Machine Learning (2024) 113:6341–6371
https://doi.org/10.1007/s10994-024-06572-5

Kalt: generating adversarial explainable chinese legal texts

Yunting Zhang[1] · Shang Li[1] · Lin Ye[1] · Hongli Zhang[1] · Zhe Chen[1] · Binxing Fang[1]

Received: 21 June 2023 / Revised: 11 March 2024 / Accepted: 28 May 2024 /
Published online: 21 June 2024

Abstract
Deep neural networks (DNNs) are vulnerable to adversarial examples (AEs), which are well-designed input samples with imperceptible perturbations. Existing methods generate AEs to evaluate the robustness of DNN-based natural language processing models. However, the AE attack performance significantly degrades in some verticals, such as law, due to overlooking essential domain knowledge. To generate explainable Chinese legal adversarial texts, we introduce legal knowledge and propose a novel black-box approach, knowledge-aware law tricker (KALT), in the framework of adversarial text generation based on word importance. Firstly, we invent a legal knowledge extraction method based on KeyBERT. The knowledge contains unique features from each category and shared features among different categories. Additionally, we design two perturbation strategies, Strengthen Similar Label and Weaken Original Label, to selectively perturb the two types of features, which can significantly reduce the classification accuracy of the target model. These two perturbation strategies can be regarded as components, which can be conveniently integrated into any perturbation method to enhance attack performance. Furthermore, we propose a strong hybrid perturbation method to introduce perturbation into the original texts. The perturbation method combines seven representative perturbation methods for Chinese. Finally, we design a formula to calculate interpretability scores, quantifying the interpretability of adversarial text generation methods. Experimental results demonstrate that KALT can effectively generate explainable Chinese legal adversarial texts that can be misclassified with high confidence and achieve excellent attack performance against the powerful Chinese BERT.

Keywords Adversarial example · Deep neural networks · Chinese text classification · Legal knowledge · Perturbation

Editor: Lijun Zhang.

✉ Lin Ye
hityelin@hit.edu.cn

[1] School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China

1 Introduction

Deep neural networks (DNNs) are widely applied in various fields, such as natural language processing (NLP) (Devlin et al., 2019; Liu et al., 2019), computer vision (CV) (Zeiler & Fergus, 2014; Jin et al., 2021), and cyber security (Wu et al., 2023; Luo et al., 2023). However, recent studies (Szegedy et al., 2014; Goodfellow et al., 2015) have found that DNNs are vulnerable in the face of adversary attacks. Adversarial examples (AEs) are well-designed input samples with imperceptible perturbations, which can confuse DNNs. Research on AE generation methods can grasp the weaknesses of current mainstream models and lay the foundation for designing corresponding defense measures and robustness assessment methods for deep learning (DL) models (Chen et al., 2023).

The majority of current research on AEs is focused on the field of CV (Szegedy et al., 2014; Goodfellow et al., 2015). However, Gao et al. (2018), Alzantot et al. (2018), Li et al. (2020), Chen et al. (2022) show that DNNs are also vulnerable to AEs in NLP. Unlike continuous image data, text is discrete, rendering the AE generation methods used in the CV field not directly transferable to the domain of NLP. Existing token-level adversarial text generation methods are typically designed in a framework based on word importance (Gao et al., 2018; Jin et al., 2020; Zhang et al., 2023). The framework includes two stages: the ranking stage and the perturbation stage. In the ranking stage, we calculate word importance scores and sort the words in descending order based on the scores. In the perturbation stage, we introduce perturbations sequentially into the important words. Based on various word importance scoring methods (Xu & Du, 2020; Gao et al., 2018; Wang et al., 2019) and perturbation methods (Li et al., 2019; Jin et al., 2020; Garg & Ramakrishnan, 2020; Li et al., 2020, 2021) proposed for English texts, the work of Wang et al. (2019), Zhang et al. (2020), Cheng et al. (2020), Zhang et al. (2023) designs adversarial text generation methods for Chinese texts, taking into account the linguistic characteristics of Chinese.

However, relatively few studies on verticals are available. In recent years, the development of smart justice has increasingly popularized the application of artificial intelligence in the judicial domain. Charge classification is a fundamental and core task in smart justice. Recent studies have shown that this task can be accomplished through DL models (Li et al., 2019, 2020), significantly enhancing the work efficiency of judges and lawyers. However, the susceptibility of DNNs to legal adversarial text attacks may lead to severe consequences such as misjudgments. Research on adversarial text generation in the legal domain can provide AEs for adversarial training, thereby enhancing the robustness of the target model through retraining. Consequently, there is an urgent need for research on legal adversarial text generation methods. Adversarial text generation methods designed for common domains often overlook domain knowledge, resulting in generated adversarial texts that considerably differ in meaning from the original texts. Therefore, we need to devise an adversarial text generation method that produces adversarial texts that are understandable to humans in their original intent but trigger errors in DL models.

To address this issue, we propose the Knowledge-Aware Law Tricker (KALT) in the frame of adversarial text generation based on word importance. KALT employs legal knowledge to improve the generation of Chinese adversarial texts and advance the attack to DL models of charge classification. Firstly, we invent a knowledge extraction method based on KeyBERT (Grootendorst, 2020). After extracting the legal knowledge of each label, we apply the extracted knowledge to the perturbation stage. In addition, we design two perturbation strategies: Strengthen Similar Label (SSL) and Weaken Original Label (WOL). The SSL strategy aims to strengthen the unique features of the similar category. Meanwhile, the WOL strategy aims to weaken the unique features of the original category so that the shared features take effect. These two perturbation strategies can be integrated into any adversarial text generation method based on word importance designed for common domains, endowing KALT with high scalability. These strategies also render the adversarial text generation process interpretable, facilitating human understanding of the original intent of the text. Furthermore, we propose a novel hybrid perturbation method named Hybrid-7, which includes seven Chinese perturbation methods. The seven perturbation methods are Shuffle, Splitting-Character (SC), Tradition, Pinyin, Synonyms, Word Embedding, and BERT-MLM. Hybrid-7 introduces various perturbations into the original texts and significantly reduces the classification accuracy of the target model. Finally, to more intuitively demonstrate the interpretability of the proposed KALT, we introduce a formula for calculating interpretability scores to quantify interpretability.

Figure 1 presents an example of a Chinese legal adversarial text generated with KALT. This adversarial text incorporates legal knowledge into the generation process by replacing the keywords "knife" and "death" in the charge of intentional homicide with their synonyms "blade" and "perish", which are not in the predefined vocabulary. At this point, the keyword "slightly injured" in the charge of intentional injury would cause the target model to misclassify the adversarial text as intentional injury. At the same time, humans can still understand the original intent of the text. It demonstrates that introducing legal knowledge makes the adversarial text generation process interpretable, misleading deep learning models without affecting human understanding of its original intent.

