Chain Association-based Attacking and Shielding Natural Language Processing Systems

Jiacheng Huang D190201005@stu.cqupt.edu.cn
Long Chen * chenlong@cqupt.edu.cn
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, 400065, Chongqing, China

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

Association as a gift enables people do not have to mention something in completely straightforward words and allows others to understand what they intend to refer to. In this paper, we propose a chain association-based adversarial attack against natural language processing systems, utilizing the comprehension gap between humans and machines. We first generate a chain association graph for Chinese characters based on the association paradigm for building search space of potential adversarial examples. Then, we introduce an discrete particle swarm optimization algorithm to search for the optimal adversarial examples. We conduct comprehensive experiments and show that advanced natural language processing models and applications, including large language models, are vulnerable to our attack, while humans appear good at understanding the perturbed text. We also explore two methods, including adversarial training and associative graph-based recovery, to shield systems from chain association-based attack. Since a few examples that use some derogatory terms, this paper contains materials that may be offensive or upsetting to some people.

Keywords: adversarial examples; chain associations; natural language processing; black box

1. Introduction

In past years, many studies has shown that the adversarial examples can cause decision-making errors in natural language processing (NLP) systems Formento et al. (2023); Ou et al. (2022), even in large language models Wang et al. (2024) (LMMs).

However, existing adversarial attacks only consider the attack strategies in a direct way while ignoring the complexity of textual adversarial attacks in reality. For example, Chinese words "幼稚", which means "*naive*", is an adjective with emotional tendency, but it will not be recognized as an emotional word by an emotion analysis system after being substituted with "拿衣服", which is a verb object phrase meaning to "*take clothes*". The relation between "幼稚" and "拿衣服" is not a simple single-layer mapping, but a multi-layer mapping. Specifically, we associate "幼稚" with "*naive*" first, which is the English translation of "幼稚", and then further associate it with "拿衣服", which is one of the Mandarin transliterations of "*naive*". Note that the above is only a simple example of word substitution based on chain association while the associative ability of human being is complex.

* corresponding author

English has analogous cases. Take "screw", a polysemy referring to "a metal object like a nail" or "having sex with someone". Attackers online can replace "screw" with its related emoji to form offensive phrases like "🔩 you", which utilizes human associations about the corresponding word of emoji and its polysemy. Note that this example is merely used as an analogy to explain our idea, in fact, this work only focuses on Chinese adversarial examples.

In this work, we investigate to what extent advanced Chinese NLP systems are sensitive to chain association-based attack and explore various shielding techniques. Specifically, we first generate a chain association graph for Chinese characters based on the association paradigm for building search space of potential adversarial examples. Then, we regard generating adversarial examples as a problem of combinatorial optimization and introduce an discrete particle swarm optimization algorithm to search for the optimal adversarial examples. We show that advanced NLP models and applications are extremely vulnerable to our attack. To our best knowledge, we are the first to introduce chain association in adversarial attack. Furthermore, we also explore two methods to protect NLP systems from our attacks.

2. Related Work

Although adversarial attacks were first proposed in the field of image recognition, however, due to the discrete nature of textual data and the uncertainty brought by perturbations to the semantic quality of text, it is difficult to directly migrate adversarial attack algorithms in the image domain to textual data types. Therefore, researchers have conducted extensive studies on textual data, forming different textual adversarial attack paradigms with different perturbation granularities, such as character-level, word-level, and sentence-level. The form of character-level attacks vary across different linguistic and cultural environments. In the English context, character-level attacks often exploit visual perturbations, including artificially constructed spelling errors such as the insertion, deletion, swapping, and modification of letters within words Formento et al. (2023). However, in the Chinese context, handwritten stroke errors on paper do not occur in electronic devices based on input methods. Therefore, character-level attacks in Chinese environments usually manifest as the replacement of homophones Cheng et al. (2020) or the utilization of visual character disassembly Ou et al. (2022).

Word-level adversarial attacks achieve the deviation of semantic vectors of examples by disturbing the input at the word level, making them cross the decision boundary and thus leading to incorrect model outputs. As the core method of this strategy, word substitution covers various strategies, including word vector similarity Jin et al. (2020), synonyms Ren et al. (2019), sememes Zang et al. (2020), and language model scores Zhang et al. (2019). Word-level adversarial attacks do not violate the grammatical rules of the text and maximize the retention of the original semantics, thus exhibiting better performance in terms of adversarial text quality and attack success rate. Additionally, the utilization of language models for control also ensures the fluency and smoothness of the adversarial text.

