A Multimodal Text Matching Model for Obfuscated Language Identification in Adversarial Communication*

Longtao Huang
Alibaba Group
Beijing, China
hlightening@163.com

Ting Ma
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
mating@iie.ac.cn

Junyu Lin†
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
linjunyu@iie.ac.cn

Jizhong Han
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
hanjizhong@iie.ac.cn

Songlin Hu
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
husonglin@iie.ac.cn

ABSTRACT

Obfuscated language is created to avoid censorship in adversarial communication such as sensitive information conveying, strong sentiment expression, secret actions plan, and illegal trading. The obfuscated sentences are usually generated by replacing one word with another to conceal the textual content. Intelligence and security agencies identify such adversarial messages by scanning with a watch-list of red-flagged terms. Though semantic expansion techniques are adopted, the precision and recall of the identification is limited due to the ambiguity and the unbounded creation way. To this end, this paper frames the obfuscated language identification problem as a text matching task, where each message is checked whether matches a red-flagged term. We propose a multimodal text matching model which combining textual and visual features. The proposed model extends a Bi-directional Long Short Term Memory network with a visual-level representation component to achieve the given task. Comparative experiments on real-world dataset demonstrate that the proposed method could achieve a better performance than the previous methods.

CCS CONCEPTS

• **Information systems** → *Web searching and information discovery*; • **Computing methodologies** → *Phonology / morphology*.

KEYWORDS

Obfuscated Language Identification, Text Matching, Natural Language Understanding

*This research is supported in part by the National Key Research and Development Program of China (No. 2017YFB1010000) and the National Natural Science Foundation of China (No. 61702500).
†Junyu Lin is the corresponding author.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313410

ACM Reference Format:
Longtao Huang, Ting Ma, Junyu Lin, Jizhong Han, and Songlin Hu. 2019. A Multimodal Text Matching Model for Obfuscated Language Identification in Adversarial Communication. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313410

1 INTRODUCTION

Motivation: Nowadays, online social networks have become more and more important in the diffusion of information by increasing the spread of live content and diverse opinions [2]. The development of online social networks provide convenience for people to get a very vast source of information on an unprecedented scale[9]. But at the same time, the security of cyberspace also faces enormous challenges. Some netizens have utilized online social networks to discuss sensitive information, plan for secret actions, trade illegal products, spread political rumours, or express strong sentiment. In order to detect such harmful communications, intelligence and security agencies would maintain a watch-list of red-flagged terms such as attack, bomb and heroin. The watch-list is used for keyword-spotting in online spread messages which are filtered as suspicious information [8].

However, netizens usually create and use obfuscated language to evade the censorship of intelligence and security agencies[13]. The original information is obfuscated by replacing the red-flagged term with an ordinary word or a brand new word (not a word before). Such substitution words are called *morphs* of the original red-flagged terms[11]. For example, "金太阳(Kim Sun)" is a morph generated from the original term "金日成(Kim Il-sung)" because the character "日" in the original term means "太阳(Sun)". The morph "火乍弓单" is generated from the original term "炸弹(Bomb)" by decomposing the Chinese characters. The morph "P0RN" of "PORN" is generated by substituting the letter "O" with the number "0" because they are visually similar. Such kind of obfuscated language can help to avoid the automatic detection since the watch-list of red-flagged terms is always manually maintained and impossible to cover all morphs.