We have conducted experiments on a real-world law dataset called CAIL (Xiao et al., 2018), containing 2,676,075 criminal cases in Mainland China. The experimental results demonstrate that KALT generates effective and readable adversarial texts that can

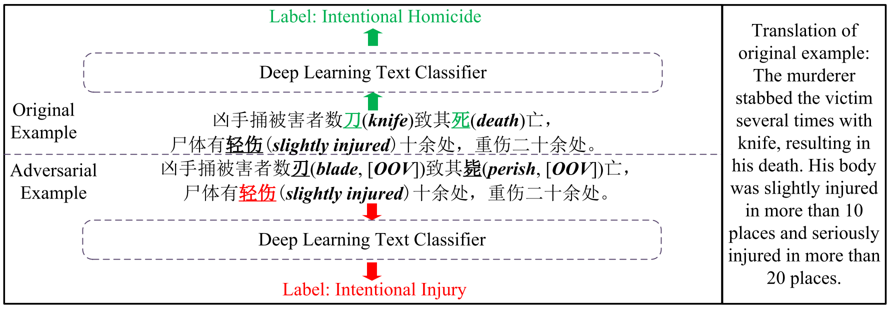

Label: Intentional Homicide
Deep Learning Text Classifier
Original Example
凶手捅被害者数刀(knife)致其死(death)亡，
尸体有轻伤(slightly injured)十余处，重伤二十余处。
Adversarial Example
凶手捅被害者数刃(blade, [OOV])致其毙(perish, [OOV])亡，
尸体有轻伤(slightly injured)十余处，重伤二十余处。
Deep Learning Text Classifier
Label: Intentional Injury
Translation of original example: The murderer stabbed the victim several times with knife, resulting in his death. His body was slightly injured in more than 10 places and seriously injured in more than 20 places.

Fig. 1 An example of an AE attack on the Chinese legal text classification model: after the knowledge-aware perturbation, the case of intentional homicide is misclassified as intentional injury by the DL classifier. Meanwhile, humans can understand the original intent of the adversarial text. "OOV" means out of the predefined vocabulary

significantly reduce the performance of the powerful Chinese BERT model with good interpretability.

Our contributions of this work are summarized as follows:

(1) We propose an innovative knowledge-aware Chinese adversarial text generation method named KALT for the charge classification task. We design a KeyBERT-based legal knowledge extraction method and employ domain knowledge to advance Chinese legal adversarial text generation.
(2) We propose two perturbation strategies, SSL and WOL, which are beneficial for selecting meaningful keywords in the context of law and constructing effective and interpretable adversarial texts. These two strategies can be incorporated as components into any adversarial text generation method based on word importance, enhancing the scalability and flexibility of KALT. In addition, we introduce a novel hybrid perturbation method that combines seven perturbation methods for Chinese. Finally, we devise a formula for calculating interpretability scores to quantify the interpretability of adversarial text generation methods.
(3) We have performed experiments on a real-world dataset of Chinese criminal cases. The powerful pre-trained Chinese BERT is attacked as a target model with KALT. The results show the effectiveness of KALT in the charge classification task, and adversarial texts generated by KALT can deceive the target model with high confidence.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes the problem formalization and threat model. Section 4 proposes the overall KALT framework and describes the details of the different perturbation methods and two perturbation strategies. Section 5 presents the experimental results and analyses. Finally, Sect. 6 contains the concluding remarks.

2 Related work

In this section, we provide a brief review of the adversarial text generation methods based on word importance. We describe the approaches adopted in the ranking stage and the perturbation stage, respectively.

In the ranking stage, various word importance scoring methods are proposed. Gao et al. (2018) propose four scoring methods: Temporal Score, Temporal Tail Score, Combined Score, and Delete Score (DS). The first three methods are applied to recurrent neural networks (RNNs), while DS is a universal method for all models. On this basis, Wang et al. (2019) utilize two of the four scoring methods and improve them by introducing the TF-IDF score. Jin et al. (2020) incorporate the label changes before and after the deletion of words to improve the DS method. Xu and Du (2020) transfer a method for image data, layer-wise relevance propagation, to the ranking stage to calculate word importance scores.

In the perturbation stage, various perturbation methods are designed for different languages. Li et al. (2019) propose five perturbation methods to generate English adversarial texts. These methods include Insert, Delete, Swap, Substitute-C, and

Substitute-W. The first four methods consist of character-level operations, including insertion, deletion, swapping, and replacement, while the last method involves word-level substitution in the word embedding space. BAE, BERT-ATTACK, and CLARE (Garg & Ramakrishnan, 2020; Li et al., 2020, 2021) employ the masked language model (MLM) in BERT to introduce perturbation into English texts. These methods generate contextually appropriate words to replace the original words. In addition to the perturbation methods for English texts, there are also perturbation methods tailored for Chinese texts. Zhang et al. (2020) propose five perturbation methods for Chinese by transferring those for English according to the work in Li et al. (2019). These methods consist of Synonyms, Shuffle, SC, Glyph, and Pinyin. The first two methods are transferred from Substitute-W and Swap. The remaining three methods are unique methods for Chinese according to the linguistic characteristics. Among them, SC splits a Chinese character into radicals, while Glyph and Pinyin perturb the original Chinese character and word based on similar appearance and pronunciation, respectively. Based on Zhang et al. (2020), Cheng et al. (2020) propose a new perturbation method, and the method involves inserting special characters into words. Based on Zhang et al. (2020) and Cheng et al. (2020), Tong et al. (2020) propose two novel perturbation methods: replacing simplified characters with traditional characters and substituting all original Chinese characters with their pinyin equivalents in a word. Ou et al. (2022) improve the pinyin rewriting in Tong et al. (2020) and introduce a multi-strategy to combine five perturbation methods, which include Synonyms, Glyph, Pinyin, special character insertion, and pinyin rewriting. Zhang et al. (2023) improve the perturbation methods based on BERT-MLM (Garg & Ramakrishnan, 2020; Li et al., 2020, 2021) by considering the characteristics of word length in Chinese, and they successfully adapt it to Chinese texts.

In summary, although previous methods have contributed to textual adversarial attacks, few studies have explored adversarial text generation methods in the legal domain. Our work aims to fill this gap. For this purpose, we propose KALT, which introduces the legal knowledge to generate explainable Chinese legal adversarial texts. Compared with other methods, KALT attacks Chinese legal text classifiers more effectively and can fool the target model with higher confidence.

3 Problem definition

In this section, we formalize adversarial texts and the knowledge-adding process. Subsequently, a threat model setting is described.

3.1 Problem formulation

Consider a set of n documents $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, with each document associated with one label from a label set $\mathbb{Y} = \{y_1, y_2, \dots, y_m\}$, where m represents the total number of labels. We have a text classification model $F : \mathbb{X} \rightarrow \mathbb{Y}$ that maps the input space $\mathbb{X}$ to the output space $\mathbb{Y}$. In a document $\mathbf{x}_i \in \mathbb{X}$, we can generate the corresponding adversarial text $\mathbf{x}'_i$ by adding perturbation $\Delta\mathbf{x}_i$ into $\mathbf{x}_i$. We introduce a function H, which can measure the gap in similarity between $\mathbf{x}_i$ and $\mathbf{x}_i'$. A successful adversarial text should satisfy the constraint conditions as follows:

$$\begin{gathered} F(\mathbf{x}_i) \neq F(\mathbf{x}_i') \\ \text{s.t. } H(\mathbf{x}_i, \mathbf{x}_i') \leq \varepsilon \end{gathered}, \tag{1}$$

where ε is the upper bound of the difference between $\mathbf{x}_i$ and $\mathbf{x}_i'$.