Sentence-level adversarial attacks treat the entire original input sentence as the target of perturbation, carefully reconstructing the textual content by generating adversarial texts that maintain the same semantic meaning as the original input but cause the victim model to make incorrect decisions. Common sentence-level adversarial attack methods include re-

encoding and decoding after encoding Han et al. (2020), adding irrelevant sentences Liang et al. (2018), and paraphrasing Xu et al. (2021). Sentence-level adversarial attacks face greater difficulties in maintaining the original semantics.

Our work is a kind of Chinese multi-level perturbation based on the gift of human beings namely chain associations, which is utilizing the comprehension gap between humans and machines in cognition.

3. Associations in Textual Adversarial Examples

3.1. Motivation

Abundant association ability is vested in human beings, who are able to associate things that seem totally different but related. The gift enables people do not have to mention something in completely straightforward words and allows others to understand what they intend to refer to. Associationism psychology holds that all complex mental processes, such as thinking, learning, and memory, can be mainly explained by the associative links that connect ideas, according to specific laws and principles Bracken et al. (2021), e.g., Philosopher David Hume's Laws of Association: (i) Law of Similarity, (ii) Law of Contiguity, and (iii) Law of Causality. The Law of Similarity states that when two things are very similar to each other, the thought of one will often trigger the thought of the other. The Law of Contiguity states that we associate things that occur close to each other in time or space. The Law of Causality states that we associate things when there is a causal relationship with them.

We believe that the distance between associative words is close in human cognition even if it is far in word meaning, and such an inconsistency between two kinds of distances of associative words provides the motivation for this paper. Specifically, the existing textual adversarial attacking strategies, such as word substitutions and misspellings, are special cases of the laws of association. For examples, the synonym-based substitution belongs to the law of similarity in word meaning and the misspellings belongs to the law of similarity in vision. All these strategies take advantage of the inconsistency between the distances in human cognition and word meaning.

Furthermore, the association of words is not necessarily single-layer as a chain association chain will be formed while associating constantly from one word to another. We believe that such a chain association of words will aggravate the inconsistency of distances in human cognition and word meaning, and cause the deep neural networks to fail blatantly.

3.2. Rules of Word Association

It should be noted that not all associations can be used in the field of textual adversarial examples, as inappropriate associative word substitutions will cause the text to be unreadable. Thus, we summarize several rules of word association can be used in textual adversarial attack, according to Philosopher David Hume's laws of association and Chinese cultural environment, as shown in Table 1. The Law of Contiguity is excluded because it is difficult to introduce this law into textual adversarial attack.

With the help of knowledge graph technology, the association chain can be fully expressed while entity refers to word and edge refer to the associative relationship between

Table 1: Rules of Word Association based on David Hume's Laws of Association. The detailed explanations and implementation methods can be see in section 4.1.

Laws of Association	Rules of Word Association
Law of Similarity	English Translation
	Transliteration
	Fuzzy Matching
	Visually Similar Characters
	Characters Disassembling
Law of Causality	Chinese Pinyin
	Acronym
	Hanzify

words. Words out of vocabulary may appear while associating words in some way, such as the example mentioned in the introduction. Although these words have no specific meaning, we believe that readers will start a word guessing process which is kind of like completing a cloze, i.e., mask the unknown words temporarily and infer what the author intends to write after reading context. Due to the association between the original word and guessed word, readers can confirm whether they guessed correctly. Moreover, the word guessing process will be easier than normal cloze, since the topics involving supervision are usually confined to crime, pornography, and dirty words.

4. Approach

Generally, words in the original sentence vary in their impact on model predictions. Minor sentence alterations, especially replacing key words, can significantly alter predictions. Following Li et al. (2019), we identify the most influential words by measuring their removal effects and replace them in order of importance. The challenge lies in generating suitable substitutes and determining optimal adversarial examples that deceive the model while resembling the original. Here, we propose 1) an associative knowledge graph and 2) an adversarial search strategy.

4.1. Associative Knowledge Graph

The associations can be represented by a graph intuitively so that we consider building an associative knowledge graph G as shown in Fig. 1. In fact, due to the complexity of human association ability, we can not completely enumerate all the types of word association but summarize some typical rules to test our ideas. Fortunately, our design is highly scalable, allowing new rules and graph updates. Next, we elaborate on these rules and their implementation.

English Translation. We view English translations of Chinese characters as word associations based on similarity of meaning. For example, "*naive*" translates to "幼稚". Many Chinese online users know both languages, and English is crucial in Chinese education. English translations can be accessed via third-party platforms.

