Annals of Emerging Technologies in Computing (AETiC)
Vol. 5, No. 2, 2021

Research Article

Codeword Detection, Focusing on Differences in Similar Words Between Two Corpora of Microblogs

Takuro Hada[1,*], Yuichi Sei[1,2], Yasuyuki Tahara[1] and Akihiko Ohsuga[1]

[1]The University of Electro-Communications, Tokyo, Japan
hada.takuro@ohsuga.lab.uec.ac.jp; tahara@uec.ac.jp; ohsuga@uec.ac.jp
[2]JST PRESTO, Saitama, Japan
seiuny@uec.ac.jp
*Correspondence: hada.takuro@ohsuga.lab.uec.ac.jp

Received: 29th January 2021; Accepted: 26th February 2021; Published: 1st April 2021

Abstract: Recently, the use of microblogs in drug trafficking has surged and become a social problem. A common method applied by cyber patrols to repress crimes, such as drug trafficking, involves searching for crime-related keywords. However, criminals who post crime-inducing messages maximally exploit "codewords" rather than keywords, such as enjo kosai, marijuana, and methamphetamine, to camouflage their criminal intentions. Research suggests that these codewords change once they gain popularity; thus, effective codeword detection requires significant effort to keep track of the latest codewords. In this study, we focused on the appearance of codewords and those likely to be included in incriminating posts to detect codewords with a high likelihood of inclusion in incriminating posts. We proposed new methods for detecting codewords based on differences in word usage and conducted experiments on concealed-word detection to evaluate the effectiveness of the method. The results showed that the proposed method could detect concealed words other than those in the initial list and to a better degree than the baseline methods. These findings demonstrated the ability of the proposed method to rapidly and automatically detect codewords that change over time and blog posts that instigate crimes, thereby potentially reducing the burden of continuous codeword surveillance.

Keywords: ***Codewords Detect; Microblog; Twitter; Word Embedding***

1. Introduction

Recently, numerous incidents have been reported related to enjo kosai (subsidized companionship) and illegal drugs promoted using microblogs. According to a previous study, enjo kosai, "A is the label used for young women who agree to meet strange men for dates, which might involve coitus in exchange for money or gifts" [1]. Particularly, enjo kosai for young women under 18 years has become a problem. The posters of enjo kosai and illegal-drug-related activities are wary of their posts and accounts being removed from social networking services by cyber patrols or of being arrested by the police. Thus, only those knowledgeable about the codewords carry out illegal transactions (Fig. 1).

For example, the word "ganja" is a popular codeword for marijuana, whereas the words "speed" and "meth" are mainly used for methamphetamine. Limited success has been achieved in the generation of keyword lists and the implementation of countermeasures for their detection because of the frequency with which they are changed to elude surveillance [2]. For example, for marijuana, the words "grass," "weed," and "joint" have previously been used. Similarly, for methamphetamine, the words "ice" and "crystal" have previously been used. Thus, cyber patrols need to continuously

Takuro Hada, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga, "Codeword Detection, Focusing on Differences in Similar Words Between Two Corpora of Microblogs," *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 90-102, Vol. 5, No. 2, 1st April 2021, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2021.02.008, Available: http://aetic.theiaer.org/archive/v5/v5n2/p8.html.

track new cryptograms and the possible addition of words to these cryptograms, which increases the surveillance burden. Hence, to support the prevention of crimes, such as drug trafficking and enjo kosai, on microblogging sites (particularly Twitter), we developed a method to detect crime-related tweets containing codewords. Previous studies on codeword usage on the Internet, such as in bulletin boards, have been published; however, a few studies have targeted short sentences, such as microblogs that involve a limited number of characters in a single post, which leads to limited understanding of the sentence meaning. This suggests that the discovery of codewords in such sentences would be significant for crime prevention because of early detection. In this study, we focused on the differences in usage of the same words between two corpora based on the idea that similar words appear interspersed with others in malicious communications. This method allows the detection of codewords likely included in crime-related tweets and among words likely to appear with them.

Here, we focused on Twitter and classified tweets related to codewords into four types:

1. Tweets that feature only known codewords (and words directly related to crime).
2. Tweets containing only unknown codewords.
3. Tweets featuring a mixture of known codewords (and words directly related to crime) and unknown codewords.
4. Tweets that feature neither known nor unknown codewords.

Moreover, we aim to detect unknown codewords based on the known codewords (and words directly related to crime), assuming that the tweets in (3) exist.

Specifically, we use a corpus of tweets targeting (1) and (3) above (hereinafter referred to as the "Bad Corpus") and a corpus of tweets targeting (4) above (hereinafter referred to as the "Good Corpus").

Furthermore, this method builds a word–distribution–expression model using Word2vec [3] for each of the two corpora and detects codewords based on the differences between the words appearing higher in cosine similarity concerning similar words.

(a) Writing in Japanese (Example). **(b) Translation of (a).**

Figure 1. Example sentences with codewords from Twitter.

2. Background

2.1. Increase in the number of crimes involving drug trafficking and enjo kosai

A news article based on a United Nations Office on Drugs and Crime report noted an increase in online drug trafficking via Facebook, Twitter, and Instagram [4].

