As we elaborated in Section 4.3.2, the CJI-Framework utilizes a set of Chinese vectors to train the TUMCCvectors. To determine the basis of transfer learning, we evaluate five sets of Chinese word vectors, which are self-built ICCvectors, self-built OCCvectors, public Tencent Vectors,[10] public Wikipedia Vectors,[11] and public Weibo Vectors.[12] With these five vectors sets individually and the ***TUMCC*** as inputs, the TUMCCvectors are generated, and then the whole jargons identification procedure is taken.

It can be seen in Table 4 and Fig. 3 that Weibo Vectors and ICCvectors achieve similar performance, which is much better than that of OCCvectors and Wikipedia Vectors. Among all source vectors, Tencent Vectors got the best result with an F1-score of 89.66%. As a result, Tencent Vectors is the most appropriate selection when applying the transfer learning method to generate TUMCCvectors.

5.3. Evaluation of the validity of transfer learning using different sizes of *TUMCC*

To demonstrate the effectiveness of the transfer learning method, we divide the ***TUMCC*** into ten subsets with an equal number of Chinese characters randomly. Each of them contains 1,000 characters. We use one to ten subsets (from ***TUMCC-10k*** to ***TUMCC-100k***, a scale of 10,000 to 100,000 Chinese characters) for transfer learning and jargons identification. Based on various sizes of the corpus, we can evaluate the results and determine which method is better: using the transfer learning method to train TUMCCvectors, or applying GloVe directly to generate TUMCCvectors. We use Tencent Vectors as the source vectors. The results are shown in Fig. 4. It can be concluded that: (1) Although the direct training of word vectors by GloVe can identify Chinese jargons, the performance when the transfer learning method being applied to whichever size of the corpus has all been significantly improved. This indicates that it is valid to employ the transfer learning method. (2) The F1-score is 89.66% on the corpus of ***TUMCC-100k*** (i.e. ***TUMCC***), which achieves the best result.

[10] Tencent Vectors. https://ai.tencent.com/ailab/nlp/en/index.html, accessed on 2022-07-25.
[11] Public Chinese Wikipedia Dataset. https://dumps.wikimedia.org/zhwiki/latest/, accessed on 2022-07-25.
[12] Public Weibo Vectors. https://github.com/Embedding/Chinese-Word-Vectors, accessed on 2022-07-25.



(a) *TUMCC-10k* (b) *TUMCC-20k*

(c) *TUMCC-30k* (d) *TUMCC-40k*

(e) *TUMCC-50k* (f) *TUMCC-60k*

(g) *TUMCC-70k* (h) *TUMCC-80k*

(i) *TUMCC-90k* (j) *TUMCC-100k*

Fig. 4. Comparison of jargons identification results based on direct training and transfer learning (TL) on different sizes of the corpus.

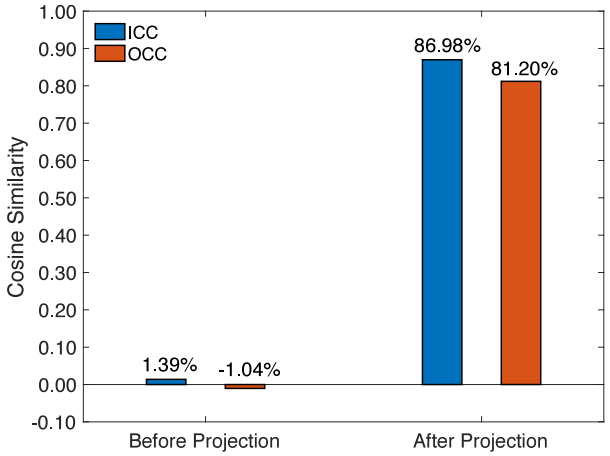


Fig. 5. The comparison results of corresponding word vectors cosine similarity before and after projection.

5.4. Evaluation of the validity of word vector projection

Because of the randomness of initial parameters such as weight parameter initialization in GloVe, even if word vectors are generated twice for the same corpus, the corresponding word vectors of the same word will still be quite different, thus their similarity value will be low. In other words, in the CJI-Framework, the vectors projection step of the *Vectors-based Features (VF) Extraction Module* is necessary to make word semantics cross-corpus comparable. To evaluate the validity of word vector projection, we generate word vectors from the ***OCC*** and the ***ICC*** twice, correspondingly. We compute word vectors' cosine similarity between the two outputs. Next, we conduct the word vector projection step based on the two outputs, and then compare corresponding word vectors of the same word again. The comparison results of corresponding word vectors cosine similarity before and after projection are shown in Fig. 5.

Experimental results show that, for the corpus ***ICC***, when word vectors are generated twice repeatedly, the average cosine similarity of each word's vectors is 1.39%; to compare, this value rises to 86.98% after the projection step. Besides, for the corpus ***OCC***, the value is −1.04% before projection and 81.20% after projection. Both the value of ***ICC*** and ***OCC*** become significantly higher, which indicates that the projection step can eliminate the influence of training randomness, so the word vectors after projection can represent a word stably. This shows that the projection step is necessary and effective to control variables, which makes the vectors generated independently comparable. As can be seen in the CJI-Framework overview in Fig. 1 above, after TUMCCvectors and OCCvectors are generated separately, we conduct the projection step to get P-TUMCCvectors and P-OCCvectors, and corresponding vectors in them can be compared. It is the same for generating PR-OCCvectors and PR-ICCvectors.