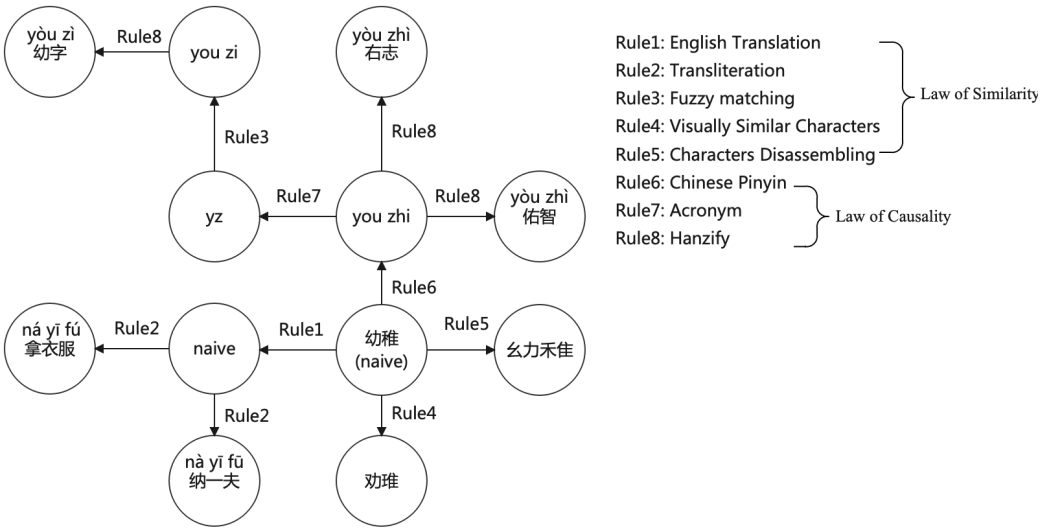

Figure 1: An example of chain-association based knowledge graph.

Chinese Pinyin. Chinese Pinyin is the Chinese phonetic notation with Latin alphabets according to Laws of Causality while each Chinese character has its corresponding Chinese Pinyin. For instance, it can be written as "*you zhi*" if we were to represent the pronunciation of the Chinese characters "幼稚" in Chinese Pinyin. The conversion between Chinese characters and Chinese Pinyin can be implemented with the Python third-party library named pypinyin.

Transliteration. Transliteration refers to the translation of foreign words with Chinese characters with similar pronunciations and it is based on Laws of Similarity. This kind of Chinese characters used for transliteration no longer have their original meaning but only retain their pronunciation and writing form. For example, consider the pronunciation of the word "*naive*" as /naɪˈiːv/. This pronunciation is akin to the Chinese characters "拿(ná)衣(yī)服(fú)" in terms of phonetics. To be more precise, /naɪ/ is similar to "*ná*", /i/ is reminiscent of "*yī*", and /v/ bears resemblance to "*fú*". We only consider the transliteration from English to Chinese in this work and implement it by establishing the mapping relationship between English phonetics and Chinese Pinyin, as the number of vowels and consonants in English is limited and their pronunciations can be similarly expressed with Chinese pinyin.

Acronym. Acronym namely a word composed of the first letters of the words in a phrase, which is often used as an abbreviated form of Chinese network language, can be used as an association word of Chinese pinyin or English word. It is based on Law of Causality as the acronym cause readers want to figure out what the complete phrase is. For example, "lol" is typically an abbreviation for "laugh out loud" in English. Similarly, in Chinese internet slang, "nb" is often an abbreviation for "牛(niú)逼(bī)", which means awesome. The conversion from Chinese Pinyin or English word to its acronym can be implemented with built-in Python string operations.

Fuzzy Matching. Fuzzy matching for Chinese Pinyin, i.e., search possible Pinyin starting with given first letters (i.e., acronym), is a common substitution for keywords in Chinese offensive text online recently, which often escapes offensive text automated detection systems. For instance, due to their shared first letters in Pinyin, "特(tè)喵(miāo)的(de)" in Chinese internet slang carries the same meaning as "他(tā)妈(mā)的(de)" which is an offensive term equivalent to "fuck", while "特(tè)喵(miāo)的(de)" originally lacks a specific meaning of its own. It is based on Law of Similarity because they have similar consonants. The operation can be implemented by arranging and combining the first letters and vowels since the first letters are often consonants.

Hanzify. Hanzify, namely conversion from Chinese pinyin to Chinese characters in this paper, can also be an association link connected behind the fuzzy matching in the association chain, which is implemented simply with Python third party library named Pinyin2Hanzi. For example, the Pinyin "hao" can be converted into various Chinese characters, such as "好"(good), "号"(number), and "郝"(a Chinese surname). It is based on Law of Causality as Chinese Pinyin cause readers to associate the corresponding Chinese characters.

Visually Similar Characters. Visual similarity can play a considerable role in replacing the important words in a sentence according to Law of Similarity, as a large number of characters with visual similar composition or shape exist in Chinese. For instance, the shape of the character "日" is similar to the character "曰" visually, even though they have completely different meanings. Specifically, the character "日" means "day" in a general context or "fuck" in an offensive context, while "曰" means "say" in classical Chinese language. To this end, about twenty thousand Chinese characters are collected and converted into image representation using Python built-in library named PIL, and we build character embedding space with visual shape for retrieving similar characters.