Fig. 2 shows the number of arrests for marijuana offenses by age group in Japan, and evidently, the number of arrests has been increasing annually, particularly among teenagers and young adults in their 20s.

Additionally, according to the National Police Agency (Japan) data on enjo kosai, the number of smartphone victims of crimes and the number of children who fall victim to crimes originating from social networking sites (SNSs) increase annually, with the latter reaching an all-time high in 2019 (Fig. 3). Moreover, among SNSs, Facebook and Twitter are commonly used [6], and Twitter reportedly has the highest number of child victims (40%) (Fig. 4). Thus, we focused on Twitter.

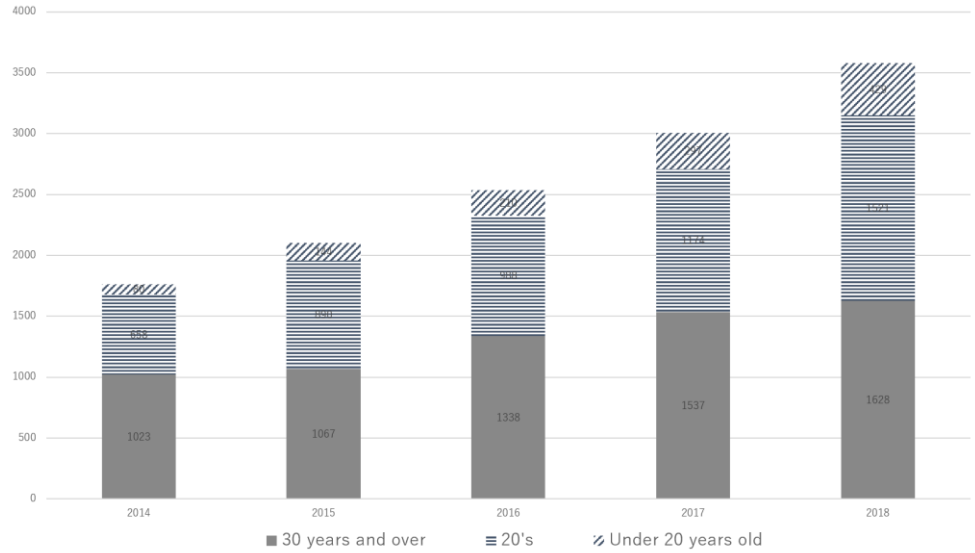

Figure 2. Changes in the number of marijuana vending arrests by ages (according to the data from the National Police Agency (Japan)) [5]

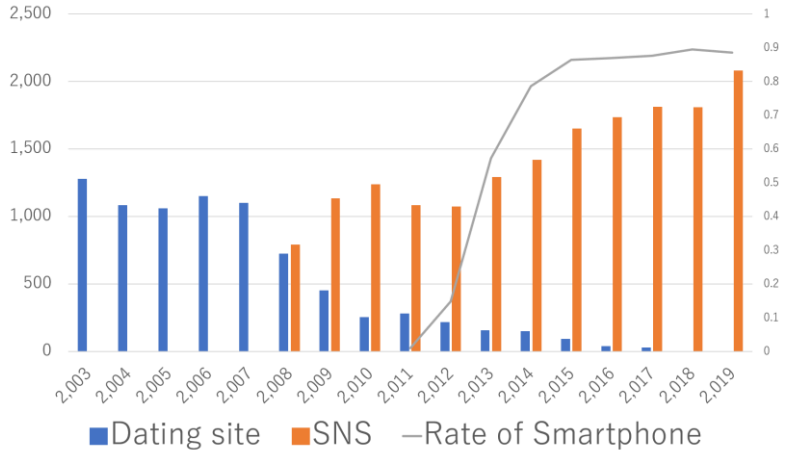

Figure 3. Number of children victimized via SNSs according to the data from the National Police Agency (JAPAN) [7] [8] and an undisclosed dating site (2018 and 2019)

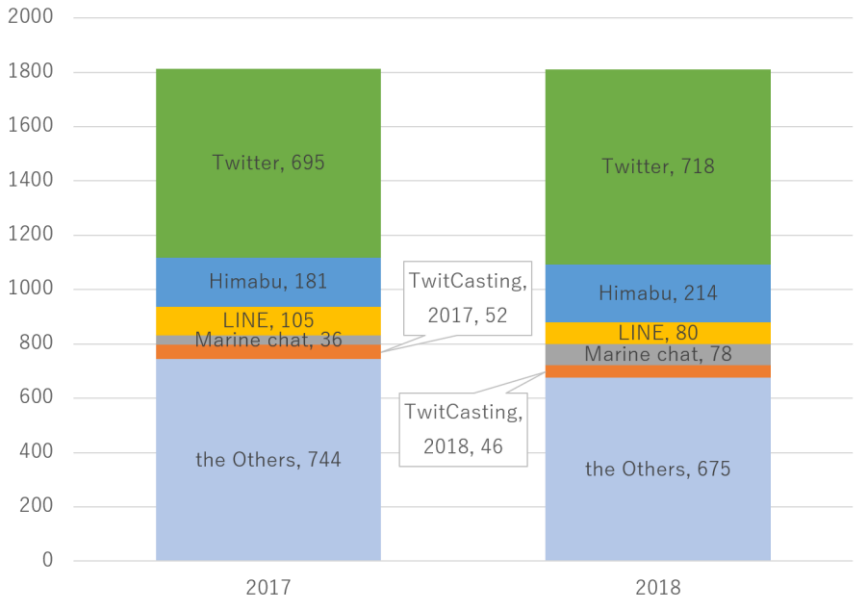

Figure 4. Sites with multiple child victims, according to data from the National Police Agency (Japan)) [9]