Problem Statement: This paper addresses the problem of identifying obfuscated language that is created to evade censorship. The

raw input of this task is a watch-list of red-flagged terms and a set of documents to be scanned. The target is to find all documents that mention any of the red-flagged terms even the original terms are obfuscated. An intuitive solution is to expand the original terms by enumerating all possible morphs. However, morphs are created in unbounded way and evolve rapidly over time. Furthermore, some morphs are hardly identified even by human. So enumerating all morphs corresponding to the original terms is impossible. This opens an unexplored area of obfuscated language processing [13]. In detail, this task entails the following challenges:

- *Synonymy and Polysemy.* Due to the synonymy of natural language, many words or phrases can express similar meaning with the original terms. Some researchers have tried to apply an external knowledge base[1, 18], while existing knowledge bases can hardly cover a wide range of original terms and similar words referred by morphs. Meanwhile, considering the polysemy of natural language, a document might be innocuous even if the red-flagged terms or corresponding morphs are used. For example, *ice* and *skiing* refer to *Cocaine* in illegal drug dealing. But they refer to a kind of winter sports in more common situations.
- *Domain Knowledge.* Some morphs are generated based on domain knowledge. It is difficult to identify such obfuscations even for a man who is not familiar with this domain. For example, "薄熙来(Xilai Bo)" is a former Chinese politician. In Chinese social networks, Bo is morphed as "平西王(Conquer West King)" who was a historical figure four hundreds years ago. They have been linked together because they have similar experiences, both governed the same region and experienced a downfall and an arrest at the end of their career.
- *Unbounded candidates.* Obfuscated language effectively hides in plain sight because anything can be a candidate referred to the original term[25]. Netizens create morphs with a motley variety of functions. Some researchers have summarized the categories of obfuscated morphs [12, 24], such as phonetic substitution, spelling decomposition, translation and transliteration, semantic interpretation,etc. However, people's imagination is infinite and any connections can be utilized to create obfuscated language. So it is impractical to enumerate all obfuscation functions.

State of the Arts: With our observation, no matter how complicated rules are applied to create obfuscated language, the core features used can fall into two types, textual features and visual features. Previous researchers mainly focus on textual features to identify obfuscated language. Some methods [3, 4, 10] utilize the known morphs as seeds and find the patterns of the co-occurrence of the known morphs and the original terms, then find new morphs based on the acquired patterns. However, morphs are not used together with the original terms in most obfuscated language. Some methods pre-define rules to generate morphs [12, 24, 25]. The rules are all based on lexical relations, sematic relations or context relations between the original terms and morphs. Our method goes beyond the previous ones by targeting at identifying the adversarial passages obfuscated with various morphs rather than only finding morphs corresponding to the original terms.

Furthermore, we observe that some morphs are generated without any textual relations. But from the visual aspects, they are similar with the original terms. We name such relations as visual features. For example, the above mentioned morphs "火乍弓单" and "P0RN" belong to such type. The morphs based on visual features are created more casually than textual features based obfuscation. And it is impossible to summarize some generation rules since people can use any visually similar characters or symbols to replace the original terms. Thus makes previous methods fail to identify most content with visual features based obfuscation.

Our Solutions: To this end, we propose a multimodal text matching model to identify obfuscated language.

Firstly, we transfer this problem to a text matching task, where each document is checked whether matches a red-flagged term based on the matching degree of their relevance. Thus can help to overcome the first challenge since we mainly care about the relevance between documents and red-flagged terms no matter the original terms appears in the documents or not.

Besides, we adopt the Bi-directional Long Short Term Memory network(Bi-LSTM) to model textual features, which can help to extract the context information. This is useful to overcome the second challenge. When the original terms or morphs are used in a specific domain, their surrounding information (people, organizations, topics, events, etc) is always similar.

In order to overcome the third challenge, we propose to involve visual features to cover as many as possible candidates. We involve visual features by regarding each document as a source image and each red-flagged term as a template image in a template matching problem from computer vision area [6].

At last, we integrate both textual features and visual features in the neural network to achieve the final matching results. To evaluate the proposed method, we construct a large-scale dataset from Sina Weibo[1], which contains manually annotated labels. Experimental results show that our method outperforms previous methods.