Characters Disassembling. Chinese characters include a large number of combined characters composed of several single characters, some of which can be disassembled transversely and not or only mildly affect reading as the characters are left-right structures according to Law of Similarity. For example, Chinese character "幼" can be disassembled into "幺" and "力", i.e., "幺力". Similarly, the Chinese character "稚" can be disassembled into "禾" and "隹", i.e., "禾隹". We collected the required data by crawler from a third-party website providing characters disassembling service.

Using the above rules, we can generate a large number of candidates for any word that needs to be replaced in a sentence. In order to modify the original sentence as little as possible, we only take the words whose importance is the one-third of the original sentence as the substituted words in this work.

4.2. Particle Swarm Optimization-based Search Strategy

The original sentences may contain multiple important words to be replaced and, for each word, abundant associative words exist according to the association rules. Besides, as we can see in Fig. 1, there are multiple paths and multiple layers to retrieve the associative words of a original word. Thus, an efficient adversarial examples search strategy is essential while a huge search space composed of numerous potential adversarial examples needs to be handled and, furthermore, the path from the original word to the final associative word is expected to be as short as possible for reducing the burden of comprehension to

readers. Hence, we consider exploiting particle swarm optimization (PSO) to search for final adversarial examples in the search space.

However, rather than the original PSO, whose search space is continuous, a variation suitable for discrete space composed of potential textual adversarial examples is what we need. Inspired by Zang et al. (2020), a *position* in the search space now corresponds to a potential adversarial example while each dimension of a *position*, i.e., sentence, corresponds to a word. Besides, the *velocity* of particle now corresponds to a probability vector, and each dimension of a velocity refers to the corresponding dimension of the related *position* will change, i.e., the probability of a word will be substituted.

Formally, we denote an original sentence as $x^o = x_1^o...x_n^o...x_N^o$, where N is the length of original sentence and x_n^o is the n-th word in original sentence. A position in the search space, i.e., a potential textual adversarial example related to x^o, is denoted as $x^a = x_1^a...x_n^a...x_N^a, x_n^a \in \mathbb{A}(x_n^o)$, where $\mathbb{A}(x_n^o)$ is the set of x_n^o and all its associative words. A velocity of a particle, is denoted as $v = v_1...v_n...v_N$, where v_n refers to the probability with which determines whether n-th dimension of the particle's position will move.

Since we expect that the final adversarial example can not only fool the victim model but also the association paths between substitution words and original words are as short as possible in total, the optimization score of a position is calculated by following the formula:

$$score(x^a) = \frac{1 - C(x^a)}{L(x^o, x^a)} \tag{1}$$

where $C(x^a)$ is the confidence of the true label of x^a given by the victim model, and $L(x^o, x^a)$ is the total number of layers from the associative words in x^a used to replace original words in x^o. If the current number of iterations reaches the maximum, the algorithm will terminate and output the position of the particle in the *global best previous position* as the search result.

When presented with an original sentence, each particle is given a stochastic *position* x and *velocity* v. Specifically, the important words in the original sentence are substituted with their direct associative words and take the modified sentence as the initial *position* of a particle. In addition, the *velocity* v of particle in discrete search space is a probability vector, hence, each dimension of v is initialized randomly between interval $[0.0, 1.0]$ and updated by the following formula:

$$\begin{aligned} v_n &= S(\omega v_n + \varphi_1 I(p, x) + \varphi_2 I(p^g, x)) \\ I(a, b) &= \begin{cases} -1, & a = b \\ 1, & a \neq b \end{cases} \end{aligned} \tag{2}$$

where $S(*)$ is a sigmoid limiting transformation for constraining v_n to the interval $[0.0, 1.0]$ since v_n is a probability. It can be reasonably calculated as that v_n is going to decrease, remain the same, or increase when the particle's position is at the global best previous position, individuals best previous position only, or both not.

In addition, compared with the fixed value, a dynamic decreasing ω derived from a measure function enables particles to explore more positions in the early stage and gather around the best positions in the final stage. Thus, we introduce a nonlinear dynamic adaptation function to update ω. Specifically,

$$\omega = \omega_{max} - \frac{t^2 \times (\omega_{max} - \omega_{min})}{t_{max}^2} \tag{3}$$

where t and t_{max} are the current and max numbers of iterations separately. ω decreases slowly during the initial iteration, which is conducive to exploring the local optimum at an early stage, and ω decreases rapidly while approaching the maximum number of iterations for improving the efficiency of converging to the global optimum.