5.5. Evaluation of the effectiveness of semantic comparison

To evaluate the effectiveness of semantic comparison, and represent the capacity of the Vectors-based Features (VF) to capture word semantics between two corpora, we make some random words be "simulated jargons" to examine whether they can be identified as jargons correctly. To be more specific, by replacing the word W_a with another word W_b (so W_b refers to W_a's meaning), W_b becomes a jargon. If the CJI-Framework can capture the semantics of jargons in a cross-corpus environment, it will identify the W_b, i.e. the abnormal word of non-common usage within two different corpora, which is a simulated jargon.

Fig. 6 shows the process which creates "simulated jargons" and evaluates the reasonableness of semantic comparison to identify jargons. Dataset A is a normal corpus, and Dataset B contains a simulated jargon W_b; the example in the figure is the word "快递 (package)". The word-embedding step generates vectors that can be processed directly by the computer, and the projection step can make the pair of vectors comparable. Based on experiment 5.4, the cosine similarity of the same word between similar contexts should be high, because its semantics have not changed a lot; meanwhile, if a word is exactly the jargon, the CJI-Framework should capture its inconsistent semantics, with low before–after similarity. In this case, word vectors of the simulated jargon W_b should have a lower value of cosine similarity than other commonly-used words.

In this experiment, we use the public WikiPedia[13] as the input dataset. The numerical choice of substituted words (i.e. simulative jargons) is set discretely ranging from 100 to 500, and the words are chosen randomly. Then we compute the average similarity of non-substituted words and substituted words, to examine whether the results can distinguish the simulated jargon. Experimental results in Table 5 show that the average cosine similarity of simulated jargons is always significantly lower than that of other commonly-used words. It indicates that the CJI-Framework can identify the cross-corpus semantics of a jargon and achieves a high performance.

[13] Chinese Wikipedia Dataset. https://dumps.wikimedia.org/zhwiki/latest/, accessed on 2022-07-25.

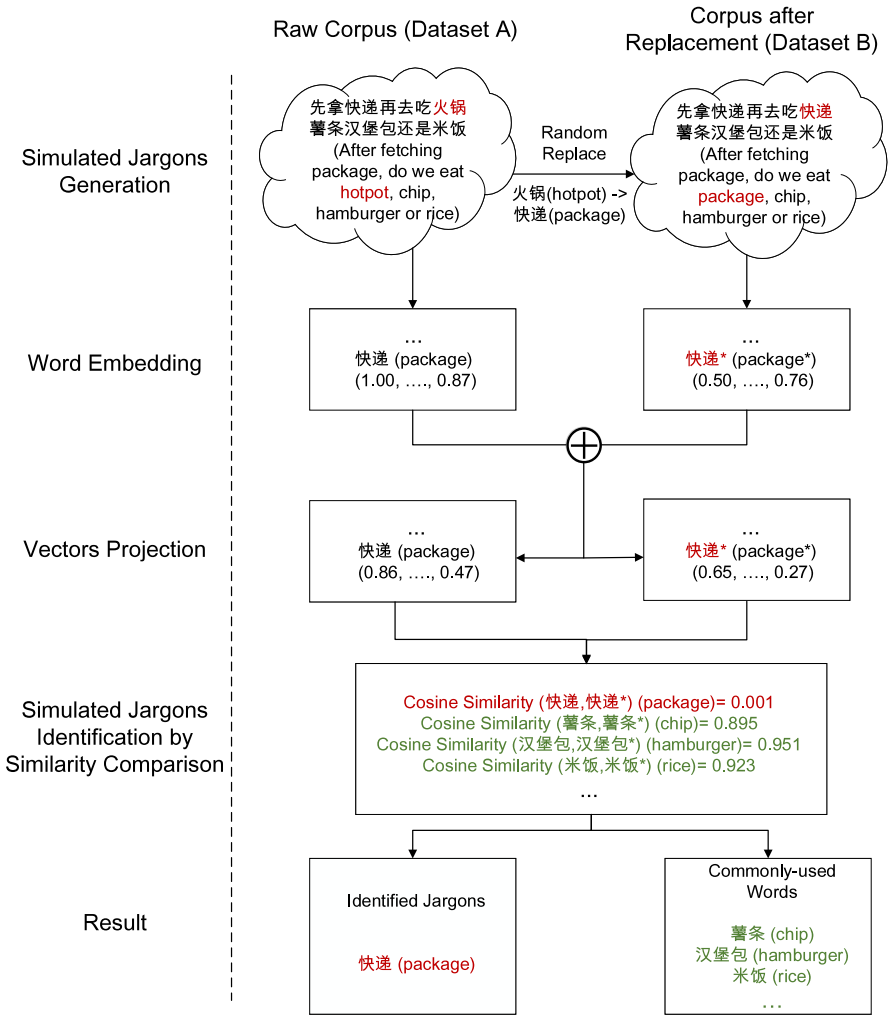


Fig. 6. Overview of the process that creates "simulated jargons" and evaluates the reasonableness of semantic comparison to identify jargons. The simulated jargon "快递 (package)" has a much lower similarity value than the average.