We introduce legal knowledge in the perturbation stage. The generation process of the adversarial text $\mathbf{x}_i'$ can be formally expressed as follows:

$$\mathbf{x}_i' = K(\mathbf{x}_i + \Delta\mathbf{x}_i), \tag{2}$$

where $K(\cdot)$ represents the perturbation strategy that leverages legal knowledge.

3.2 Threat model

In this work, we consider that the attack occurs in the black-box setting. In such a setting, the adversary does not have access to any specific details inside the model, such as the model structure and the weight of each neuron. The adversary is only allowed to query the model with a meticulously crafted input and obtain the output that consists of labels and the corresponding confidence scores.

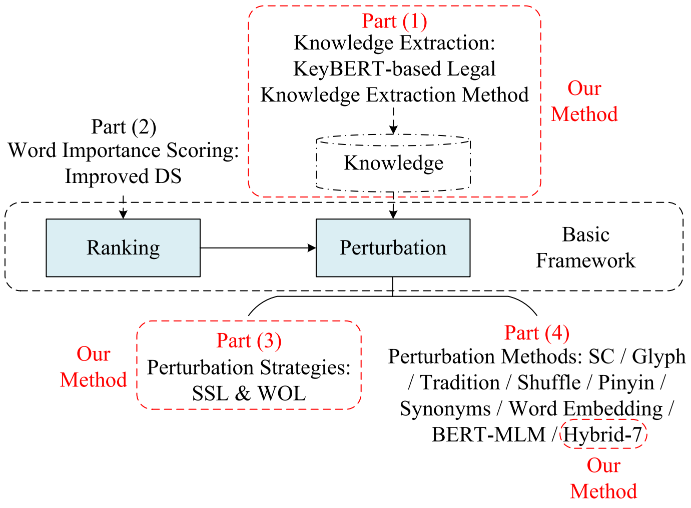


Fig. 2 The overview of KALT: KALT consists of four components, which include knowledge extraction, word importance scoring, perturbation strategies, and perturbation methods. Among these, the aspects of knowledge extraction, perturbation strategies, and the Hybrid-7 perturbation method are unique to KALT, distinguishing it from other adversarial text generation methods

4 Method

In this section, we describe the proposed KALT in detail. Figure 2 shows an overview of KALT according to the basic framework of adversarial text generation based on word importance. The framework consists of the ranking stage and the perturbation stage. Before the ranking stage, we extract legal knowledge, adopting a method based on KeyBERT (Grootendorst, 2020). In the ranking stage, the improved DS (Jin et al., 2020) is applied to calculate word importance scores. In the perturbation stage, the extracted knowledge is utilized to implement the two perturbation strategies, SSL and WOL. In the adversarial text generation framework based on word importance, these two perturbation strategies can be integrated into any perturbation method. Our work

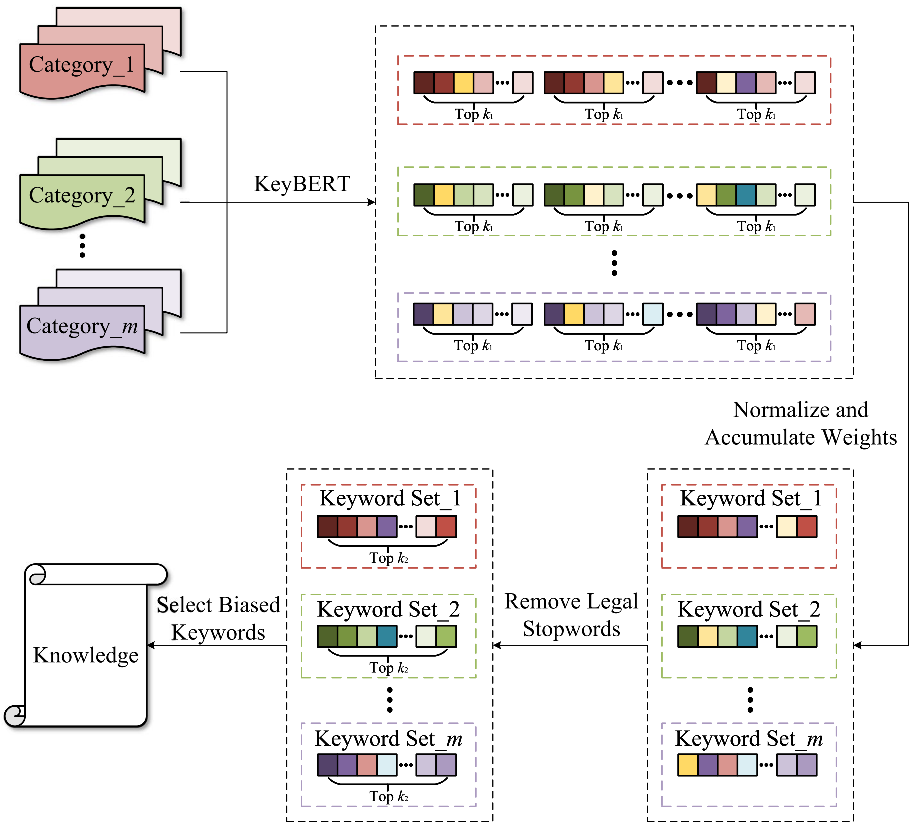


Fig. 3 The process of knowledge extraction: KeyBERT (Grootendorst, 2020) is adopted to extract the top k_1 keywords of a legal text in the dataset. During the extraction process, we apply linear normalization to standardize the weights of keywords extracted from each text. The weights of the identical keyword across different texts in the same category are accumulated to obtain the weight of the keyword in the category. After removing legal stopwords, we choose the top k_2 keywords from each category. Among these keywords, we select some biased words. These top k_2 keywords and the biased words are regarded as the knowledge of each category

primarily employs the nine perturbation methods, including our proposed Hybrid-7, displayed in Fig. 2.

The details of knowledge extraction, word importance scoring, perturbation strategies, and perturbation methods will be elucidated in Sects. 4.1–4.4, respectively.

4.1 Knowledge extraction

We design a method based on KeyBERT (Grootendorst, 2020) to extract knowledge from the dataset. KeyBERT is one of the most outstanding methods for keyword extraction. Unlike keyword extraction methods based on the bag-of-words model, DL-based KeyBERT leverages BERT embeddings and cosine similarity to identify sub-phrases most similar to the original text. This approach ingeniously transforms the task of keyword extraction into one of text similarity calculation. Relying on the powerful BERT model, we can extract keywords more accurately. Figure 3 displays the process of knowledge extraction.

As illustrated in Fig. 3, there are three important steps during the knowledge extraction process: normalizing weights, removing legal stopwords, and selecting biased keywords. The aims of these three steps are sequentially introduced below.

KeyBERT employs cosine similarity to measure the resemblance between sub-phrases and the original text, assigning a weight in the range $(-1, 1)$ for each word. In order to identify representative keywords for each category, it is necessary to eliminate words with negative weights and map the weights of keywords from a single article to all texts in the category. During the process, weight standardization is required. Consequently, the application of linear normalization facilitates the subsequent accumulation of word weights in the category.

Legal stopwords are the words that appear in most legal texts and contain little critical information, such as “court” and “happen”. These words may interfere with the effect of KALT to some extent. Therefore, we need to remove these words to obtain more accurate knowledge.