Contributions: This paper presents a multimodal text matching model for identifying obfuscated language given a watch-list of red-flagged terms. To sum up, the contributions are as follows:

- the task of obfuscated language identification is novel and has not been carefully studied in previous methods. In this paper, we frame the problem as a text matching task and evaluate several methods suitable for this task.
- we introduce a multimodal text matching model that combines textual and visual features to identify obfuscated language.
- we conduct experiments on real-world dataset and validate the effectiveness of obfuscated language identification with multimodal features.

2 PROBLEM DEFINITION

The target of this paper is to identify obfuscated language in adversarial communication. The core problem of obfuscated language identification is to check whether a document is relevant to a red-flagged term in the given watch-list. To this end, we frame the given problem as a text matching task formalized as follows.

[1] http://weibo.com

Definition 2.1. Given a set of documents $D = \{d_1, d_2, ..., d_m\}$ and a watch-list of red-flagged term $Q = \{q_1, q_2, ..., q_n\}$. The matching task is to measure each pair of $< d_i, q_j >$ through a scoring function based on the representation:

$$match(d_i, q_j) = F_S(\Phi(d_i), \Phi(q_j))$$

where Φ is a function to map d_i and q_j to a representation vector, and F_S is the scoring function based on the relevance between them.

In this paper, we add a virtual term q_0 to the set Q which means none red-flagged. For each $d_i \in D$, if d_i have the highest matching score with $q_j (j \neq 0)$, then d_i is identified as an obfuscated document relevant to q_j. Note that the original terms or the substitution might appear, or both or neither appears in the obfuscated document.

3 THE PROPOSED MODEL

In this section, we describe our multimodal model for matching each document d_i to a given term q_j. The overall architecture of our proposed model is shown in Figure 1. For clarity, we describe our model in three parts: Textual-level Representation, Visual-level Representation, and Multimodal Fusion. The input of our model is a document d_i and a set of given terms Q. The output of our model is the selected term which gets the maximum probability that d_i belongs to. We will illustrate the details of the proposed framework in the following section.

3.1 Textual-level Representation

The textual-level representation tries to capture textual features of each document. The proposed method is based on the following hypothesis:

Hypothesis 1: When people talk about a subject, the textual context (entities, attributes or events) will be similar no matter whether the corresponding morphs are used or not.

For example, "Big Yao" is a morph of the NBA star "Ming Yao". When people use "Big Yao" to describe "Ming Yao", "Houston Rockets", "NBA", "Basketball" are mostly mentioned in the context, which is almost the same when talking about "Ming Yao".

According to hypothesis 1, the context around the terms and the corresponding morphs would not change much. Thus, we adopt word2vec[15] to generate the feature vector $V = \{v(x_1), ..., v(x_n)\}$ for each document, where $\{x_1, ..., x_n\}$ symbolizes the segments of the document.

To capture the context from both the past and the future, we adopt the Bidirectional LSTM(Bi-LSTM) network, which is a bidirectional formal model of LSTM [16]. Bi-LSTM can use the historical and future states data of input series simultaneously. It connects two recurrent neural networks with positive(forward state) and negative(backward state) time direction to a single output which has proved to be a good solution for expressing context information. With this structure, the output layer can obtain more tremendous information. Meanwhile, Bi-LSTM has the ability to solve the issue of learning long-term dependencies.

The structure of Bi-LSTM model is shown in Figure 2. Two directional neurons of Bi-LSTM do not interact. The general training process is with similar algorithms as LSTM. Forward and backward states are processed first in the forward pass, then do forward pass for output neurons. For the backward pass, output neurons are processed before forward and backward states are passed. Backward pass is similar to standard Back Propagation which involves reusing chain rules. Weights are updated only after the forward and backward passes are complete.

In a single LSTM unit, at the time slice t, the updating equations are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$h_t = o_t \times \tanh(C_t) \tag{5}$$

where f_t, i_t, o_t denote the forget gate's activation vector, the input gate's activation vector, and the output gate's activation vector respectively. σ and $tanh$ denote the sigmoid function and the $tanh$ function. C_t denotes the cell state vector and h_t denotes the output vector of the LSTM unit. $W_f, W_i, W_o, W_C, b_f, b_i, b_o, b_C$ denote the weight matrices and the bias vector parameters which need to be learned during training.