Besides, the original position update formula that makes addition is also not suitable for discrete space. Inspired by Kennedy and Eberhart (1997), we propose a probabilistic approach to update the position of particles. A probability P is introduced with which a particle determines whether one of its position's dimensions, i.e., x_n, moves to the corresponding dimension of *global best previous position* p_n^g. The movement of each dimension of a particle' position at an iteration is redefined by the following rule:

$$\begin{aligned} &if\ rand() < P\ and\ rand() < v_n \\ &then\ x_n = p_n^g \\ &else\ rand() > P\ and\ rand() < v_n \\ &then\ x_n = G_{adj}(x_n) \end{aligned}$$

where $rand()$ refers to a random number selected from a uniform distribution in $[0.0, 1.0]$, and $G_{adj}(x_n)$ refers to one of the words adjacent to x_n at random in the associative graph G. In addition, to encourage particles to explore more positions according to associative graph G at an early stage and search for better positions around the global best position at a final stage, P varies with iteration as follow:

$$P = P_{min} + \frac{t^2 \times (P_{max} - P_{min})}{t_{max}^2} \tag{4}$$

where $0 < P_{min} < P_{max} < 1$, and we can see the probability of particles moving to the *global previous best position* will increase with the number of iterations.

5. Experiments

In this section, we conduct comprehensive experiments to evaluate our chain association-based attack on Chinese NLP systems and study the methods of shielding against chain association-based adversarial attack.

5.1. Attacking

Victims Models and Applications. To investigate the effects of chain association-based attack, we evaluate our attack method on five text classification models, namely Fasttext Joulin et al. (2017), TextCNN Guo et al. (2019), Attention-based Bidirectional LSTM (BiLSTM) Xiaoyan and Raga (2023) and BERT Si and Wei (2023). Besides, we also test our attack methods on two industry-leading commercial applications used for

offensive text detection, namely Baidu Text Censoring[1] (BaiduCensor) and Alibaba Content Security[2] (AliSecurity).

In addition, due to the significant impact of large-scale models on artificial intelligence recently, we also conduct experiments on LLMs, i.e., ERNIE Bot and Tongyi Qianwen, released by Baidu and Alibaba respectively. Due to the outputs of some LMMs are not easy to control while using the prompt-based paradigm, it is difficult to apply our attack algorithm directly to LMMs while each iteration in our adversarial attacking algorithm requires getting the confidence of classification of the victim model. Thus, as an alternative, we used the paradigm of transfer-based attack to conduct experiments on LMMs, that is, we obtain adversarial examples generated on the local model, and then use prompt-based paradigm to get answers for classification of LMMs by inputting prompt. All prompts used in this paper are presented in Appendix A.

Dataset and Associative Graph. We use public Chinese datasets, i.e., Meituan [3] and Amazon comments. Besides, we collect offensive text and normal text from online social platforms labeled by native Chinese speakers. We divide the dataset into two parts, i.e., 80% for training and 20% for testing. Details of the datasets are shown in Table 2.

Table 2: Statistics for the datasets. 'Avg.Len' refer to the average length of examples.

Dataset	Class	Avg.Len	Train	Test
Meituan Comments	2	19	9600	2400
Amazon Comments	2	24	9600	2400
Offensive Text	2	51	8000	2000

Baseline Methods. To evaluate our chain association-based adversarial attack more comprehensively, we implemented two baseline methods and compared them with ours. The two baseline methods are 1) GreedyAttack Ou et al. (2022), and 2) WordChange Cheng et al. (2020), both of which represent multi-strategy approaches for generating Chinese adversarial examples.

Attacking Performance. The performances and attack results of all models and applications on Meituan, Amazon and offensive text are listed in Table 3, Table 4 and Table 5 respectively. "ACC" refers to the accuracy of models and applications in different conditions and "WMD" refers to Word Mover's Distance between original text and perturbed text. We observe that our chain association-based adversarial attacking method decreases the accuracy of the victim models and applications the most compared with the other two baseline methods. For example, it attacks Baidu Text Censoring's service and reduce its accuracy from 96.4% to 25.2% notably for offensive text detection, which demonstrates the vulnerability of existing offensive text automated detection applications.

Besides, the attack results of LMMs are shown in Table 6, which demonstrates that LMMs can be also affected by the adversarial examples generated by our method. But we also observe that the performances of adversarial attacks on LMMs when applied to offensive text are not significant. This could be due to the reason that the offensive texts

1. https://ai.baidu.com/solution/censoring
2. https://homenew.console.aliyun.com
3. Meituan is a platform for ordering takeaway, which contains positive and negative user comments

contain a high degree of offensive context, which enables LMMs to maintain their robustness in the face of such kinds of adversarial examples.

Table 3: Attack performances of different attack methods against victim models on Meituan Comments.