2.2. Codewords

Codewords are used in various industries, and various words are often applied to convey the message. In this study, we define codewords as those used in illegal transactions to elude police surveillance. Different words with similar meanings can be used as codewords. For example, Yuan *et al*. [2] described "Dark Jargon" as a codeword. We targeted the following codeword types.

1. The name of the target itself, which is a criminal act.

a) Widely known words, e.g., "LSD" "marijuana." Since these words are generally recognized and considered ineffective as codewords for transactions, we classify them as "related words," not "codewords," as described in Section 4.4 "Annotation" below.

b) Minimally known words. Minimally known words have the same effect as a codeword even when used for extended periods without replacement; this is because the target word is generally not recognized, and only certain people can understand it even if used as it is. For example, in Japan, the words "white kush" and "white widow" are not well known as a type of marijuana; therefore, we included this category because such words are occasionally used as trading words.

2. It is not the target name itself that is a criminal act.

a) Diversion (camouflage). This class includes words used to camouflage commonly used words by giving them a cryptic meaning. For example, there are codewords for marijuana, "grass," and codewords for methamphetamine, "ice," and "crystal," in the context of drug trafficking. Conversely, as a codeword related to enjo kosai, "Yukichi" is used as a codeword referring to a unit of 10,000 yen from "Fukuzawa Yukichi," the name of the person in the 10,000-yen portrait.

b) Coined words. Words intentionally created to conduct illegal transactions fall under this category. For example, for marijuana, "Mary Jane" and "420" are used as codewords related to drug trafficking, and "shabu" and "gan−koro" are used for methamphetamine in Japan. Contrarily, there are some codewords related to enjo kosai, such as "kami-matchi (God waiting)." Some of the words are unfamiliar to us; however, they have similar sounds or can be associated with Chinese characters. For example, "Enko" and "En" are codewords related to enjo kosai in Japan.

2.3. Research of codewords

In this study, we focused on codewords related to drug trafficking and enjo kosai. Although such codewords from websites have been previously analyzed [10], the findings do not apply to microblogging sites, such as Twitter. De la Rosa *et al*. [11] described the following features of microblogs:

- Short character length. Microblogs comprise as little as a single word to less than a paragraph at most. For Twitter, there is a limit of 140 characters per post.
- Informal and unstructured formats. Microblogs contain slang, misspellings, and abbreviations.

Thus, it is difficult to maintain up-to-date dictionaries using prepared matching methods, given the frequency of codeword changes and considering slang use and misspellings. Furthermore, machine learning methods for detecting codewords are difficult to apply because of the limited length of the sentences and word presentation, which eliminates context. However, the analysis of tweets featuring codewords indicated multiple cases of similar word appearance, suggesting that codewords might be detected if word-embedding representation can be used to vectorize the words and identify similar words in their vicinity. Here, we propose a new method for detecting unknown words by focusing on the similarity of known words.

3. Related Work

3.1. Related works on codeword detection

Several studies have attempted to resolve the issue of the increasing number of crimes instigated on Twitter [12], [13]. These studies included the detection of offensive and malicious words [14], [15],

[16]. In criminal exchanges, codewords are sometimes used for transactions and cleverly hidden among common words to avoid their detection. Studies have been undertaken to detect such codewords. For example, on the dark web, cannabis is exchanged using the names "popcorn" and "blueberries" and child pornography is referred to as "cheese pizza." Yuan *et al*. [2] proposed a method for automatically identifying jargon from the dark web. Since it is impossible to identify such jargon in a single corpus using Word2vec, they prepared multiple corpora and detected jargon according to a semantic contradiction of terms appearing in two different corpora. However, this research did not cover tweets, which are short and contain slangs and misspellings. Zhao *et al*. [17] focused on the jargons used in cybercrime associated with the underground market in China and used unsupervised learning for their detection. They concluded that "CBOW + NS" was the optimal setting for Word2vec, resulting in 20% higher detection rates than those by the "LDA" approach. However, these methods represent first-stage research [2]. Furthermore, Aoki *et al*. [18] detected nonstandard word usage involving definitions that differed from their original meaning. These words were not limited to use in crime-related contexts, and it is conceivable that crime-related codewords function with other methods to conceal a given message. As an effort to detect crime-related codewords in Japanese, Hada *et al* [19] Focused on the difference in similar words between two corpora, and are working on codeword detection. However, there is room for improvement such as improvement of accuracy and expansion of corpus scale.