The selection of biased words facilitates the implementation of subsequent perturbation strategies. We carry out the SSL or WOL strategy in the perturbation stage. Different perturbation strategies correspond to different types of biased words, and the selection of biased words is related to the extracted knowledge. The knowledge that we obtain from each category contains shared and unique keywords. The former are keywords present in the original category and its similar category. In contrast, the latter are other keywords excluding the shared keywords in the original category or the similar category. SSL biased words typically appear in pairs and are selected from the unique keywords of two similar categories. However, not all unique keywords of a category can serve as SSL biased words. The selection criteria will be detailed in Sect. 4.1.3. WOL biased words are shared keywords that lean toward one of two similar categories. For example, though “slightly injured" is a shared keyword of the charges of intentional injury and intentional homicide, it occurs much more frequently in the charge of intentional injury than in the charge of intentional homicide. Therefore, “slightly injured” is more representative of the charge of intentional injury, and it is a biased keyword for the charge of intentional injury. The details regarding the selection of WOL biased words are described in Sect. 4.1.4.

Algorithm 1 exhibits details of the knowledge extraction process and the components of $\mathbb{S}$. In this work, N is set to 500, γ is set to 1.0, and both k_1 and k_2 are set to 50. The parameters N and γ will be elaborated on in Sects. 4.1.2 and 4.1.4, respectively.

```
Algorithm 1: Knowledge Extraction Process
Input: the text set $\mathbb{X}$, the universal stopword set $\mathbb{U}$, the number of categories
       $m$, the parameter introduced in the process of selecting legal
       stopwords $N$, the parameter introduced in the process of selecting
       WOL biased words $\gamma$, the number of keywords select in each text $k_1$,
       the number of keywords select in each category $k_2$
Output: the knowledge set $\mathbb{S}$

Step 1: Implement word segmentation and remove the stopwords in $\mathbb{U}$;
Step 2: Utilize KeyBERT to extract the top $k_1$ keywords from each text;
Step 3: Apply linear normalization to standardize weights and accumulate
  weights of the same keyword in different texts of a category, obtaining the
  original knowledge set $\mathbb{S}$;
Step 4: Select legal stopwords and remove them from $\mathbb{S}$;
Step 5: Select the top $k_2$ keywords and update $\mathbb{S}$;
Step 6: For each category, find its similar category which has the most
  shared keywords with it and incorporate the information into $\mathbb{S}$;
Step 7: For each category and its similar category, select their shared and
  unique keywords and incorporate the information into $\mathbb{S}$;
Step 8: Select SSL biased words and incorporate the information into $\mathbb{S}$;
Step 9: Select WOL biased words and incorporate the information into $\mathbb{S}$.

return $\mathbb{S}$
```

In Algorithm 1, Steps 3, 4, 8, and 9 are critical, as they determine the quality of the knowledge set $\mathbb{S}$. We sequentially introduce the details of these steps in Sects. 4.1.1–4.1.4.

4.1.1 Weight normalization

As mentioned above, after conducting preliminary keyword extraction on each text with KeyBERT, it is necessary to perform linear normalization on the weights to facilitate the accumulation of weights for the keywords at the category level. For each document, we extract the top k_1 words with the highest weights, where the range of the weights of these words is (-1, 1). If the extracted keywords have negative weights, these words are removed. Subsequently, the weights of the remaining words are standardized, employing linear normalization. Assuming that in a given document, the extracted words with positive weights $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p$ ($p \leq k_1$) with their corresponding weights given by $\alpha_1, \alpha_2, \ldots, \alpha_p$, the normalized weight $\overline{\alpha}_i$ of the word $\mathbf{w}_i$ can be represented by the following formula:

$$\overline{\alpha}_i = \frac{\alpha_i}{\sum_{l=0}^{p} \alpha_l}. \tag{3}$$

We extend the calculation of keyword weights from the document level to the category level. Assume there is a keyword $\mathbf{w}_i$ in a category. For a document $\mathbf{x}_j$, the weight of $\mathbf{w}_i$ can be denoted as β_{ij}. We introduce $L_j(\mathbf{w}_i)$ to denote the position of $\mathbf{w}_i$ in $\mathbf{x}_j$. The weight β_{ij} can be calculated using the following formula:

$$\beta_{ij} = \begin{cases} \overline{\alpha}_{hj}, & \text{if } h = L_j(\mathbf{w}_i) \wedge \mathbf{w}_i \in \overline{\mathbf{x}}_j \\ 0, & \text{if } \mathbf{w}_i \notin \overline{\mathbf{x}}_j \end{cases}, \tag{4}$$

where $\overline{\mathbf{x}}_j$ represents the set of words with positive weights in $\mathbf{x}_j$. Notably, due to the extension of weights from the document level to the category level, the weight $\overline{\alpha}_i$ in Eq. 3 has acquired an additional dimension in Eq. 4.

The weight of $\mathbf{w}_i$ in the category, denoted as β_i, can be determined by the following formula:

$$\beta_i = \sum_{j=0}^{q} \beta_{ij}, \tag{5}$$

where q represents the number of documents in the category.

4.1.2 Legal stopword selection

In the process of legal stopword selection, we introduce a new parameter, N, which should satisfy the following condition:

$$N \geq 2k_2, \tag{6}$$

where k_2 is the number of selected keywords for each category. We extract the top N keywords with the highest weights of each category to build a keyword set $\mathbb{S}'$ ($|\mathbb{S}'| = m \times N$). In the set $\mathbb{S}'$, if a keyword of a category is also in all the remaining categories, we regard the keyword as a legal stopword. The details of the process are exhibited in Algorithm 2.

Algorithm 2: Legal Stopword Selection

```
Input: the original knowledge set $\mathbb{S}$ output in Step 3 of Algorithm 1, the
       number of categories $m$, the parameter $N$, the number of keywords
       select in each category $k_2$
Output: the legal stopword set $\mathbb{L}$
1  $\mathbb{L} \leftarrow \varnothing$; // legal stopwords
2  $\mathbb{S}' \leftarrow$ get_keywords($\mathbb{S}$, $\mathbb{L}$, $N$); // $N \geq 2k_2$ and $|\mathbb{S}'| = m \times N$
3  for $current\_c \in \mathbb{S}'$ do
4      for $current\_k \in current\_c$ do
5          flag $\leftarrow$ Ture;
6          for $other\_c \in \mathbb{S}'$ do
7              if $current\_c = other\_c$ then
8                  continue;
9              end
10             if $current\_k \notin other\_c$ then
11                 flag $\leftarrow$ False;
12                 break;
13             end
14         end
15         if $flag\ is\ True$ then
16             $\mathbb{L} \leftarrow \mathbb{L} \cup$ {current_k};
17         end
18     end
19 end
20 return $\mathbb{L}$
```

4.1.3 SSL biased word selection

As mentioned above, SSL biased words typically appear in pairs. When we extract a pair of SSL biased words from two similar categories that are suitable for the SSL perturbation strategy, the pair of SSL biased words should satisfy the following conditions: (1) The meaning of the two words should be similar; (2) The part of speech of the two words should be consistent.

Based on the above two conditions, we first extract word pairs with an edit distance of 1 from the unique keywords of the two similar categories. Subsequently, we filter out word pairs with different parts of speech. Finally, a simple manual selection is conducted to obtain the final pairs of SSL biased words.