Bi-LSTM contains two separate LSTMs to extract both past and future information, where one encodes the sentence from the start to the end, and the other encodes the sentence with the reverse direction. Thus, we can obtain two representations $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, then the two representations are concatenated to acquire the final output:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{6}$$

After achieving the final hidden states $h_t (t = 1, 2, ..., n)$, the matching probability vector P_{text} corresponding to terms Q can be obtained using $Softmax$, which is the extracted feature of the textual-level representation.

$$P_{text} = Softmax(h_1, h_2, ..., h_n) \tag{7}$$

3.2 Visual-level Representation

The visual-level representation module captures the visual features based on template matching technique in computer vision. Template matching has proven to be a useful solution in digital image processing for finding small parts of a source image which matches a template image. This is similar to match documents with red-flagged terms. The red-flagged terms can be regarded as template images, and the documents to be checked can be regarded as source images. Then the matching process is like the template matching process to find the source images that match the template images. An example of a source image, a template image and the search area is shown in Figure 3. The arrow indicates the direction of the movement of the template.

Firstly, the red-flagged terms Q and documents D are all transferred to corresponding text images, where the font, size and color are unified. Given a document d_i and a red-flagged term q_j, the corresponding images are denoted as I and T. Then the template matching process is as follows.

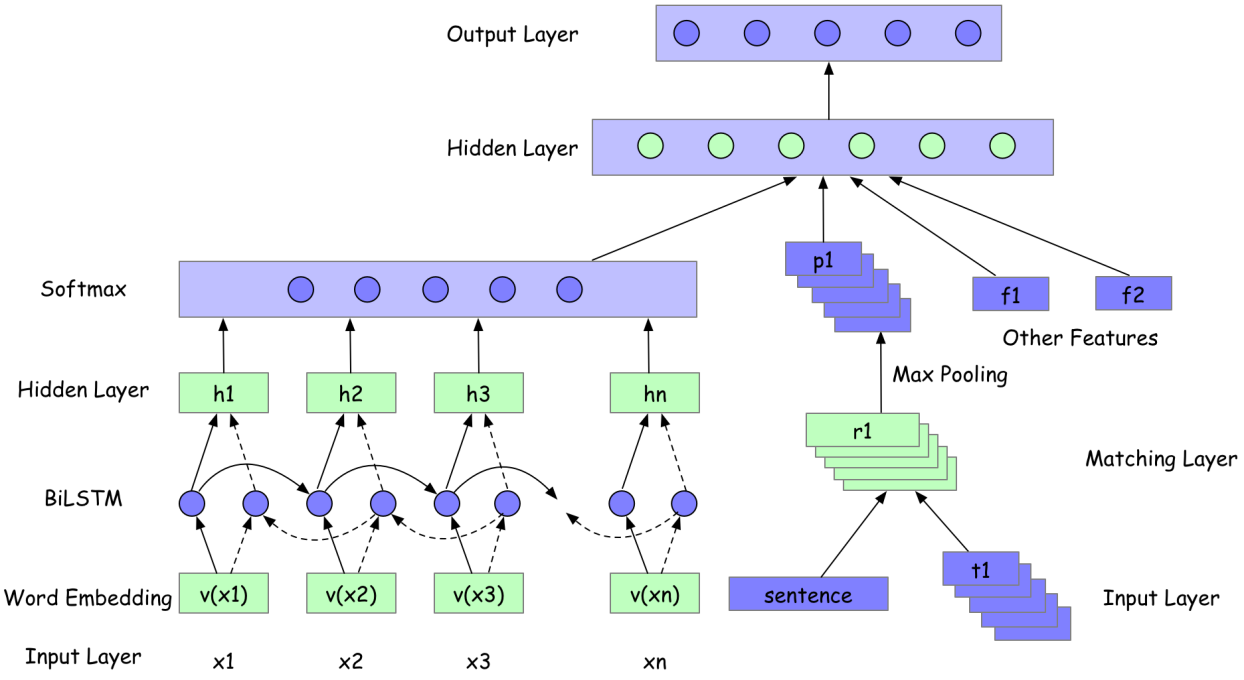

Figure 1: Architecture of the proposed multimodal model.