Model	Original		GreedyAttack		WordChange		Ours	
	ACC	WMD	ACC	WMD	ACC	WMD	ACC	WMD
Fasttext	0.858	N/A	0.350	0.536	0.364	0.528	**0.184**	0.524
TextCNN	0.885	N/A	0.396	0.494	0.375	0.497	**0.227**	0.499
BiLSTM	0.894	N/A	0.374	0.521	0.371	0.485	**0.279**	0.492
BERT	0.934	N/A	0.543	0.498	0.528	0.534	**0.328**	0.503

Table 4: Attack performances of different attack methods against victim models on Amazon Comments.

Model	Original		GreedyAttack		WordChange		Ours	
	ACC	WMD	ACC	WMD	ACC	WMD	ACC	WMD
Fasttext	0.898	N/A	0.346	0.476	0.453	0.482	**0.327**	0.498
TextCNN	0.900	N/A	0.375	0.490	0.490	0.480	**0.335**	0.512
BiLSTM	0.936	N/A	0.412	0.472	0.532	0.477	**0.381**	0.508
BERT	0.944	N/A	0.748	0.501	0.724	0.548	**0.532**	0.528

Table 5: Attack performances of different attack methods against victim models and applications on offensive text dataset. Note that the ACC only represents the accuracies of victims classifying offensive text since we only consider the approaches causing the victims failed to recognize offensive text in this task.

Model/Application	Original		GreedyAttack		WordChange		Ours	
	ACC	WMD	ACC	WMD	ACC	WMD	ACC	WMD
Fasttext	0.826	N/A	0.598	0.481	0.738	0.427	**0.496**	0.482
TextCNN	0.853	N/A	0.584	0.478	0.736	0.432	**0.448**	0.492
BiLSTM	0.712	N/A	0.558	0.484	0.580	0.425	**0.425**	0.497
BERT	0.858	N/A	0.768	0.476	0.724	0.430	**0.328**	0.485
BaiduCensor	0.964	N/A	0.326	0.462	0.378	0.425	**0.252**	0.454
AliSecurity	0.898	N/A	0.456	0.465	0.496	0.428	**0.366**	0.472

Human Annotation. We also asked three human annotators to recover the original sentences given some perturbed text. Specifically, for each dataset, every annotator is required to recover 50 randomly picked sentences generated by our approach. Our rationale

Table 6: Attack performances of different attack methods against LMMs on Meituan Comments, Amazon Comments and offensive text dataset.

LMMs	Dataset	Original	GreedyAttack	WordChange	Ours
ERNIE Bot	Meituan Comments	0.892	0.653	0.749	0.577
	Amazon Comments	0.863	0.538	0.691	0.506
	Offensive Text	0.931	0.869	0.876	0.864
Tongyi Qianwen	Meituan Comments	0.847	0.785	0.791	0.665
	Amazon Comments	0.868	0.703	0.764	0.621
	Offensive Text	0.922	0.845	0.873	0.827

for including this recovery task is to test robustness of human perception under our perturbations. We evaluate by measuring the Word Mover's Distance between the recovered and the original text, averaged over all sequence pairs and all human annotators. The results are shown in Table 5.1 and show that adversarial examples generated by WordChange are the most easily recoverable to the original text by humans, whereas the adversarial examples generated by our method are somewhat more challenging to restore. However, given the attack performance of our method, we believe that this trade-off is justified.

Table 7: The Word Mover's Distance between original vs. perturbed and original vs. recovered text for different adversarial attacks.

Attack Strategy	WMD	
	original vs. perturbed	original vs. recovered
GreedyAttack	0.526	0.163
WordChange	0.528	0.084
Ours	0.507	0.179

Transferability. The transferability of adversarial examples reflects a attacking generalization ability while adversarial examples with high transferability can fool different victims successfully, and it allows attackers to attack the target model without accessing to it. We evaluate the transferability of our adversarial examples by inputing adversarial examples generated for each models into other different models and record the accuracies. Table 8 shows the accuracies of models classifying transfered adversarial examples and it demonstrates that our chain association-based adversarial attack crafts adversarial examples with a notable transferability.

5.2. Shielding

Without losing generality, we study two methods for shielding our attack on offensive text dataset, namely adversarial training (**AT**) and associative graph-based recovery (**AGBR**). For **AT**, we replace different percentages of the original offensive training set with perturbed text generated by our attacking method and retrain local victim models to improve the robustness. For **AGBR**, we recover the perturbed text by replacing the abnormal tokens,

Table 8: The accuracies of models classifying transfered adversarial examples generated on offensive text dataset. The transferability of the same model is meaningless, thus N/A is filled in the corresponding cells.