In addition, to ensure that the word pairs we extract have similar meanings, we utilize a million law data to train a Word2Vec (Mikolov et al., 2013) model and calculate the similarities between these selected word pairs on the model. Among the 1,481,300 words, the similarities of the chosen word pairs almost rank in the top 20.

4.1.4 WOL biased word selection

We select WOL biased words based on weights that are normalized and accumulated. We introduce a new parameter, γ, to measure the bias of a keyword. For a shared keyword $\mathbf{w}_{sk}$ of category y_a and its similar category y_b, β_a and β_b denote the weights of $\mathbf{w}_{sk}$ in y_a and y_b, respectively. If β_a and β_b satisfy:

$$\beta_a / \beta_b \;\; \gamma, \tag{7}$$

we regard $\mathbf{w}_{sk}$ as a biased word of y_a. Algorithm 3 displays the details of the process.

Algorithm 3: WOL Biased Word Selection

```
Input: the knowledge set $\mathbb{S}$ output in Step 8 of Algorithm 1, the bias
       coefficient $\gamma$
Output: the final knowledge set $\mathbb{S}$
$\mathbb{C} \leftarrow \varnothing$; // WOL biased words of all categories
for $c\_s \in \mathbb{S}$ do
    current_c ← c_s.current_c;
    similar_c ← c_s.similar_c;
    $\mathbb{D} \leftarrow \varnothing$; // WOL biased words of each category
    for $current\_kw \in current\_c$ do
        for $similar\_kw \in similar\_c$ do
            if $current\_kw.word = similar\_kw.word$ then
                if $current\_kw.weight \,/\, similar\_kw.weight > \gamma$ then
                    $\mathbb{D} \leftarrow \mathbb{D} \cup \{\text{current\_kw.word}\}$;
                end
            end
        end
    end
    $\mathbb{C} \leftarrow \mathbb{C} \cup \{\mathbb{D}\}$;
end
$\mathbb{S} \leftarrow$ update($\mathbb{S}$, $\mathbb{C}$);
return $\mathbb{S}$
```

4.2 Word importance scoring method

We employ the improved DS (Jin et al., 2020) to calculate word importance scores. The calculation process can be formalized as follows. In the context of the problem definition described in Sect. 3.1, for a given document $\mathbf{x}$, its predicted label is denoted as y. In this work, we assume that the predicted label is consistent with the ground truth. For a given label y', we represent the confidence of $\mathbf{x}$ on y' with $F_{y'}(\mathbf{x})$. We assume that $\mathbf{w}_i$ is a word in $\mathbf{x}$. The importance score of $\mathbf{w}_i$ can be calculated as follows:

$$s_{\mathbf{w}_i} = \begin{cases} F_y(\mathbf{x}) - F_y\left(\mathbf{x}\backslash\mathbf{w}_i\right), & \text{if } F(\mathbf{x}) = F\left(\mathbf{x}\backslash\mathbf{w}_i\right) = y \\ F_y(\mathbf{x}) - F_y\left(\mathbf{x}\backslash\mathbf{w}_i\right) + \left[F_{y'}\left(\mathbf{x}\backslash\mathbf{w}_i\right) - F_{y'}(\mathbf{x})\right], & \\ \text{if } F(\mathbf{x}) = y \wedge F\left(\mathbf{x}\backslash\mathbf{w}_i\right) = y' \wedge y \neq y' & \end{cases}, \tag{8}$$

where $\mathbf{x} \backslash \mathbf{w}_i$ denotes the removal of $\mathbf{w}_i$ from $\mathbf{x}$.

If $s_{\mathbf{w}_i} > 0$, then $\mathbf{w}_i$ has a positive effect on label y, and vice versa. We only select words whose importance scores are larger than zero. We arrange these words in descending order according to their importance scores to perturb them in order in the perturbation stage.

4.3 Perturbation strategy

In the perturbation stage, we provide two perturbation strategies: SSL and WOL. The SSL strategy aims to strengthen the unique features of the label, which are similar to the original one. We substitute the SSL biased words of the original label with the ones of its similar label to implement the SSL strategy. Meanwhile, the WOL strategy aims to weaken the unique features of the original label. To execute the WOL strategy, we replace the unique keywords of the original label with words that are out of the predefined vocabulary so that the shared features take effect. The replacement process employs various perturbation methods described in Sect. 4.4. In this work, if the shared keyword number of the original and another label is the highest, we regard the latter as the most similar label to the original one. Figure 4 demonstrates the process of the two perturbation strategies.

The selection of SSL and WOL biased words has been elaborately described in Sects. 4.1.3 and 4.1.4, respectively. Beyond the process described above, there are three details to note regarding the SSL and WOL strategies: (1) Because selecting SSL biased word pairs needs to meet relatively strict constraints, not all categories are suitable for the SSL strategy. In this work, the categories applicable to the SSL strategy are the charges of larceny, robbery, and forcible seizure. The WOL strategy is applicable to all categories, but its attack effectiveness is slightly inferior to that of the SSL strategy; (2) In addition to the WOL biased words, the words requiring perturbation in $\mathbb{SK}$ also include the most important words selected by the improved DS method in each text; (3) Some important words selected by the improved DS are sometimes outside the keyword set we extract in the knowledge extraction stage. Under these circumstances, we can ignore the two strategies and directly add perturbation into the selected word with perturbation methods.

The details of SSL and WOL strategies are shown in Algorithm 4. In Algorithm 4, we employ *fit_to_SSL*() to determine whether a label is applicable to SSL (**line 4**). We choose the SSL strategy if the label is applicable to it (**lines 6–16**); otherwise, we adopt the WOL strategy (**lines 19–30**). Various perturbation methods are applied to perturb words(**lines 13 and 27**), whose details are shown in Sect. 4.4.