Table 5
The average cosine similarity of simulated jargons and other commonly-used words.

Number of substituted words	Average similarity of simulated jargons (%)	Average similarity of other commonly-used words (%)
100	0.44	85.73
200	3.99	85.60
300	1.47	85.51
400	4.75	85.45
500	5.06	85.68

Table 6
The CJI-Framework maintains a high accuracy rate with various quantity of simulated jargons.

Number of Simulated Jargons	100	200	300	400	500
Accuracy (%)	98.97	99.47	99.65	98.96	98.33

In addition, we also record the percentage of simulated jargons whose before–after similarity is lower than that of commonly-used words. The value shows how well jargons can be precisely identified. As Table 6 shows, the CJI-Framework always maintains good performance at a high accuracy rate with various quantities of simulated jargons.

Table 7
Jargons identification results of different feature sets.

Features sets	Categories of features included	Accuracy	Precision	Recall	F1-score
F	VF+LF+DF	87.50	92.86	86.67	89.66
F/DF	VF+LF	87.30	76.47	68.42	72.22
F/LF	VF+DF	65.08	45.45	78.95	57.69
F/VF	LF+DF	55.56	39.53	89.47	54.83

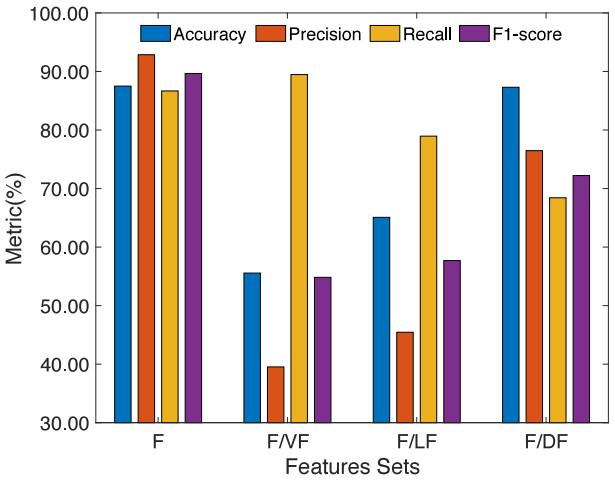


Fig. 7. Jargons identification results of different feature sets.

5.6. Evaluation of feature effectiveness

We extract seven brand-new features and they are divided into 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-100k*** (i.e. the ***TUMCC***) corpus. 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 feature subsets can be elaborated by the following set-difference function:

$$F/F' = \left\{x \mid x \in F \wedge x \notin F'\right\} \tag{12}$$

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.

The experimental results of jargons identification with F, F/DF, F/LF, and F/VF are shown in Table 7 and Fig. 7. It demonstrates that: (1) When seven features of three categories are all taken into consideration at the same time, we get the best result of an F1-score of 89.66%. Excluding any category of features will lead to a decrease in performance. (2) Among all conditions, the approach performs worst with an F1-score of 54.83% when using the feature set of F/VF, which indicates that the validity of Vectors-based Features (VF) is the greatest. (3) The validity of feature categories can be sorted from highest to lowest as the VF, the LF, and the DF. The contribution of the VF is 2.86% higher than the LF, and 17.39% higher than the DF.

Furthermore, there are various popular lexical analysis algorithms. To confirm the most appropriate lexical analysis algorithm when calculating $\beta_{wordclass}$, we compare five common candidates: THULAC,[14] CoreNLP,[15] LTP,[16] HanLP,[17] and BaiduLAC.[18] Then we will carry on the whole jargons identification process to obtain the final result. The final experimental results are shown in Fig. 8. Experimental results show that the BaiduLAC algorithm achieves the highest F1-score. This shows that it has an advantage compared with other analysis algorithms. Therefore, when implementing the CJI-Framework, BaiduLAC lexical analysis algorithm is adopted to build the TCD.

[14] THUNLP. http://thulac.thunlp.org/, accessed on 2022-07-25.
[15] CoreNLP. https://stanfordnlp.github.io/CoreNLP/, accessed on 2022-07-25.
[16] LTP. http://www.ltp-cloud.com/, accessed on 2022-07-25.
[17] HanLP. https://www.hanlp.com/, accessed on 2022-07-25.
[18] BaiduLac. https://github.com/baidu/lac, accessed on 2022-07-25.

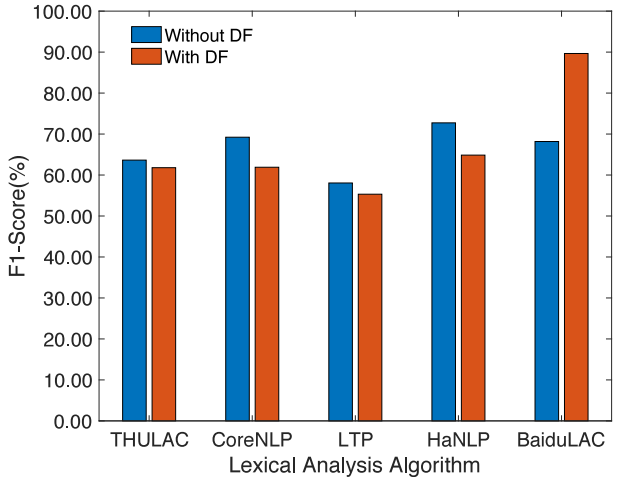


Fig. 8. Results when different lexical analysis algorithms are being applied or not applied to carry on the whole jargons identification process.