3.2. Word embedding

A word embedding is a vector representation of semantic information of words.

There are, for example, a thesaurus-based method and a count-based method for acquiring word embedding.

Count-based methods can also be used to obtain word similarities; however, they are more dependent on the corpus than on the direct handling of co-occurrence counts.

Thus, we decided to use an inference-based method that treats surrounding words probabilistically, i.e., Word2vec, which is famous for its high accuracy.

In Word2vec, neural networks are used for vectorization, and there are two types: CBoW (continuous bag-of-words) and skip-gram (continuous skip-gram model).

This study applies the skip-gram model, which is more accurate and widely used than CBoW. Skip-gram is a simple neural network consisting of three layers (input, middle, and output) that predict the context (output) using the words (input) (Fig. 5).

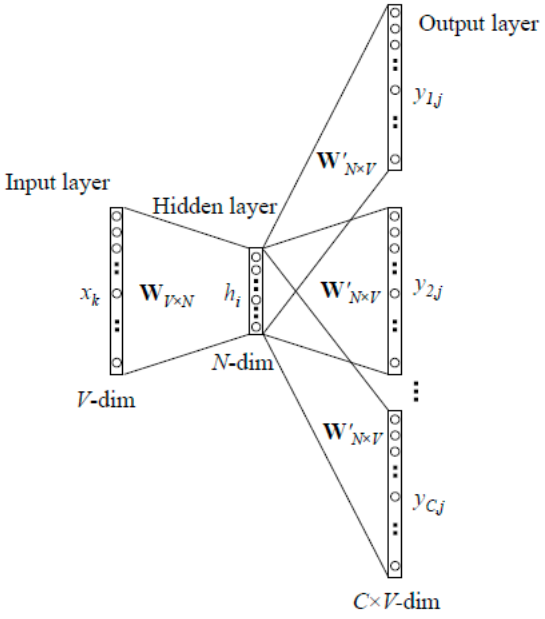

Figure 5. The skip-gram model quoted from [20] Fig. 3.

4. Approach

4.1. Core idea

Those who post crime-related codewords to camouflage their criminal intent tend to be clever; however, there are few differences in the contextual nature of back-and-forth exchanges. We hypothesized that words used in illegal negotiations are used in the same sense as their analogs. Thus, we speculated that unknown codewords might appear as words similar to known codewords in a codeword corpus. Using data obtained from Twitter, we prepared a set of tweets focused on illegal trading and divided them into two groups based on malicious intent: The Bad Corpus is a collection of tweets containing one or more words in the word list, whereas the Good Corpus is a collection of all tweets not included in the Bad Corpus. Subsequently, we performed word-distribution analysis on each corpus [3] and calculated the cosine similarity using gensim [21]. We defined a word with a high cosine similarity as W and referred to a set of such words as "Similar words" of W. For example, the word "paper" is a codeword for "LSD," a type of methamphetamine. Similar words in each corpus are shown in Table 1. As Table 1 shows, for "paper," the detection of similar words in the two corpora resulted in different results. Moreover, we found that six of the top 10 similar words in the Bad Corpus were codewords. To discover unknown codewords, we focused mainly on two points: 1) similar words, W, in the Good Corpus differ from those in the Bad Corpus, and 2) searches for similar words in the Bad Corpus result in similar metonymy and related maliciousness. We selected a list of codewords related to drug trafficking and enjo kosai.

Table 1. Top 10 words similar to "paper" in each corpus (Bold words are codewords).

Rank	Good Corpus		Bad Corpus	
	Result	Meaning	Result	Meaning
1	字詰め	Letter-writing	業販	Commercial sales
2	試筆	Test brush	市内	Within the city
3	便箋	Letterhead	営業中	Open for business
4	裏紙	Backing paper	メニュー	Menu
5	ハードカバー	Hardcover	スカンク	Skunk
6	アルシュ	Archetypal	リキッド	Liquid
7	用紙	Paper	ノーザン	Northern
8	断裁	Cutting	グミ	Gummi
9	模造紙	Imitation paper	ハイレギュラー	High regular
10	方眼	Graph paper	ヘイズ	Hayes

4.2. Procedure

The outline of the system is shown in Fig. 6, Fig.7.

The detailed flow of the method is illustrated in Algorithms 1 and 2.

1) For each word in the word list, calculate the score for each of the two corpora (Function SIMILAR).

a) For each word, similar words up to the top N of the cosine similarity are searched using the pre-constructed word distribution expression model (Good_Corpus, Bad_Corpus) (Get_similar_words). In this experiment, we set N as 20.

b) Match the retrieved N similar words individually against the matching list (Codeword_List).

c) If a match is found in the codeword list, add points (X = X + 1) to a maximum of 20 points.

d) If the word, W, does not match any in the codeword list, it is possible that the word is not registered as a codeword; thus, similar words up to the N/2 highest cosine similarity to the word W are considered. If the score is greater than or equal to the threshold, points are added to X (X = X + 1).

e) For each similar word to W, if the word does not match any in the hidden word list, N/4 similar words are searched based on the similar word, and word W is evaluated.