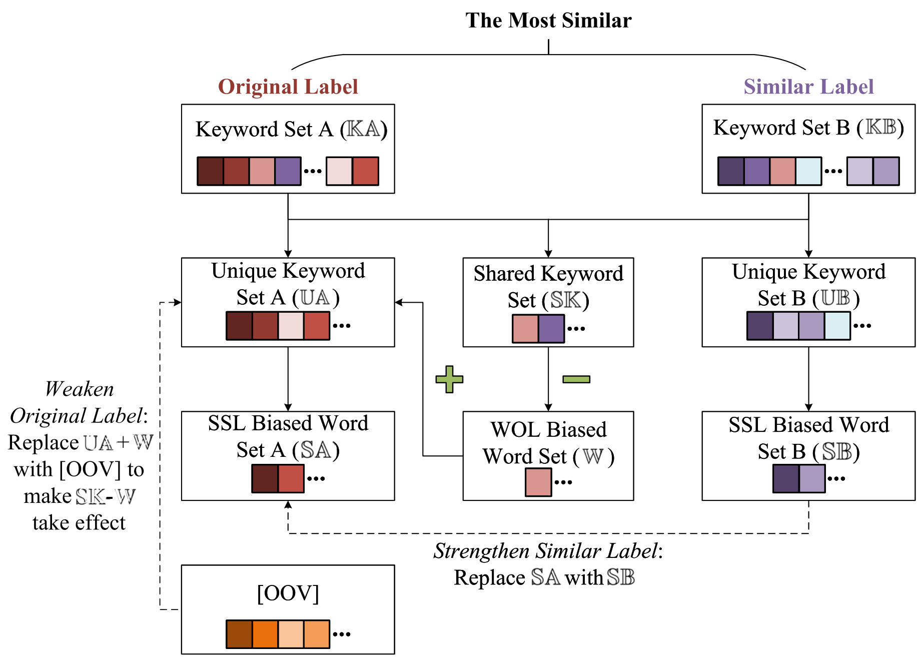

The Most Similar
Original Label
Similar Label
Keyword Set A ($\mathbb{KA}$)
Keyword Set B ($\mathbb{KB}$)
Unique Keyword Set A ($\mathbb{UA}$)
Shared Keyword Set ($\mathbb{SK}$)
Unique Keyword Set B ($\mathbb{UB}$)
SSL Biased Word Set A ($\mathbb{SA}$)
WOL Biased Word Set ($\mathbb{W}$)
SSL Biased Word Set B ($\mathbb{SB}$)
Weaken Original Label: Replace $\mathbb{UA}+\mathbb{W}$ with [OOV] to make $\mathbb{SK}$-$\mathbb{W}$ take effect
Strengthen Similar Label: Replace $\mathbb{SA}$ with $\mathbb{SB}$
[OOV]

Fig. 4 The process of the SSL and WOL strategies: for the SSL strategy, we replace the words in $\mathbb{SA}$ with the words in $\mathbb{SB}$. For the WOL strategies, we first select the WOL biased words from $\mathbb{SK}$. Subsequently, we substitute the words in $\mathbb{UA}$ and $\mathbb{W}$ with the words that are out of the predefined vocabulary. “OOV” means out of the predefined vocabulary

Algorithm 4: Perturbation Strategy

```
Input: the knowledge set $\mathbb{S}$, the original label $y_a$, the sorted word list
       using improved DS $\mathbb{Z}$, the number of perturbed words $num$
Output: the set of the original words and their corresponding
        perturbed words $\mathbb{O}$
1  $\mathbb{O} \leftarrow \varnothing$;
2  $\mathbb{Z} \leftarrow \mathbb{Z}[:num]$;
3  $\mathbb{A} \leftarrow \mathbb{S}$.get_knowledge($y_a$);
4  if fit_to_SSL(y) then
5      /* SSL strategy */
6      $\mathbb{SA}$, $\mathbb{SB} \leftarrow \mathbb{A}$.biased_words_SSL();
7      for old_word $\in \mathbb{Z}$ do
8          ptb.old $\leftarrow$ old_word;
9          if old_word $\in \mathbb{SA}$ then
10             ptb.new $\leftarrow$ edit_distance(old_word, $\mathbb{SB}$);
11             $\mathbb{O} \leftarrow \mathbb{O} \cup \{$ptb$\}$;
12         else
13             ptb.new $\leftarrow$ transformation(old_word);
14             $\mathbb{O} \leftarrow \mathbb{O} \cup \{$ptb$\}$;
15         end
16     end
17 else
18     /* WOL strategy */
19     $\mathbb{SK} \leftarrow \mathbb{A}$.shared_keys();
20     $\mathbb{W} \leftarrow \mathbb{A}$.biased_words_WOL();
21     for old_word $\in \mathbb{Z}$ do
22         $\mathbb{SP} \leftarrow \mathbb{W} \cup \{\mathbb{Z}[0]\}$;
23         if old_word $\in \mathbb{SK}$ - $\mathbb{SP}$ then
24             continue;
25         else
26             ptb.old $\leftarrow$ old_word;
27             ptb.new $\leftarrow$ transformation(old_word);
28             $\mathbb{O} \leftarrow \mathbb{O} \cup \{$ptb$\}$;
29         end
30     end
31 end
32 return $\mathbb{O}$
```

Figures 5 and 6 display two examples of the SSL and WOL strategies, respectively.

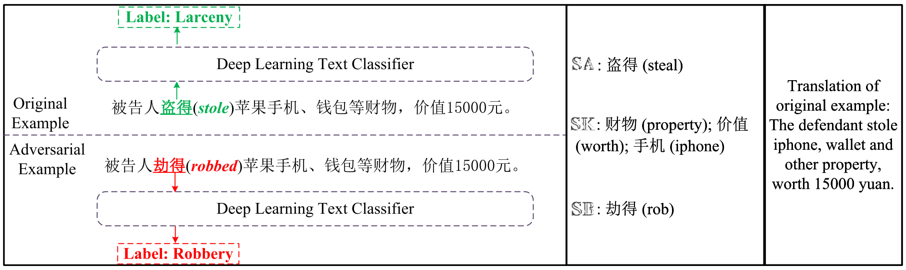

Label: Larceny
Deep Learning Text Classifier
Original Example
被告人盗得(stole)苹果手机、钱包等财物，价值15000元。
Adversarial Example
被告人劫得(robbed)苹果手机、钱包等财物，价值15000元。
Deep Learning Text Classifier
Label: Robbery
$\mathbb{SA}$: 盗得 (steal)
$\mathbb{SK}$: 财物 (property); 价值 (worth); 手机 (iphone)
$\mathbb{SB}$: 劫得 (rob)
Translation of original example: The defendant stole iphone, wallet and other property, worth 15000 yuan.

Fig. 5 An example of the SSL strategy: "盗得" ("steal" in English) is an SSL biased word for the charge of larceny, while "劫得" ("rob" in English) is an SSL biased word for the charge of robbery. The two words have similar meanings. When we replace "盗得" with "劫得", the label changes the charge of larceny to the charge of robbery

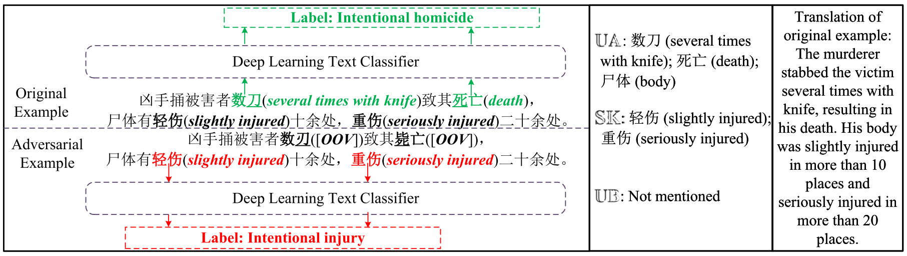

Label: Intentional homicide
Deep Learning Text Classifier
Original Example
凶手捅被害者数刀(several times with knife)致其死亡(death)，
尸体有轻伤(slightly injured)十余处，重伤(seriously injured)二十余处。
Adversarial Example
凶手捅被害者数刃([OOV])致其毙亡([OOV])，
尸体有轻伤(slightly injured)十余处，重伤(seriously injured)二十余处。
Deep Learning Text Classifier
Label: Intentional injury
$\mathbb{UA}$: 数刀 (several times with knife); 死亡 (death); 尸体 (body)
$\mathbb{SK}$: 轻伤 (slightly injured); 重伤 (seriously injured)
$\mathbb{UB}$: Not mentioned
Translation of original example: The murderer stabbed the victim several times with knife, resulting in his death. His body was slightly injured in more than 10 places and seriously injured in more than 20 places.