Table 8
Overview of the Jargon Corpus, Oral Corpus and Interpretative Corpus (all in English).

Corpus (Statistical Objects)	Sources	Numbers
Jargon Corpus (Sentences)	CantReader Raw Text[a]	12,406
Jargon Corpus (Words)	CantReader Raw Text	4,005,425
Oral Corpus (Sentences)	Reddit Dataset[b]	30,000
Oral Corpus (Words)	Reddit Dataset	8,524,574
Interpretative Corpus (Sentences)	Public Wikipedia-Eng Dataset[c]	219,582
Interpretative Corpus (Words)	Public Wikipedia-Eng Dataset	11,198,682

[a]CantReader Raw Text. https://drive.google.com/file/d/1E9nQH8btRDu7zJDnl2_jRhtPIBFO3wKQ/view, accessed on 2022-07-25.
[b]Reddit Dataset. https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/, accessed on 2022-07-25.
[c]Public Wikipedia-Eng (enWiki) Dataset. https://dumps.wikimedia.org/enwiki/, accessed on 2022-07-25.

5.7. Evaluation of the proposed framework

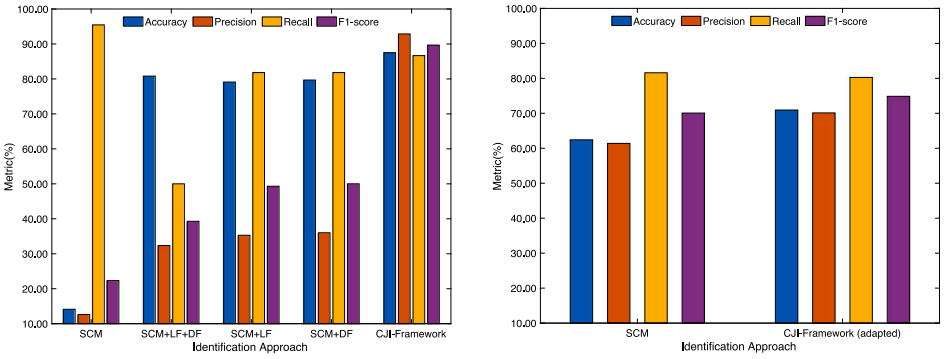
The CJI-Framework contains four modules and extracts seven features for Chinese jargons. It has achieved ideal results for Chinese jargons identification task. Before that, an effective model in the area of jargons identification was the SCM model (Yuan et al., 2018) aimed at English jargons. By improving Word2Vec, the SCM makes it possible to train two comparable word vectors sets at the same time, avoiding the influence of randomness. Nevertheless, it does not utilize the various features of jargons. To evaluate the CJI-Framework, we use the SCM and the CJI-Framework separately to carry a Chinese jargons identification task based on the ***TUMCC***, ***OCC***, and ***ICC***. To evaluate the adaptability and scalability of the SCM for comparison, we also add the feature category of the LF and the DF for it. On the other hand, we would also like to evaluate the SCM model and the CJI-Framework by carrying out an English dark jargons identification task. Nevertheless, the CJI-Framework is implemented specifically for Chinese. Therefore, in order to let the CJI-Framework be capable of handling English dark jargons, we need to replace several corresponding modules. Firstly, we changed the word splitting step of the raw corpus in the *Corpus Preparation Module*. Secondly, we replaced Chinese-targeted BaiduLAC with the general NLTK toolkit[19] as the word class tagger in the *Lexical analysis-based Features (LF) Extraction Module*. Moreover, we changed the source word vectors set for the transfer learning step from public Tencent Vectors (Chinese) to GloVe Pre-trained Word Vectors[20] (English) in the *Vectors-based Features (VF) Extraction Module*. And finally, we used the corresponding English dictionary query tool WordNet[21] instead of OpenHowNet in the *Dictionary analysis-based Features (DF) Extraction Module*. The English datasets used by the CJI-Framework and the SCM model are shown in Table 8.

Experimental results are shown in Table 9 and Fig. 9. (1) For Chinese jargons identification: It can be seen that the F1-score of the SCM is only 22.32% in the Chinese environment. After combining the LF, the performance of SCM has been effectively improved, with an F1-score of 49.31%. Besides, with the DF being added, the F1-score reaches 50.00%. When the LF and the 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

[19] Natural Language Toolkit (NLTK). https://www.nltk.org/, accessed on 2022-07-25.
[20] GloVe Pre-trained Word Vectors. https://nlp.stanford.edu/projects/glove/, accessed on 2022-07-25.
[21] WordNet. https://wordnet.princeton.edu/, accessed on 2022-07-25.

Table 9
The performance of jargons identification approaches for Chinese and English.