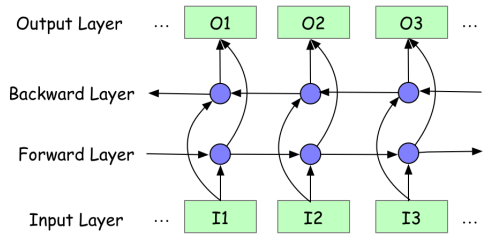

Figure 2: Structure of the Bi-LSTM model.

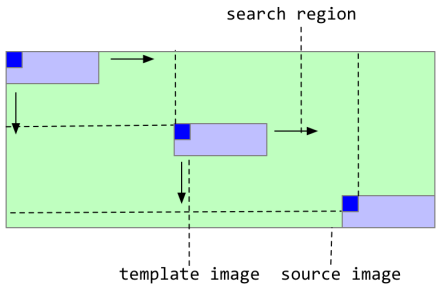

Figure 3: Source image, template image and search area.

(1) Position the template T over the image I at every possible location. In order to avoid phrases spanning multiple lines, I is generated with all content in only one line. That is, the height of I is identical to T. Then T is moved horizontally over I with one pixel each time.

(2) For each time T is positioned, we will compute the numeric metric of similarity between the template T and the image segment it currently overlaps with. The metric represents how similar the template T is to that segment of the source image I.

(3) For each location of T over I, the metric is stored in the result matrix R. Since the height of I is identical to T, R is an array in our experiment. Each location (x, y) in R contains the matching metric (normalized correlation matching):

$$R(x,y) = \frac{\sum_{x',y'} T(x',y') \cdot I(x+x', y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot I(x+x', y+y')^2}} \tag{8}$$

where (x, y) is the coordinate of the location T over I. (x', y') is the coordinate of the pixels in T. $T(x, y)$ and $I(x, y)$ is the RGB value of the pixel at the location.

(4) After T is slidden over the whole area of I and each $R(x, y)$ is calculated, the global max pooling is applied on the $R(x, y)$, which is done by applying a max filter to non-overlapping subregions of the initial representation. Then a vector P_{visual} is obtained:

$$P_{visual} = MaxPool(R) \tag{9}$$

Thus, P_{visual} is the output of the visual-level representation module. The highest value $Smax$ in P_{visual} corresponds to the matching position in R.

3.3 Multimodal Fusion

After achieving the outputs of the textual-level and the visual-level representations with the equations (7) and (9), we come to the multimodal fusion process. An existing work demonstrates that some non-linear interactions between the sentence vectors can help to learn good representations[17]. To achieve this, we include two common pair-wise vector operations: subtraction and multiplication. Then the integration process is as follows.

$$f_1 = P_{text} - P_{visual} \tag{10}$$

$$f_2 = P_{text} \times P_{visual} \tag{11}$$

$$F = P_{text} \oplus P_{visual} \oplus f_1 \oplus f_2 \tag{12}$$

where f_1 and f_2 denote two combining operations of P_{text} and P_{visual}. F denotes the final fusion features.

Finally, we utilize a Multi-Layer Perceptron (MLP) to serve as a matching model in which F is a input vector. Then F is considered to compute the probability distribution and the output layer presents the matching results.