2) Calculate the difference (Diff) between the calculated Good (Cnt_Good) and Bad (Cnt_Bad) point totals.
3) If the Bad point total exceeds the threshold, it is identified as a codeword. If the Diff is above a certain level, the threshold value for the Bad point total decreases.

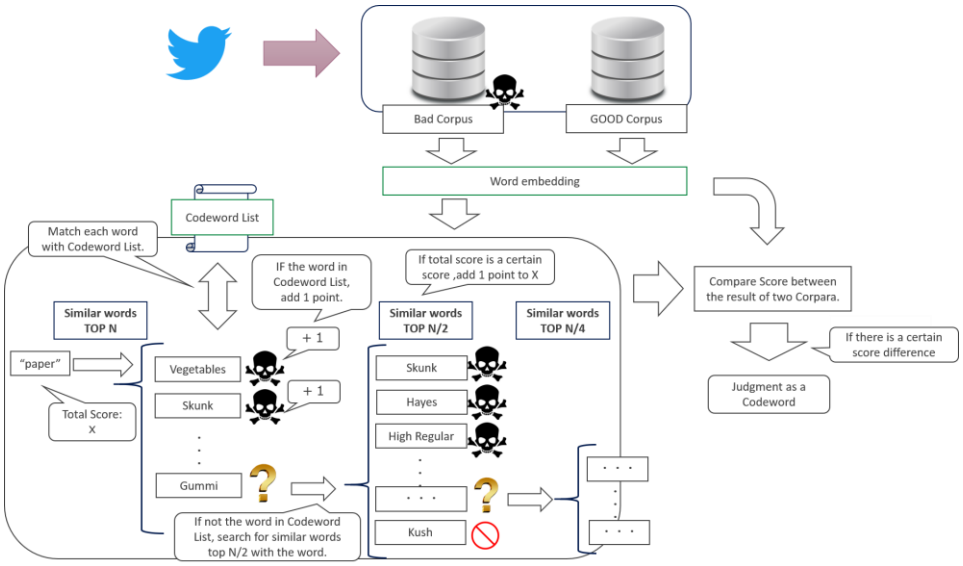

Figure 6. Schematic of the system

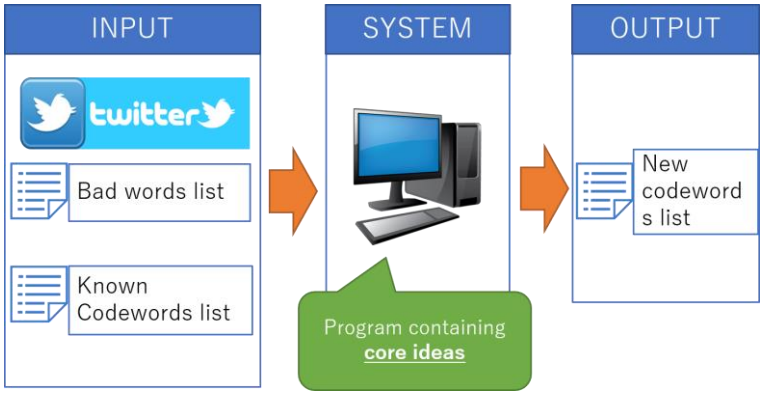

Figure 7. Input/output of the system

Algorithm 1. Main Program

```
1: Input: Word List, N, Good_Corpus, Bad_Corpus
2: Output: Codewords
3: for all Word in Word List do
4: Cnt_Bad ⇐ SIMILAR(Word, N, Bad_Copus, 1)
5: Cnt_Good ⇐ SIMILAR(Word, N, Good_Copus, 1)
6: Diff ⇐ abs(Cnt_Bad – Cnt_Good)
7: if (Cnt_Bad/N >= 0.2) or ((Diff/N >= 0.15) and
8: (Cnt_Bad/N >= 0.1)) then
9: Codeword List.append(WORD)
10: end if
11: end for
12: return(Codeword List)
```

Algorithm 2. Function SIMILAR

```
1: Input: Word, N, Corpus, Loop_count
2: Output: Number of matches with codewords
3: X ⇐ 0
4: Sim_words ⇐ Corpus.Get_similar_words(Word, N)
5: for all Sim_word in Sim_words do
6: if Sim word in Codeword_List then
7: X ⇐ X + 1
8: else if Loop_count <= 2 then
```

```
9: Y ⇐ SIMILAR(Sim_word, N/2, Corpus, Loop_count+1)
10: if Y/N >= 0.2 then
11: X ⇐ X + 1
12: end if
13: end if
14: end for
15: return(X)
```

4.3. Introduction of the "whitelist" function

Besides the core idea, we considered that there are two major ways to further improve the accuracy of codeword detection:

1. Reduce the number of irrelevant words detected
2. Increase the number of detected codewords

Of these, we considered "1" in this study. We attempted to improve precision by reducing the number of irrelevant words detected. Specifically, to avoid the detection of unnecessary words, we examined a method of determining the irrelevance of words. Specifically, we assumed that if the same word is used similarly in the two corpora, it is not specifically used for malicious purposes, i.e., it is not a cryptic word. Therefore, we searched the top N similar words of the same word in both corpora, and if a certain percentage of the words in both corpora were common words, we judged that the word was used for similar purposes in both corpora and was unlikely to be used as a cryptic word. In this case, we exclude them from the list of cryptic words. We added these words to the list of irrelevant words, i.e., the white list, and introduced a mechanism to reduce the matching score in Algorithm 2 when these words appear in the similarity search.