Approach	Corpora composition	Corpus language	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SCM	Telegram Underground Market Chinese Corpus, Oral Chinese Corpus, Interpretative Chinese Corpus.	Chinese	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**
SCM	Jargon Corpus, Oral Corpus, Interpretative Corpus.	English	62.41	61.39	**81.58**	70.06
CJI-Framework (adapted)			**70.92**	**70.11**	80.26	**74.85**



(a) for Chinese (b) for English

Fig. 9. The performance of jargons identification approaches for Chinese and English.

and recall rate, the final metric, F1-score, is still low (the exact value is 39.28%). As a comparison, 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 module design of the CJI-Framework is targeted at Chinese while the SCM model is proposed specifically for English, so the process and implementation of SCM cannot be transferred directly to Chinese; even if it adds the DF and the LF, the CJI-Framework can still achieve better performance. (3) The SCM model is not optimally designed for a small-scale corpus with dark jargons, while the CJI-Framework contains more targeted and interpretable features for Chinese on the premise of a small-scale dataset, and extracts a total of seven features in three categories. In this case, we would like to recommend the CJI-Framework for Chinese jargons identification. (2) For English jargons identification: Though the CJI-Framework (adapted) has a lower recall rate due to the relatively strict strategy of being identified as a jargon, it can be seen that its F1-score of 74.84% is still 4.78% higher than the SCM model. It preliminarily shows that our framework is cross-language adaptable, and it performs better than the SCM model in both the Chinese and English environments.

6. Discussion and implications

6.1. Implications

Our goal is to focus on the identification of dark jargons for Telegram Chinese underground markets. Below are key points that require consideration: (1) Method adaptability for the language type and corpora source. Different languages need targeted syntactic and semantic analysis methods. As for Chinese, one highlighted characteristic is that a meaningful word phrase usually contains more than one Chinese character. Thus methods for other languages, for example, (Yuan et al., 2018; Zhu et al., 2021) for English, (Aoki et al., 2017; Hada et al., 2020) for Japanese, may not suit our task. (2) The lack of a well-prepared jargon list, and the dependence on annotated texts. All the studies of Aoki et al. (2017), Hada et al. (2020) and Zhao et al. (2016) require a well-prepared jargon list, which is hard to satisfy due to the concealment of underground markets and the difficulty of understanding jargons as an outsider. Thus, an improved method is required for better identification performance.

To address the above points, we build the ***TUMCC*** dataset and propose the CJI-Framework. **Firstly**, the lexical analysis, vectors generation and projection, and dictionary analysis methods work together and perform well, focusing on dealing with Chinese and filling the blank of identifying Chinese jargons for IM software (e.g. Telegram). This is highly valuable in the context of IM communications, where dark jargons are rapidly evolving. **Secondly**, our method does not require a ground-truth jargon list as input, which is more practical than previous studies. **Moreover**, we introduce the transfer learning technology to the word vectors generation procedure. Word vectors generation depends on a large-scale corpus and suitable parameters, which limit the application of this technology. However, with the help of transfer learning, we can generate valid word vectors easily to capture

jargons accurately. **And finally**, the new features we propose are interpretable and extensible. We have released the first publicly accessible Chinese corpus containing the chat history of Telegram groups related to transactions in underground markets, which makes our work friendly for subsequent researchers to reproduce and develop the procedures. Based on our above work, it is possible to further detect the usage, distribution, and interpretation of dark jargons. This will be helpful for investigations into cybercrime and the underground ecosystem.

6.2. The context-based embedding is not required in this study

We tried ELMo (Peters, Neumann, Iyyer et al., 2018) in the early stages of our study, but it did not work well. We believe that the Out of Vocabulary (OOV) situations tend to appear frequently with morphology-rich data, in which case ELMo will work better than GloVe; however, Chinese happens to be a language with limited morphology, so it may affect the performance of ELMo. Thus, we choose the non-semantic general embedding model followed by a necessary word sense disambiguation step, i.e. the *Dictionary analysis-based Features (DF) Extraction Module*. We think it has already reached a good performance on our task. Meanwhile, we can ensure the rationality and interpretability of the framework during its design and implementation.

6.3. The selection of applying an outlier detection method

In this paper, we tried to identify dark jargons from the deep perspective of word-level, and have extracted seven new features of three categories, the LF, VF and DF; each feature we propose is highly interpretable, with their performance being evaluated in Section 5.6. Since each feature can be used as a perspective for jargons identification, we think a rule-based threshold mechanism can help us highlight the contribution of each feature with controllable weights. The existing studies show that rule-based determination method is clear and controllable, and is easy to be improved and extended (Boukerche, Zheng, & Alfandi, 2020). In this case, we think the classicality and robustness of the threshold mechanism based on outlier detection make it more appropriate for our study than other methods. Details of the jargons determination step can be seen at the end of Section 4.

6.4. Limitations and future work of the CJI-Framework

There are still some considerations: (1) the framework we designed for Chinese needs further work when it is applied to other languages. Experimental results have demonstrated a preliminary cross-language applicability of the CJI-Framework, and we think there is still a need for fine-grained adaptations for different languages to enhance the framework generality; (2) feature engineering can continue to be extended. The seven features we proposed are linguistically general, and there may be more to be added to our framework based on different languages.