4 EXPERIMENTS

4.1 Datasets

We crawl 1,245,812 tweets from Sina Weibo from October 1 to 31, 2017. We select 55 keywords as red-flagged terms and 186 morphs corresponding to the terms are found from the tweets. Some examples are shown in Table 1. Then we sample 43,206 non-redundant tweets as the experimental dataset and label them by three independent annotators. The annotated dataset is then randomly split into training set(27,642 tweets), development set(6,910 tweets) and test set(8,654 tweets). We tune parameters on the development set and apply them to get final results on test set.

All the 43,206 tweets are annotated by three Chinese native speakers. Each tweet is annotated with a red-flagged term or not. A tweet would be labeled with a given term as long as the tweet mentions the meaning of the term no matter the original term is used or not. Due to the polysemy of natural language, a tweet including a red-flagged term is possible to be annotated irrelevant with the given term.

4.2 Baselines

In this part, we will describe the methods in the comparative experiments. We mainly concern three groups of methods suitable for the given problem: classification methods, morph recognition methods and text matching methods. The methods are listed as follows:

- **Classification methods**: Such methods treat the problem in this paper as a classification problem. Each document will be classified into two categories (Relevant with the red-flagged terms or not). We select four popular models: **Naive Bayes (NB)**[20], **Logistic Regression (LR)**[21], **Decision Tree (DT)**[14] and **Support Vector Machine (SVM)**[22].
- **Morph Recognition methods**: Such methods are used to find all possible morphs to expand red-flagged terms for scanning. We select two state-of-art work on morph recognition **CEMD**[25] and **KIEM**[12]
- **Text Matching methods**: Such methods are used to match each document to the given terms, which is the same as our methods. We select one state-of-art work **MatchPyramid**[19], which takes text matching as image recognition.

4.3 Evaluation Metrics

To evaluate different methods mentioned above, we compare the outputs of different methods against the ground truth. The metrics for comparison are precision, recall and F1.

$$precision = \frac{|M_i \cap GT|}{|M_i|} \tag{13}$$

$$recall = \frac{|M_i \cap GT|}{|GT|} \tag{14}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{15}$$

where M_i is the set of output tweets by the i-th method, GT is the set of tweets with red-flagged labels.

4.4 Parameter Setting

To initialize the word embeddings used in our method, we pre-train the word embeddings on 2 million documents from Sina Weibo and news websites. The sentence length is set to 20. The dimensions of the word embeddings and Bi-LSTM units are set 200. The model is trained using RMSprop optimizer with a learning rate 0.001.

4.5 Results Analysis

Overall Evaluation. Table 2 shows the comparison results on precision, recall and F-measure. From the results, we can observe that our approach performs better than the other methods. The morph recognition methods CEMD and KIEM perform better than the classification methods because they extend the original terms by finding more morphs. But they fail to generate all correct morphs so that the precision and recall are not as good as our proposed methods. Though MatchPyramid also solve this problem in a text matching way, the F1 of our approach is 12% better than it. This validates the benefit of involving visual features.

Zero-shot Evaluation. We now turn to the evaluation of generalization behavior with respect to some obfuscated morphs that have been never seen by the systems. We consider the zero-shot

Table 1: Some Examples in Datasets

Red-flagged Terms	Morphs	Tweets
炸弹(Bomb)	火(Fire)乍(Sudden)弓(Arrow)单(Single)	如何组装N2火乍弓单 How to assemble the bomb N2
赌博(Gamble)	堵(block)博(doctor)	我的梦想就是靠堵博发家致富 My dream is to become rich through gambling
裸体(Nude)	果(Fruit)体(Body)	电影中十大令人难忘的果体场面... Top-10 nude scenes in movies...