4.4. Annotation

The word list extracted from the corpora models comprised 950 nouns, which were classified into the following three categories based on the tweets of two to three people with no prior knowledge of the words.

- Codewords. Words judged to have a meaning different from their original meaning.
- Related words. Although these words could not be categorized as codewords, they tended to appear alongside codewords and were judged as rarely appearing in general tweets (e.g., "stock" and "price").
- Unrelated words. Words that do not meet the criteria of the previous two categories.

5. Experiment

5.1. Summary of the experiment

We performed an experiment to detect codewords using 950 pre-annotated words and included 10 of 45 known codewords among the 950 words in a known-codeword list for matching. Thereafter, the system evaluated the similarity of words to codewords to identify the remaining 35 words. The experiment was performed using the following steps.

5.2. Experiment process

5.2.1. Data collection

Twitter data (47 days, 5.4 GB) were collected using the Twitter API, and only the text was used for analysis. The following words not considered to be related to the pre-processed codewords were removed before analysis.

- Single-byte alphanumeric characters,
- URLs,
- Full-width symbols,
- Line-feed characters,
- Words frequently appearing on Twitter (e.g., "RT," "Favorite," etc.).

5.2.2. Creating the corpora

Creating the corpora: The pre-processed Twitter data were analyzed to identify 10 words (words judged to have been posted for criminal purposes related to drug trafficking and enjo kosai) in each tweet, followed by classification into the following two corpora.

- Bad Corpus (8 MB). This represented a group of tweets containing one or more of the 10 words. We assumed that words from tweets related to illegal transactions were collected in this corpus.
- Good Corpus (4 GB). This represented a group of tweets not including words from the Bad Corpus. We assumed that most of these tweets were general interactions.

5.2.3. Morphological analysis

We focused on Twitter because of its use of short sentences, new words and slang, and limited character length. This suggested that some sentences might not be correctly separated. Moreover, the Japanese language has a unique sentence structure not separated by spaces; thus, segmentation was necessary before word distribution. We segmented sentences from Twitter using Sudachi [22] for the following reasons:

- Sudachi is continuously improved and maintained, its dictionary is regularly updated, and it is expected to have the most up-to-date word list,
- A word division unit can be selected.

5.2.4. Word embedding

After split writing, word distribution was performed using Word2vec with the parameters shown in Table 2.

Table 2. Parameters of Word2vec.

Parameter	Value
Size	200
Min_count	20
Window size	5
Skip-Gram or CBow	Skip-Gram[23]

5.2.5. System execution

We created a word list from the corpus model generated by word distribution and extracted 940 nouns common between the two corpora models and 10 nouns from the Bad Corpus model. Afterward, we executed the method using this word list.

5.3. Comparative approach

To perform comparative analysis, we prepared a baseline method in which all nouns in the tweets containing words used in a malicious exchange were defined as codewords. Fig. 8 shows the results of the comparison of the proposed method with the baseline method.

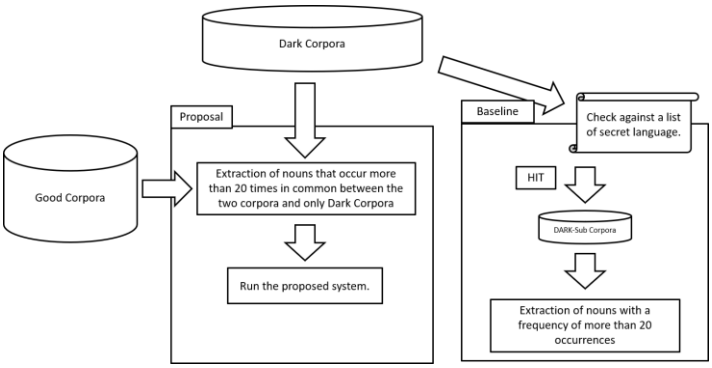

Figure 8. Relationship between the proposed and baseline methods

The baseline method for detecting codewords is described as follows:

- The Bad Corpus was used to analyze the codeword list used in the proposed method.
- All sentences containing words from the hidden word list were extracted, and a Bad subcorpus was created.

- From the Bad subcorpus, only nouns were extracted, with these considered codewords.

5.4. Measure of accuracy

The evaluation was conducted using the following four indices: true positive (TP), false positive (FP), false negative (FN), and true negative (TN).