7. Conclusion

This paper proposes a novel CJI-Framework to identify jargons in Chinese online underground markets automatically. We extract a total of seven features in three categories aimed at Chinese, including the VF, the LF, and the DF, to distinguish between jargons and commonly-used words. Specifically, by employing a transfer learning method for word vectors generation, the CJI-Framework reaches good performance when computing the VF, which contributes the most among three categories of features, i.e. 2.86% higher than the LF and 17.39% higher than the DF. Furthermore, in order to evaluate our framework, we construct the ***TUMCC*** with a scale of 100,000 Chinese characters, and it is the first Chinese corpus containing chat history of Telegram groups related to transactions in online underground markets. The experimental results show that the CJI-Framework reaches an F1-score of 89.66% for Chinese jargons identification, and is also 4.78% more efficient for English than the state-of-the-art.

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.

CRediT authorship contribution statement

Yiwei Hou: Methodology, Software, Validation, Writing – original draft. **Hailin Wang:** Methodology, Software, Validation, Writing – original draft. **Haizhou Wang:** Conceptualization, Supervision, Writing – review & editing.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61802271, 61802270, 81602935, and 81773548. In addition, this work is also partially supported by Joint Research Fund of China Ministry of Education and China Mobile Company (No. CM20200409), Sichuan University and Yibin Municipal People's Government University and City Strategic Cooperation Special Fund Project, China (No. 2020CDYB-29), Science and Technology Plan Transfer Payment Project of Sichuan Province, China (No. 2021ZYSF007); The Key Research and Development Program of Science and Technology Department of Sichuan Province, China (No. 2020YFS0575, No. 2021YFG0159, No. 2021KJT0012-2021YFS0067).

References

Alassad, M., Spann, B., & Agarwal, N. (2021). Combining advanced computational social science and graph theoretic techniques to reveal adversarial information operations. *Information Processing & Management*, *58*(1), Article 102385, 10.1016/j.ipm.2020.102385.

Aoki, T., Sasano, R., Takamura, H., & Okumura, M. (2017). Distinguishing Japanese non-standard usages from standard ones. In *Proceedings of the 14th Conference on empirical methods in natural language processing* (pp. 2323–2328). Copenhagen, Denmark: http://dx.doi.org/10.18653/v1/D17-1246.

Artetxe, M., Labaka, G., & Agirre, E. (2018). 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* (pp. 789–798). Melbourne, Australia: http://dx.doi.org/10.18653/v1/P18-1073.

Boukerche, A., Zheng, L., & Alfandi, O. (2020). Outlier detection: Methods, models, and classification. *ACM Computing Surveys*, *53*(3), 1–37. http://dx.doi.org/10.1145/3381028.

Dasgupta, S., Piplai, A., Kotal, A., Joshi, A., et al. (2020). A comparative study of deep learning based named entity recognition algorithms for cybersecurity. In *4th International workshop on big data analytics for cyber intelligence and defense, IEEE International conference on big data*. Virtual event: http://dx.doi.org/10.1109/BigData50022.2020.9378482.

Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., & Yu, P. S. (2020). Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International conference on information & knowledge management* (pp. 315–324). Virtual event: http://dx.doi.org/10.1145/3340531.3411903.

Fan, Y., Ye, Y., Peng, Q., Zhang, J., Zhang, Y., Xiao, X., et al. (2020). Metagraph aggregated heterogeneous graph neural network for illicit traded product identification in underground market. In *Proceedings of the 20th IEEE International conference on data mining* (pp. 132–141). Virtual event: http://dx.doi.org/10.1109/ICDM50108.2020.00022.

Farrell, T., Araque, O., Fernandez, M., & Alani, H. (2020). On the use of jargon and word embeddings to explore subculture within the reddit's manosphere. In *12th ACM Conference on web science* (pp. 221–230). Virtual event: http://dx.doi.org/10.1145/3394231.3397912.

Gupta, M., Gao, J., Aggarwal, C. C., & Han, J. (2013). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *26*(9), 2250–2267. http://dx.doi.org/10.1109/TKDE.2013.184.

Haasio, A., Harviainen, J. T., & Savolainen, R. (2020). Information needs of drug users on a local dark web marketplace. *Information Processing & Management*, *57*(2), Article 102080. http://dx.doi.org/10.1016/j.ipm.2019.102080.

Hada, T., Sei, Y., Tahara, Y., & Ohsuga, A. (2020). 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). Southend, UK: http://dx.doi.org/10.1109/iCCECE49321.2020.9231109.

Hoseini, M., Melo, P., Júnior, M., Benevenuto, F., Chandrasekaran, B., Feldmann, A., et al. (2020). Demystifying the messaging platforms' ecosystem through the lens of Twitter. In *Proceedings of the 20th ACM internet measurement conference* (pp. 345–359). Virtual event: http://dx.doi.org/10.1145/3419394.3423651.

Huang, S.-Y., & Ban, T. (2020). Monitoring social media for vulnerability-threat prediction and topic analysis. In *Proceedings of the 19th International conference on trust, security and privacy in computing and communications* (pp. 1771–1776). Virtual event: http://dx.doi.org/10.1109/TrustCom50675.2020.00243.

Kumar, R., Yadav, S., Daniulaityte, R., Lamy, F., Thirunarayan, K., Lokala, U., et al. (2020). Edarkfind: Unsupervised multi-view learning for sybil account detection. In *Proceedings of the 29th International world wide web conference* (pp. 1955–1965). Taipei: http://dx.doi.org/10.1145/3366423.3380263.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International conference on machine learning, vol. 32 no. 2* (pp. 1188–1196). Bejing, China.