	Fasttext	TextCNN	BiLSTM	BERT	BaiduCensor	AliSecurity
Fasttext	N/A	0.500	0.428	0.768	0.180	0.420
TextCNN	0.528	N/A	0.406	0.750	0.176	0.434
BiLSTM	0.594	0.498	N/A	0.766	0.182	0.448
BERT	0.646	0.596	0.546	N/A	0.183	0.430
BaiduCensor	0.644	0.528	0.506	0.784	N/A	0.408
AliSecurity	0.728	0.704	0.592	0.776	0.430	N/A

which exist in the associative graph but out of vocabulary, with the normal word in the input stream, where we define the normal word as the word which is accessible to the abnormal token in the associative graph and do not out of vocabulary. Next, we report the shielding performance of the methods above.

AT. Fig. 2 (left) illustrates the results of **AT** and we can see the accuracies of all models improve immediately when **AT** starts with 10% perturbed offensive examples. The higher the percentage of perturbed text added to training set, the higher the accuracies of the models increase classifying perturbed offensive text. However, it is somewhat strange that the trade-off between robustness and accuracy is not shown in Fig. 2 (right), i.e., the accuracies of models classifying clean and perturbed text both increase, which is opposite to previous literature Tsipras et al. (2019).

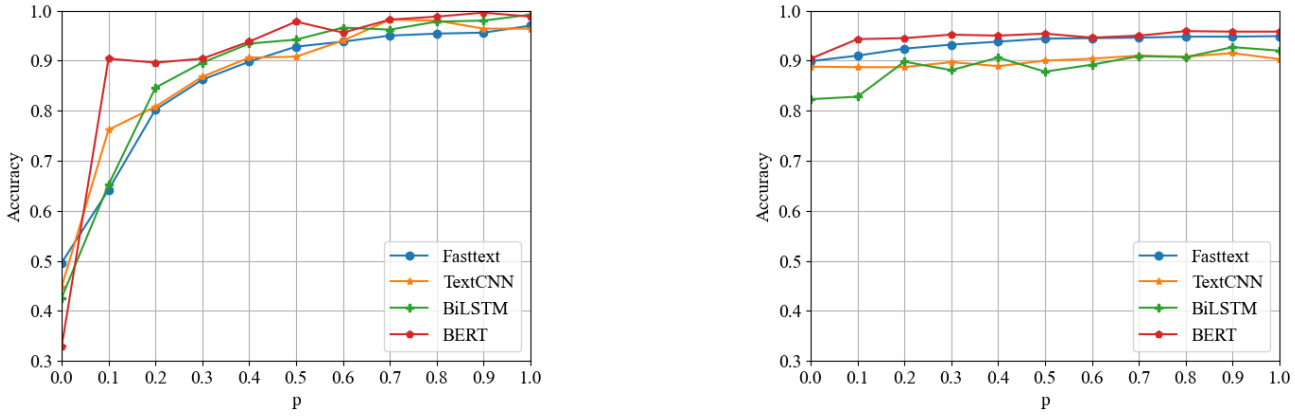

Figure 2: Accuracies of models retrained with different percentages (denoted as p) of perturbed offensive text replacing the original offensive text in training set. The figure left and right illustrate the change of accuracies when models classify perturbed offensive text and all clean text separately.

AGBR. As shown as Table 9, the accuracies of victims classifying adversarial examples, i.e., perturbed offensive text, improve significantly when we shield victims using **AGBR**, although the accuracies of most victims decreased slightly when classifying all clean test set (offensive and non-offensive text). Interestingly, the accuracies of BiLSTM and BERT classifying all clean test set increase rather than decrease when we shield victims using **AGBR**.

Table 9: The changes of accuracies in different conditions using **AGBR**. ΔADV and ΔALL refer to the changes of accuracies classifying adversarial examples (perturbed offensive text) and all clean test set (offensive and non-offensive) respectively.

Models	**ΔADV**	**ΔALL**
Fasttext	0.378	-0.001
TextCNN	0.448	-0.005
BiLSTM	0.437	0.003
BERT	0.568	0.002
BaiduCensor	0.570	-0.003
AliSecurity	0.452	-0.013

6. Conclusion

In this work, we propose a chain association-based perturbation approach, which is inspired by the strong association ability of humans, to attack Chinese NLP system. We reveal the vulnerability of the state-of-the-art NLP models and industrial-leading applications to our attack and show that human are able to understand the perturbed text with their strong association ability, showing that adversarial attack based on chain association can cause serious impact. We also explore methods to shield systems from chain association-based attack and show the effectiveness of associative knowledge graph in shielding such attack. Our work shows that gaps between humans and machines exist in reading comprehension while humans are able to associate things that seem totally different but related, and we hope that our work can inspire others to investigate more attacking and shielding technologies combining traits of the human thinking.

Acknowledgments

This work is supported by the Key Cooperation Project of Chongqing Municipal Education Commission (No. HZ2021008).