5.4.1. Precision

The precision was calculated using the goodness-of-fit ratio, which was calculated as the percentage of data that are actually positive out of the data that are predicted to be positive:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

5.4.2. Recall

This refers to the recurrence rate, and it is calculated as the percentage of predicted positive values out of the actual positive ones:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

5.4.3. Accuracy

Accuracy is the percentage of data predicted to be positive or negative that is actually the case:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \quad (3)$$

5.4.4. F-score

The F-score is defined as the weighted harmonic mean of the precision and recall:

$$F - Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (4)$$

5.5. Results

The annotation resulted in 45 codewords, of which 10 were prepared as a known-codeword list. The system was executed using 940 words that excluded these 10 words. Thirty-nine words were detected as predicted codewords, with 17 of these identified as true codewords. Table 3 shows the results of the proposed method and the baseline method. Table 4 shows that the proposed method detected codewords at a higher rate than the baseline method; however, the number of codewords detected by the proposed method was lower than that by the baseline method. Furthermore, we determined four indicators of the method performance (precision, recall, accuracy, and F-score) (Table 4).

Table 3. Evaluation Results.

Classified	All words		Proposed method		Baseline method	
	Quantity	Rate	Quantity	Rate	Quantity	Rate
Codewords	35	3.7%	17	43.6%	23	5.7%
Others	905	96.3%	22	56.4%	379	94.3%
SUM	940		39		402	

Table 4. Details of the Results.

Evaluation method	Proposed	Baseline	Difference
Precision	0.436	0.057	0.379
Recall	0.486	0.657	−0.171
Accuracy	0.957	0.584	0.373
F-score	0.459	0.105	0.354

Table 5 shows that the proposed method returned better results in terms of precision, accuracy, and F-score relative to the baseline method. Moreover, the proposed method detected words, such as "diesel," "skunk," "gummi," "lemon," and "joint."

Table 5. Words Similar to "Ice."

Rank	Result	Meaning	Annotation
1	市内	Within the City	△
2	郵送	Posts	△
3	営業中	In Business	△

4	野菜	Vegetable	○
5	極上	The Best	△
6	業販	Commercial Sales	△
7	ブラック	Black	○
8	おはようございます	Good Morning	×
9	メニュー	Menu	△
10	テレ	Tele	△

6. Discussion

6.1. Detection challenges

In this study, we developed a method to detect known codewords from short sentences, such as those used in tweets on Twitter. Although we anticipated a lower recall in the proposed method relative to that in the baseline method, which uses a wide range of codewords, we observed minimal differences between the two results; however, the proposed method showed higher precision, accuracy, and F-score than those of the baseline method. These results indicated that the proposed method could detect codewords with higher accuracy than the baseline method. However, the proposed method could not detect "typical" codewords, such as "ice" and "vegetable." Thus, we identified similar words to "ice" in the word-distributed expression model created from the Bad Corpus (Table 5).

Table 5 shows that multiple words used for malicious communications were identified as being similar to "ice." However, the number of matching words was small because the number of words in the known-codeword list was too small. Thus, using more words in the known-codeword list might improve the recall of the proposed method. Furthermore, it was found that many related words defined in Section 4.4, such as "post" and "in business," also appeared. Therefore, expectedly, the recall can be improved by introducing a mechanism for matching related words.

6.2. Detection of related words

Table 6 shows the results from including both codewords and related words. Table 7 shows that the precision for the proposed method was high (0.718) and that multiple related words were identified, even if they were not codewords. Therefore, our future work will introduce a mechanism that can detect codewords and related words.

Table 6. Result Evaluations Involving Related Words.

Classified	**All words**		**Proposed method**	
	Quantity	Rate	Quantity	Rate
Codewords	35	3.7%	17	43.6%
Related word	119	12.7%	11	28.2%
Unrelated	786	83.6%	11	28.2%
SUM	940		39	

Table 7. Results Generated by the Inclusion of Related Words.

Evaluation method	**Result**
Precision	0.718
Recall	0.182
Accuracy	0.854
F-score	0.290

6.3. Applicability of the model to other languages

Although we used the proposed method to analyze sentences written in the Japanese language, the method is versatile enough to apply to other languages. Since the proposed method is based on the idea that bad words may appear in the same way as similar bad words. Even in languages different from Japanese, it is considered necessary to include at least three pieces of information ("the object of the transaction," "the location," and "the amount") when conducting transactions using codewords. The "transaction method" and the "quality of the transaction object" would be included

to realize a quick exchange while avoiding cyber patrols. For these reasons, we believe that the proposed method can be applied to languages different from Japanese since it is for detecting codewords from similar words.

7. Conclusions

We proposed a method that focuses on the difference in similar words among corpora to detect codewords to support cyber patrol. The proposed method compares the similarity of the same word between two corpora based on the hypothesis that the word's similarity is different between corpora categorized by the presence or absence of malicious intent. Thereafter, we conducted a codeword detection experiment using the proposed method and detected codewords other than those used for matching.

The method successfully detected codewords not included in the known-codeword list and outperformed a baseline method in terms of precision, accuracy, and F-score. These findings suggest the efficacy of this method for the automatic detection of frequently changing codewords.