Fig. 6 An example of the WOL strategy: "数刀" ("several times with a knife" in English) and "死亡" ("death" in English) are unique keywords for the charge of intentional homicide. "轻伤" ("slightly injured" in English) and "重伤" ("seriously injured" in English) are the shared keywords between the charges of intentional homicide and intentional injury. When we introduce perturbations into the two unique keywords, the shared keywords begin to take effect. Accordingly, the label changes the charge of intentional homicide to the charge of intentional injury. "OOV" means out of the predefined vocabulary

4.4 Perturbation method

We comprehensively adopt nine representative token-level perturbation methods for Chinese while ensuring the readability of adversarial texts. The details of these perturbation methods are presented below.

(1) Shuffle (Cheng et al., 2020): Shuffle is a character-level perturbation method that introduces perturbations into a word by scrambling the order of Chinese characters in the word.
(2) Splitting-Character (SC) (Cheng et al., 2020): SC is a character-level perturbation method that adds perturbation into a Chinese character with a left-right structure by decomposing the character into its constituent radicals.
(3) Tradition (Tong et al., 2020): Tradition is a character-level perturbation method that replaces a simplified Chinese character with its traditional counterpart. This method is applicable in cases where the traditional and simplified forms of a character differ.

(4) Glyph (Zhang et al., 2020): Glyph is a character-level perturbation method that involves replacing a Chinese character with another Chinese character that has a similar visual appearance to the original one.
(5) Pinyin (Zhang et al., 2020): Pinyin is a word-level perturbation method that involves replacing a word with another word that has a similar pronunciation to the original word.
(6) Synonyms (Ou et al., 2022): Synonyms is a word-level perturbation method that selects synonyms for the original word based on a thesaurus and replaces the original word with its synonym.
(7) Word Embedding (Jin et al., 2020): Word Embedding is a word-level perturbation method that finds the top k words closest to the original word in the word embedding space and obtains a candidate word list for replacement. This method employs the Word2Vec word vector model, which is consistent with the model mentioned in Sect. 4.1.3.
(8) BERT-MLM (Zhang et al., 2023): BERT-MLM is a word-level perturbation method that utilizes the MLM mechanism of the BERT model to predict words for masked positions based on context, thereby generating a list of candidate replacement words for the original word. Notably, to enhance the fluency of the adversarial texts, we make minor modifications to the perturbation methods proposed in Chinese BERT Tricker (Zhang et al., 2023). Considering the linguistic characteristics of Chinese, we ensure that the replacement words generated for original words with a length of one character also have a length of one character, and the replacement words generated for original words with a length greater than one character have a length of two characters.
(9) Hybrid-7: We propose a novel and strong hybrid perturbation method called Hybrid-7. To ensure the efficiency of adversarial text generation, we combine the aforementioned perturbation methods, excluding Glyph, into Hybrid-7. This method constructs a list of replacement words generated through various perturbation methods for the original words to expand the range of perturbation types available for selection, thereby considerably reducing the classification accuracy of the target model.

Word-level perturbation methods typically generate a candidate word list for replacing the original word, from which we select the most appropriate word for substitution. In this work, we replace the original word with the one that results in the most remarkable change in label confidence before and after replacement. Assuming a word $\mathbf{w}$ in a text $\mathbf{x}$ has a candidate word list $\mathbb{C}$, where the confidence change $\Delta c_{\mathbf{c}_i}$ for any candidate word $\mathbf{c}_i$ in $\mathbb{C}$ can be calculated using the following formula:

$$\Delta c_{\mathbf{c}_i} = \begin{cases} F_y(\mathbf{x}) - F_y(\hat{\mathbf{x}}), \quad \text{if } F(\mathbf{x}) = F(\hat{\mathbf{x}}) = y \\ F_y(\mathbf{x}) - F_y(\hat{\mathbf{x}}) + \left[F_{y'}(\hat{\mathbf{x}}) - F_{y'}(\mathbf{x})\right], \\ \text{if } F(\mathbf{x}) = y \wedge F(\hat{\mathbf{x}}) = y' \wedge y \neq y' \end{cases}, \tag{9}$$

where $\hat{\mathbf{x}}$ denotes the text resulting from the substitution of $\mathbf{w}$ with $\mathbf{c}_i$. The final replacement word $\mathbf{w}'$ can be expressed as:

$$\mathbf{w}' = \underset{\mathbf{c}_i \in \mathbb{C}}{\arg\max} \ \Delta c_{\mathbf{c}_i}. \tag{10}$$

5 Experiments

5.1 Experimental setup

5.1.1 Dataset

In this study, we utilize the largest publicly available Chinese law dataset, CAIL2018 (Xiao et al., 2018), which encompasses 2,676,075 criminal cases, 183 criminal law articles, and 202 distinct charges. Our research focuses on 11 categories of charges: larceny, robbery, forcible seizure, dangerous driving, traffic accident, smuggling, selling, trafficking, and producing drugs, providing venues for drug users, illegal possession of drugs, intentional injury, intentional homicide, and negligent homicide. For each category, we select 4,500 cases to constitute the training set and 500 cases for the validation set. Concurrently, we randomly choose 1,100 cases that are not in the training data to generate adversarial texts.

5.1.2 Target model and training details

We employ the powerful Transformer-based Chinese BERT model as the target model. We fine-tune the bert-base-chinese model to adapt it for the Chinese criminal charge classification task. The hidden size is set to 768. During training, the padding size, batch size, and epochs are configured to 256, 16, and 3, respectively. We adopt an Adam optimizer with a learning rate of 5×10^{-5} for training, and the GPU utilized is the NVIDIA GeForce RTX 3080.

5.1.3 Baselines and our methods

To control for variables, we consistently employ the word importance scoring method, improved DS (Jin et al., 2020), as the word importance scoring method for all baselines and our methods. For a detailed introduction to the improved DS, please refer to Sect. 4.2. We combine the improved DS with various perturbation methods introduced in Sect. 4.4, which serve as the baselines utilized in this experiment. We incorporate the proposed KALT as an extension component into these baselines to constitute our methods. The baselines and our methods are as follows.

5.1.4 Baselines

(1) WordChange-Sh (Cheng et al., 2020): WordChange-Sh combines Shuffle with the improved DS.
(2) WordChange-SC (Cheng et al., 2020): WordChange-Sh combines SC with the improved DS.
(3) CWordAttacker-T (CWA-T) (Tong et al., 2020): CWA-T combines Tradition with the improved DS.
(4) Argot-G (Zhang et al., 2020): Argot-G combines Glyph with the improved DS.
(5) Argot-P (Zhang et al., 2020): Argot-P combines Pinyin with the improved DS.
(6) GreedyAttack-S (Ou et al., 2022): GreedyAttack-S combines Synonyms with the improved DS.

(7) TextFooler (Jin et al., 2020): TextFooler combines Word Embedding with the improved DS.
(8) Chinese BERT Tricker (CBT) (Zhang et al., 2023): CBT combines BERT-MLM with the improved DS.

5.1.5 Our Methods

(1) Attack-7: Attack-7 combines Hybrid-7 with the improved DS.
(2) Attack-7+KALT: Attack-7+KALT combines Attack-7 with KALT.
(3) WordChange-Sh+KALT: WordChange-Sh+KALT combines WordChange-Sh with KALT.
(4) WordChange-SC+KALT: WordChange-SC+KALT combines WordChange-SC with KALT.
(5) CWA-T+KALT: CWA-T+KALT combines CWA-T with KALT.
(6) Argot-G+KALT: Argot-G+KALT combines Argot-G with KALT.
(7) Argot-P+KALT: Argot-P+KALT combines Argot-P with KALT.
(8) GreedyAttack-S+KALT: GreedyAttack-S+KALT combines GreedyAttack-S with KALT.
(9) TextFooler+KALT: TextFooler+KALT combines TextFooler with KALT.
(10) CBT+KALT: CBT+KALT combines CBT with KALT.