References

Eric Bracken, Brendon Billings, Maria Barnes, and Muhammad Spocter. *Encyclopedia of Evolutionary Psychological Science*, chapter Associationism, pages 404–415. Springer International Publishing, 04 2021. ISBN 978-3-319-19650-3.

Nuo Cheng, Guoqin Chang, Haichang Gao, Ge Pei, and Yang Zhang. Wordchange: Adversarial examples generation approach for chinese text classification. *IEEE Access*, 8: 79561–79572, 2020. doi: 10.1109/ACCESS.2020.2988786. URL https://doi.org/10.1109/ACCESS.2020.2988786.

Brian Formento, Chuan Sheng Foo, Luu Anh Tuan, and See Kiong Ng. Using punctuation as an adversarial attack on deep learning-based NLP systems: An empirical study. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1–34, Dubrovnik, Croatia, 2023. Association for Computational Linguistics.

Bao Guo, Chunxia Zhang, Junmin Liu, and Xiaoyi Ma. Improving text classification with weighted word embeddings via a multi-channel textcnn model. *Neurocomputing*, 363:366–374, 2019. doi: 10.1016/J.NEUCOM.2019.07.052. URL https://doi.org/10.1016/j.neucom.2019.07.052.

Wenjuan Han, Liwen Zhang, Yong Jiang, and Kewei Tu. Adversarial attack and defense of structured prediction models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2327–2338. Association for Computational Linguistics, 2020.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press, 2020.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2068.

James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, volume 5, pages 4104–4108, Orlando, FL, USA, 1997. IEEE.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*, San Diego, California, USA, 2019. The Internet Society. doi: 10.14722/ndss.2019.23138. URL http://dx.doi.org/10.14722/ndss.2019.23138.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-*

19, 2018, Stockholm, Sweden, pages 4208–4215, Stockholm, Sweden, 2018. IJCAI. doi: 10.24963/ijcai.2018/585. URL https://doi.org/10.24963/ijcai.2018/585.

Hongxu Ou, Long Yu, Shengwei Tian, and Xin Chen. Chinese adversarial examples generation approach with multi-strategy based on semantic. *Knowl. Inf. Syst.*, 64(4):1101–1119, 2022. doi: 10.1007/S10115-022-01652-1.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 1085–1097. Association for Computational Linguistics, 2019.

Hongying Si and Xianyong Wei. Sentiment analysis of social network comment text based on LSTM and bert. *J. Circuits Syst. Comput.*, 32(17):2350292:1–2350292:14, 2023. doi: 10.1142/S0218126623502924. URL https://doi.org/10.1142/S0218126623502924.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019. OpenReview.net. URL https://openreview.net/forum?id=SyxAb30cY7.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *IEEE Data Eng. Bull.*, 47(1):48–62, 2024.

Li Xiaoyan and Rodolfo C. Raga. Bilstm model with attention mechanism for sentiment classification on chinese mixed text comments. *IEEE Access*, 11:26199–26210, 2023. doi: 10.1109/ACCESS.2023.3255990. URL https://doi.org/10.1109/ACCESS.2023.3255990.

Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. Grey-box adversarial attack and defence for sentiment classification. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 2021. Association for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6066–6080, Online, 2020. Association for Computational Linguistics.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5564–5569. Association for Computational Linguistics, July 2019.

Appendix A. Details on Prompts

We list all prompts used in this study in Table 10.

Table 10: All prompts used in this study.

Dataset	Prompt
Meituan / Amazon Comments	Now you are a text classification model. No matter what I input, please classify the following text as positive or negative emotions. Note that you only need to reply to one word in "positive" or "negative" , and do not reply other words. (in Chinese)
Offensive Text	Now you are a text classification model. No matter what I input, please classify the following text as positive or negative emotions. Note that you only need to reply to one word in "offensive" or "non-offensive", and do not reply other words. (in Chinese)

Appendix B. Case study

We display some adversarial examples generated by the baselines and ours on Meituan comments in Table 11. The examples of offensive text are not shown since there are many indecent words.

Table 11: Adversarial examples generated by baselines and ours on Meituan comments.

Meituan Comments Examples
Original Input (Prediction = **Positive**)
今天快(kuài)递员的速度比较快(kuài)，服务也好(hǎo)，辛苦(xīn kǔ)了！ (The courier is fast today and the service is also good, he has worked hard!)
GreedyAttack (Prediction = **Negative**)
今天快(kuài)递员的速度比较块(kuài)，服务也hao，辜(kù)酷了！
WordChange (Prediction = **Negative**)
今天快(kuài)递员的速度比较脍(kuài)，服务也耗(hào)，辛」苦了！
Ours (Prediction = **Negative**)
今天快递员的速度比较发(fā)思(sī)特(tè)，服务也女子，刑(xíng)苦了！