Lee, S., Yoon, C., Kang, H., Kim, Y., Kim, Y., Han, D., et al. (2019). Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web. In *Proceedings of the 26th Network and distributed system security symposium* (pp. 1–15). San Diego, USA: http://dx.doi.org/10.14722/ndss.2019.23055.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, *27*, 2177–2185.

Li, Y., Cheng, J., Huang, C., Chen, Z., & Niu, W. (2021). NEDetector: Automatically extracting cybersecurity neologisms from hacker forums. *Journal of Information Security and Applications*, *58*, Article 102784. http://dx.doi.org/10.1016/j.jisa.2021.102784.

Liu, T., Ungar, L., & Sedoc, J. (2019). Unsupervised post-processing of word vectors via conceptor negation. In *Proceedings of the 33rd AAAI Conference on artificial intelligence* (pp. 6778–6785). Hawaii, USA: http://dx.doi.org/10.1609/aaai.v33i01.33016778.

Lusthaus, J. (2019). Beneath the dark web: Excavating the layers of cybercrime's underground economy. In *Proceedings of the 40th IEEE European symposium on security and privacy workshops* (pp. 474–480). Stockholm, Sweden: http://dx.doi.org/10.1109/EuroSPW.2019.00059.

Maddela, M., Xu, W., & Preoţiuc-Pietro, D. (2019). Multi-task pairwise neural ranking for hashtag segmentation. In *Proceedings of the 57th Annual meeting of the association for computational linguistics* (pp. 2538–2549). Florence, Italy: http://dx.doi.org/10.18653/v1/p19-1242.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, *26*, 3111–3119.

Morgia, M. L., Mei, A., Raponi, S., & Stefa, J. (2018). Time-zone geolocation of crowds in the dark web. In *Proceedings of the 38th IEEE International conference on distributed computing systems* (pp. 445–455). Vienna, Austria: http://dx.doi.org/10.1109/ICDCS.2018.00051.

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2019). Textual keyword extraction and summarization: State-of-the-art. *Information Processing & Management*, *56*(6), Article 102088. http://dx.doi.org/10.1016/j.ipm.2019.102088.

Niu, Y., Xie, R., Liu, Z., & Sun, M. (2017). Improved word representation learning with sememes. In *Proceedings of the 55th Annual meeting of the association for computational linguistics, vol. 1* (pp. 2049–2058). Vancouver, Canada: http://dx.doi.org/10.18653/v1/P17-1187.

Nobari, A. D., Reshadatmand, N., & Neshati, M. (2017). Analysis of telegram, an instant messaging service. In *Proceedings of the 26th ACM on Conference on information and knowledge management* (pp. 2035–2038). Singapore: http://dx.doi.org/10.1145/3132847.3133132.

Pastrana, S., Hutchings, A., Caines, A., & Buttery, P. (2018). Characterizing eve: Analysing cybercrime actors in a large underground forum. In *The 21st International symposium on research in attacks, intrusions, and defenses* (pp. 207–227). Heraklion, Greece: http://dx.doi.org/10.1007/978-3-030-00470-5_10.

Pastrana, S., Hutchings, A., Thomas, D., & Tapiador, J. (2019). Measuring ewhoring. In *Proceedings of the 19th Internet measurement conference* (pp. 463–477). Amsterdam, Netherlands: http://dx.doi.org/10.1145/3355369.3355597.

Pastrana, S., Thomas, D. R., Hutchings, A., & Clayton, R. (2018). Crimebb: Enabling cybercrime research on underground forums at scale. In *Proceedings of the 27th International world wide web conference* (pp. 1845–1854). Lyon, France: http://dx.doi.org/10.1145/3178876.3186178.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. In *Proceedings of the 16th Conference of the north american chapter of the association for computational linguistics: human language technologies, vol. 1* (pp. 2227–2237). New Orleans, Louisiana, USA.

Peters, M. E., Neumann, M., Zettlemoyer, L., & Yih, W.-t. (2018). Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1499–1509). Brussels, Belgium: http://dx.doi.org/10.18653/v1/D18-1179.

Portnoff, R. S., Afroz, S., Durrett, G., Kummerfeld, J. K., Berg-Kirkpatrick, T., McCoy, D., et al. (2017). Tools for automated analysis of cybercriminal markets. In *Proceedings of the 26th International conference on world wide web* (pp. 657–666). Perth, Australia: http://dx.doi.org/10.1145/3038912.3052600.

Qian, C., Feng, F., Wen, L., & Chua, T.-S. (2021). Conceptualized and contextualized Gaussian embedding. In *Proceedings of the 35th Conference on artificial intelligence, vol. 35 no. 15* (pp. 13683–13691). Virtual event.

Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). 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* (pp. 99–110). Alencia, Spain.

Reid, M., Marrese-Taylor, E., & Matsuo, Y. (2020). 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* (pp. 6331–6344). Punta Cana, Dominican: http://dx.doi.org/10.18653/v1/2020.emnlp-main.513.

Samtani, S., Zhu, H., & Chen, H. (2020). Proactively identifying emerging hacker threats from the dark web: A diachronic graph embedding framework (D-GEF). *ACM Transactions on Privacy and Security*, *23*(4), 1–33. http://dx.doi.org/10.1145/3409289.