In the future, we intend to improve the proposed method by implementing the following.

a) Expansion of the corpus size. In this study, we aimed at detecting codewords based on the tweets that matched the corpus classification list, targeting the known codewords in (1) and the mixed codewords in (3) among the tweets classified in Section 1. In the future, to broaden the detection range of unknown codewords, we will focus on users who are tweeting with criminal intentions, i.e., tweets categorized as (2) (tweets in which only unknown codewords are used.)

b) Expansion of detection target. By understanding codewords, it is possible to detect inappropriate postings by performing keyword searches of actual postings using the codewords as keys. However, it is necessary to scrutinize whether the postings are illegal or not from among the double-meaning codewords because it is expected that many postings with general meanings are also included.

Therefore, we aim to not only detect codewords but also to detect inappropriate posts based on the codewords.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP17H04705, JP18H03229, JP18H03340, JP18K19835, JP19K12107, JP19H04113, and JST, PRESTO Grant Number JPMJPR1934.

References

[1] L. Miller, "Those Naughty Teenage Girls: Japanese Kogals, Slang, and Media Assessments", *Journal of Linguistic Anthropology*, Vol. 14, Issue 2, Pages 225-247, December 2004.

[2] K. Yuan, H. Lu, X. Liao and X. Wang, "Reading Thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces". in *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, pp. 1027–1041, 2018, Available: https://dl.acm.org/doi/10.5555/3277203.3277280.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space. in 1st International Conference on Learning Representations", In *ICLR 2013*, Arizona, USA, May 2013.

[4] Asia-Pacific drug trade thrives amid the COVID-19 pandemic. https://www.reuters.com/article/us-asia-crime-drugs/asia-pacific-drugtradethrives-amid-the-covid-19-pandemic-idUSKBN22R0E0.

[5] Situation of organized crime in 2018, Available: https://www.npa.go.jp/sosikihanzai/kikakubunseki/sotaikikaku04/h30.sotaijousei.pdf.

[6] Jawaid A. Mangnejo, Arif R. Khuhawar, Muneer A. Kartio and Saima S. Soomro, "Inherent Flaws in Login Systems of Facebook and Twitter with Mobile Numbers", *Annals of Emerging Technologies in Computing*, Vol. 2, Issue 4, pp. 53–61, 2018, DOI: 10.33166/AETiC.2018.04.005.

[7] Juvenile Delinquency, "child abuse and sexual assault of children in 2019", Available: https://www.npa.go.jp/safetylife/syonen/hikou gyakutai sakusyu/R1.pdf.

[8] Current Situation and Measures for Children Victimized by SNS, Available: https://www8.cao.go.jp/youth/kankyou/internet torikumi/kentokai/40/ pdf/s4.pdf.

[9] Status of Children Victimized by SNS in 2018, Available: https://www8.cao.go.jp/youth/kankyou/internet torikumi/kentokai/41/pdf/s4-b.pdf.

[10] W. Lee, S. S. Lee, S. Chung and D. An, "Harmful contents classification using the harmful word filtering and SVM" in *Computational Science – ICCS 2007*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 18–25, 2007.
[11] K. Dela Rosa and J. Ellen, "Text classification methodologies applied to micro-text in military chat", pp. 710–714, 2009.
[12] D. O'Day and R. Calix, "Text message corpus: Applying natural language processing to mobile device forensics" pp. 1–6, 2013.
[13] C. Kansara, R. Gupta, S. D. Joshi and S. Patil, "Crime mitigation at twitter using big data analytics and risk modeling", in *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. pp. 1–5, 2016.
[14] G. Xiang, B. Fan, L. Wang, J. Hong and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus", pp. 1980–1984, 2012.
[15] G. Wiedemann, E. Ruppert, R. Jindal and C. Biemann. "Transfer Learning from LDA to BiLSTM-CNN for Offensive Language Detection in Twitter", *CoRR*, vol. abs/1811.02906, 2018.
[16] A. Hakimi Parizi, M. King and P. Cook, "UNBNLP at SemEval2019 task 5 and 6: Using language models to detect hate speech and offensive language", in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 514–518, 2019.
[17] K. Zhao, Y. Zhang, C. Xing, W. Li and H. Chen, "Chinese underground market jargon analysis based on unsupervised learning" in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pp. 97– 102, 2016.
[18] T. Aoki, R. Sasano, H. Takamura, and M. Okumura. (2017). Distinguishing Japanese nonstandard usages from standard ones," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2323–2328.
[19] T. Hada, Y. Sei, Y. Tahara, A. Ohsuga, "Codewords detection in microblogs focusing on differences in word use between two corpora", *International Conference on Computing, Electronics Communications Engineering (iCCECE)*, pp. 103–108, UK, 2020.
[20] Xin Rong, "Word2vec Parameter Learning Explained", *CoRR*, abs/1411.2738, 2014.
[21] R. Rˇ ehu˚rˇek and P. Sojka, "Software framework for topic modeling with large corpora", In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta: ELRA, pp. 45–50, 2010.
[22] K. Takaoka, S. Hisamoto, N. Kawahara, M. Sakamoto, Y. Uchida and Y. Matsumoto, "Sudachi: a Japanese tokenizer for business" in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan: European Language Resources Association (ELRA), 2018.
[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality" *CoRR*, Vol. abs/1310.4546, 2013.