Table 2: Overall Evaluation Result

Methods	Precision	Recall	F1
NB	0.82	0.64	0.72
LR	0.83	0.66	0.74
DT	0.84	0.63	0.72
SVM	0.85	0.66	0.74
CEMD	0.86	0.81	0.83
KIEM	0.88	0.85	0.86
MatchPyramid	0.87	0.77	0.82
Our Approach	**0.93**	**0.91**	**0.92**

setting, i.e. some morphs for the original terms only appear in the test set and are not used during training. This is more similar with practical cases because morphs are not static and evolves rapidly over time. The evaluation results are summarized in Table 3. Our approach also performs best in comparison with the other methods. The drop ratio means the decrease compared with the overall evaluation. We can observe that our approach also achieves the lowest drop ratio. Previous methods fail to keep high precision and recall if they never see the morphs in training set since the context might also change when a new morph is created. The result also validates that our approach can cover more kinds of obfuscation by combining textual and visual features.

Table 3: Zero-Shot Evaluation Result

Methods	Precision	Recall	F1	Drop Ratio
NB	0.49	0.34	0.40	44.16%
LR	0.67	0.41	0.51	30.82%
DT	0.67	0.43	0.52	27.25%
SVM	0.74	0.45	0.56	24.68%
CEMD	0.76	0.67	0.71	14.63%
KIEM	0.78	0.71	0.74	14.04%
MatchPyramid	0.75	0.64	0.69	15.46%
Our Approach	**0.84**	**0.82**	**0.83**	**9.79%**

5 RELATED WORK

Obfuscated Language Identification has been recently proposed [13]. Several previous work on name alias detection, word obfuscation, and morph resolution can contribute to this task. The basic idea of detecting name alias is automatically finding the patterns between the real names and aliases, and then these patterns are utilized to search for candidates. In the end, various ranking scores are calculated to measure the associations between a name and its candidates [3, 4, 10]. Yin et al. introduce a system to extract and rank name aliases in emails [23]. However, the co-occurrence of the real names and aliases is not common in most adversarial communications. Brennan et al. summarize three types of approaches to create adversarial messages: obfuscation, imitation, and translation [5]. They find manual circumvention approaches work best, while automated translation approaches are not so effective. Other recent work detects word obfuscation in adversarial communication using existing commonsense Knowledge Bases such as ConceptNet to expand similar words for substitution [1, 7]. Morph resolution is another closely related work to this task which aims to resolve the morphs to the target entities. Encoding and decoding entity morph is a special case of coded name alias to hide the original entities for evading censorship[11, 12, 24, 25]. Recent works propose a number of novel methods to automatically encode implicit and relevant morphs[24], including Phonetic Substitution, Spelling Decomposition, Nickname Generation, Translation and Transliteration and Historical Figure Mapping, which can effectively avoid decoding detection. They also study on effective morph decoder methods, which can automatically recognize and resolve entity morphs to their real targets [11, 12, 25]. Such methods need to take the encoding rules into consideration.

Our model differs from these previous methods, and improves the model in two critical ways: (1) propose a multimodal text matching model which can overcome the challenges of Synonymy and Polysemy and Domain Knowledge. (2) incorporate visual features to cover as many as possible morphs in obfuscated language

6 CONCLUSIONS

In this work, we propose a multimodal text matching model for obfuscated language identification. We introduce a multimodal model by integrating a text-level module and a visual-level module. In textual level, we employ Bi-LSTM model to extract the matching probability feature. In visual level, we utilize template matching to get the visual matching feature. Finally, we evaluate our approach with a manually annotated dataset collected from Sina Weibo. Compared to other state-of-the-art models that use only textual features, the performance of the proposed method is much better. We will try to extend to other languages in the future.

REFERENCES

[1] S. Agarwal and A. Sureka. 2015. Using common-sense knowledge-base for detecting word obfuscation in adversarial communication. In *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*. 1–6. https://doi.org/10.1109/COMSNETS.2015.7098738

[2] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The Role of Social Networks in Information Diffusion. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*. ACM, New York, NY, USA, 519–528. https://doi.org/10.1145/2187836.2187907

[3] Danushka Bollegala, Taiki Honma, Yutaka Matsuo, and Mitsuru Ishizuka. 2008. Mining for personal name aliases on the web. In *International Conference on World Wide Web*. 1107–1108.