Sasano, R., & Korhonen, A. (2020). Investigating word-class distributions in word vector spaces. In *Proceedings of the 58th Annual meeting of the association for computational linguistics* (pp. 3657–3666). Virtual event: http://dx.doi.org/10.18653/v1/2020.acl-main.337.

Spinde, T., Rudnitckaia, L., Mitrović, J., Hamborg, F., Granitzer, M., Gipp, B., et al. (2021). Automated identification of bias inducing words in news articles using linguistic and context-oriented features. *Information Processing & Management*, *58*(3), Article 102505. http://dx.doi.org/10.1016/j.ipm.2021.102505.

Sutikno, T., Handayani, L., Stiawan, D., Riyadi, M. A., & Subroto, I. M. I. (2016). WhatsApp, Viber and Telegram: Which is the best for instant messaging? *International Journal of Electrical & Computer Engineering*, *6*(3), 2088–8708. http://dx.doi.org/10.11591/ijece.v6i3.10271.

Tamaazousti, Y., Le Borgne, H., Hudelot, C., Seddik, M. E. A., & Tamaazousti, M. (2020). Learning more universal representations for transfer-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(9), 2212–2224. http://dx.doi.org/10.1109/TPAMI.2019.2913857.

Tayebi, M. A., Ester, M., Glässer, U., & Brantingham, P. L. (2014). Spatially embedded co-offence prediction using supervised learning. In *Proceedings of the 20th ACM SIGKDD International conference on knowledge discovery and data mining* (pp. 1789–1798). New York, USA: http://dx.doi.org/10.1145/2623330.2623353.

Thomas, K., McCoy, D., Grier, C., Kolcz, A., & Paxson, V. (2013). Trafficking fraudulent accounts: The role of the underground market in Twitter spam and abuse. In *Proceedings of the 22nd USENIX security symposium* (pp. 195–210). Washington D.C., USA.

Wang, H., Hou, Y., & Wang, H. (2021). 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). Athens, Greece: IEEE, http://dx.doi.org/10.1109/ICCCN52240.2021.9522221.

Wegberg, R. v., Miedema, F., Akyazi, U., Noroozian, A., Klievink, B., & van Eeten, M. (2020). 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* (pp. 816–826). Taipei: http://dx.doi.org/10.1145/3366423.3380162.

Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information Sciences*, *307*, 39–52. http://dx.doi.org/10.1016/j.ins.2015.02.024.

Yang, H., Ma, X., Du, K., Li, Z., Duan, H., Su, X., et al. (2017). 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). San Jose, USA: http://dx.doi.org/10.1109/SP.2017.11.

Yuan, K., Lu, H., Liao, X., & Wang, X. (2018). Reading thieves' cant: Automatically identifying and understanding dark jargons from cybercrime marketplaces. In *Proceedings of the 27th USENIX Security symposium* (pp. 1027–1041). Baltimore, USA.

Zhang, Y., Fan, Y., Song, W., Hou, S., Ye, Y., Li, X., et al. (2019). Your style your identity: Leveraging writing and photography styles for drug trafficker identification in darknet markets over attributed heterogeneous information network. In *Proceedings of the 28th International conference on world wide web* (pp. 3448–3454). San Francisco, USA: http://dx.doi.org/10.1145/3308558.3313537.

Zhang, Y., Fan, Y., Ye, Y., Zhao, L., & Shi, C. (2019). Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *Proceedings of the 28th ACM International conference on information and knowledge management* (pp. 549–558). Beijing, China: http://dx.doi.org/10.1145/3357384.3357876.

Zhang, Y., Qian, Y., Fan, Y., Ye, Y., Li, X., Xiong, Q., et al. (2020). Dstyle-GAN: Generative adversarial network based on writing and photography styles for drug identification in darknet markets. In *Proceedings of the 36th Annual computer security applications conference* (pp. 669–680). Virtual event: http://dx.doi.org/10.1145/3427228.3427603.

Zhang, B., Xiong, D., & Su, J. (2018). Neural machine translation with deep attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(1), 154–163. http://dx.doi.org/10.1109/TPAMI.2018.2876404.

Zhao, J., Liu, X., Yan, Q., Li, B., Shao, M., Peng, H., et al. (2021). Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning. *Computers & Security*, *102*, Article 102152. http://dx.doi.org/10.1016/j.cose.2020.102152.

Zhao, K., Zhang, Y., Xing, C., Li, W., & Chen, H. (2016). Chinese underground market jargon analysis based on unsupervised learning. In *Proceedings of the 14th IEEE Conference on intelligence and security informatics* (pp. 97–102). Tucson, USA: http://dx.doi.org/10.1109/ISI.2016.7745450.

Zheng, J., Cai, F., Chen, H., & de Rijke, M. (2020). Pre-train, interact, fine-tune: A novel interaction representation for text classification. *Information Processing & Management*, *57*(6), Article 102215. http://dx.doi.org/10.1016/j.ipm.2020.102215.

Zhu, W., Gong, H., Bansal, R., Weinberg, Z., Christin, N., Fanti, G., et al. (2021). Self-supervised euphemism detection and identification for content moderation. In *Proceedings of the 43rd IEEE Symposium on security and privacy* (pp. 229–246). Virtual Event: http://dx.doi.org/10.1109/SP40001.2021.00075.