[4] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011. Automatic Discovery of Personal Name Aliases from the Web. *IEEE Transactions on Knowledge & Data Engineering* 23, 6 (2011), 831–844.

[5] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry:Circumventing authorship recognition to preserve privacy and anonymity. *Acm Transactions on Information & System Security* 15, 3 (2012), 1–22.

[6] Roberto Brunelli. 2009. *Template Matching Techniques in Computer Vision: Theory and Practice*. WILEY. 261 pages.

[7] Sonal N. Deshmukh, Ratnadeep R. Deshmukh, and Sachin N. Deshmukh. 2014. Performance Analysis of Different Sentence Oddity Measures Applied on Google and Google News Repository for Detection of Substitution. *IRJES Com* (2014).

[8] Sw. Fong, D. Roussinov, and D. B. Skillicorn. 2008. Detecting Word Substitutions in Text. *IEEE Transactions on Knowledge & Data Engineering* 20, 8 (2008), 1067–1076.

[9] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. 2013. Information Diffusion in Online Social Networks: A Survey. *SIGMOD Rec.* 42, 2 (July 2013), 17–28. https://doi.org/10.1145/2503792.2503797

[10] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. 2014. Discovering emerging entities with ambiguous names. In *International Conference on World Wide Web*. 385–396.

[11] Hongzhao Huang, Zhen Wen, Dian Yu, Heng Ji, Yizhou Sun, Jiawei Han, and He Li. 2013. Resolving Entity Morphs in Censored Data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 1083–1093.

[12] Longtao Huang, Lin Zhao, Shangwen Lv, Fangzhou Lu, Yue Zhai, and Songlin Hu. 2017. KIEM: A Knowledge Graph Based Method to Identify Entity Morphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM, New York, NY, USA, 2111–2114. https://doi.org/10.1145/3132847.3133123

[13] Heng Ji and Kevin Knight. 2018. Creative Language Encoding under Censorship. In *Proceedings of COLING2018 Workshop on NLP for Internet Freedom*. 23–33.

[14] D Landgrebe. 2002. A survey of decision tree classifier methodology. *IEEE Transactions on Systems Man & Cybernetics* 21, 3 (2002), 660–674.

[15] Changzhou Li, Yao Lu, Junfeng Wu, Yongrui Zhang, Zhongzhou Xia, Tianchen Wang, Dantian Yu, Xurui Chen, Peidong Liu, and Junyu Guo. 2018. LDA Meets Word2Vec: A Novel Model for Academic Abstract Clustering. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018*. 1699–1706.

[16] Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

[17] Allen Nie, Erin D. Bennett, and Noah D. Goodman. 2017. DisSent: Sentence Representation Learning from Explicit Discourse Relations. *CoRR* abs/1710.04334 (2017). http://arxiv.org/abs/1710.04334

[18] Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised Entity Linking with Abstract Meaning Representation. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. 1130–1139.

[19] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016*. 2793–2799.

[20] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Machine Learning, Proceedings of the Twentieth International Conference ICML 2003,*. 616–623.

[21] Walter A. Shewhart and Samuel S. Wilks. 2005. *Applied Logistic Regression, Second Edition*.

[22] Abhisek Ukil. 2002. Support Vector Machine. *Computer Science* 1, 4 (2002), 1–28.

[23] Meijuan Yin, Xiaonan Liu, Junyong Luo, and Xiangyang Luo. 2013. A System for Extracting and Ranking Name Aliases in Emails. *Journal of software* 8, 3 (2013), 737–745.

[24] Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and Bulent Yener. 2014. Be Appropriate and Funny: Automatic Entity Morph Encoding. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*. 706–711.

[25] Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Sujian Li, Chin-Yew Lin, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and Bulent Yener. 2015. Context-aware Entity Morph Decoding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*. 586–595.