5.2 Experimental results

In this section, we evaluate the attack performance of our proposed KALT from four dimensions: effectiveness, text similarity, interpretability, and the proportion of high-confidence adversarial texts. Finally, adversarial training is employed to examine the effectiveness of KALT against defensive measures.

5.2.1 Effectiveness

The attack effectiveness of adversarial text generation methods aimed at text classification tasks is typically assessed by two metrics: the change in the classification accuracy of the target model before and after the attack and the attack success rate on the target model. A lower classification accuracy and a higher attack success rate indicate higher attack effectiveness. In this work, the former is adopted as the evaluation measure for attack effectiveness. Table 1 displays the classification accuracy of the target model after attacks by the baselines and our methods. In Table 1, "Initial" denotes the original method, while "+KALT" denotes the method combined with KALT. The bold values indicate the classification accuracy of the target model after being attacked by more effective methods.

From Table 1, it is observed that with the increase in perturbation rate, the classification accuracy of the target model decreases. Concurrently, our method demonstrates varying degrees of enhancement in the attack effectiveness for all character-level adversarial text generation methods, as well as some word-level adversarial text generation methods, such as Argot-P. Among these, the effectiveness improvement provided by our proposed KALT is more significant for character-level adversarial text generation methods and less so for Argot-P. KALT does not improve attack effectiveness for some word-level and hybrid perturbation-based adversarial text generation methods, such as TextFooler, CBT, and Attack-7. This is attributed to the rich candidate word selection

Table 1 Classification accuracy (%) of target model after attacks by various methods

Basic method	With KALT or not	Perturbation rate									
		0.00	0.01	0.02	0.03	0.04	0.05	0.08	0.10	0.13	0.15
WordChange-Sh	Initial	90.36	89.73	89.36	89.36	89.18	89.09	89.00	88.82	88.82	88.82
	+KALT∗	90.36	**85.55**	**84.91**	**84.64**	**84.18**	**83.73**	**83.55**	**83.45**	**83.09**	**83.18**
WordChange-SC	Initial	90.36	86.36	85.09	84.18	83.64	82.82	82.09	81.64	81.18	80.91
	+KALT∗	90.36	**82.64**	**81.27**	**79.82**	**79.09**	**78.27**	**77.36**	**76.82**	**76.45**	**75.73**
CWA-T	Initial	90.36	90.36	90.27	90.27	90.27	90.27	90.18	90.18	90.18	90.18
	+KALT∗	90.36	**86.36**	**85.73**	**85.27**	**85.00**	**85.00**	**84.64**	**84.55**	**84.27**	**84.18**
Argot-G	Initial	90.36	80.09	77.55	75.00	72.00	69.36	65.36	63.73	61.36	60.55
	+KALT∗	90.36	**77.64**	**74.91**	**72.09**	**69.45**	**67.36**	**63.73**	**62.00**	**60.00**	**59.18**
Argot-P	Initial	90.36	83.27	80.36	79.09	77.55	75.73	73.45	72.55	72.18	72.09
	+KALT∗	90.36	**79.45**	**75.91**	**74.09**	**72.64**	**70.73**	**68.91**	**68.09**	**67.36**	**67.36**
GreedyAttack-S	Initial	90.36	75.09	**72.36**	70.09	**68.09**	**66.73**	**62.00**	**60.00**	57.82	**57.00**
	+KALT∗	90.36	75.09	73.00	70.09	68.18	67.27	62.18	60.18	**57.64**	57.09
TextFooler	Initial	90.36	**64.27**	**58.64**	**53.36**	**50.36**	**47.09**	**41.73**	**39.73**	**36.45**	**35.18**
	+KALT∗	90.36	66.73	61.45	56.27	53.64	50.64	45.00	42.82	40.45	38.91
CBT	Initial	90.36	**54.27**	**45.18**	**40.64**	**36.55**	**34.73**	**29.36**	**26.27**	**22.73**	**21.91**
	+KALT∗	90.36	55.09	47.00	41.82	37.91	36.27	31.27	28.09	24.27	23.55
Attack-7∗	Initial	90.36	**50.09**	**39.64**	**35.45**	**31.82**	**29.18**	**22.55**	**20.18**	**17.55**	**15.82**
	+KALT	90.36	52.00	42.64	37.45	33.91	31.64	24.55	21.73	19.27	17.82

of these methods, which compensates for the lack of knowledge to some extent. However, these methods fail to capture knowledge accurately, resulting in adversarial texts generated by them being inferior in text similarity, interpretability, and the proportion of high-confidence adversarial texts compared to those that incorporate KALT. A detailed analysis of these three aspects is provided in Sects. 5.2.2 to 5.2.4.

5.2.2 Text similarity

This work adopts four evaluation metrics to comprehensively assess the similarity between adversarial texts generated using various methods and the original texts. These four evaluation metrics are cosine similarity, word mover's distance (WMD), edit distance, and Jaccard similarity coefficient. A brief introduction to each is provided below.

(1) Cosine similarity: Cosine similarity is a common method for calculating text similarity. The closer the cosine similarity of word or sentence vectors is to 1, the more similar the words or sentences are. Consider a vector $\mathbf{a} = \left(a_1, a_2, \dots, a_n\right)$ and a vector $\mathbf{b} = \left(b_1, b_2, \dots, b_n\right)$ in the word embedding space. The cosine similarity $C(\mathbf{a}, \mathbf{b})$ can be expressed by the following equation:

$$C(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} \left(a_i\right)^2} \times \sqrt{\sum_{i=1}^{n} \left(b_i\right)^2}}. \tag{11}$$

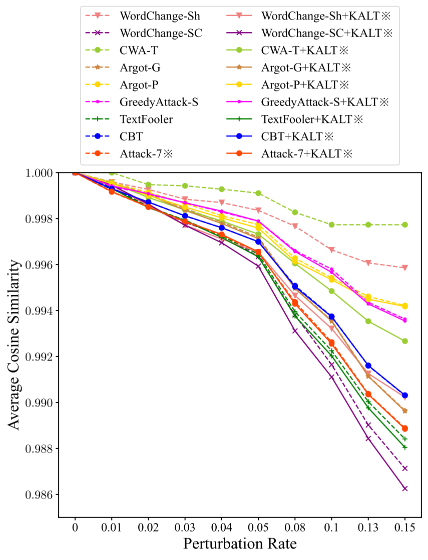
(2) WMD: In the word vector space generated by Word2Vec (Mikolov et al., 2013), the WMD calculates text similarity using the Euclidean distance. For specific details, please refer to (Kusner et al., 2015). The smaller the WMD, the more similar the texts are.

(3) Edit distance: Edit distance is the minimum number of character changes required to transform one string into another, with the condition that only one character can be modified at a time. This work adopts the Levenshtein distance to calculate text similarity. The permitted operations are insertion, deletion, and substitution. The smaller the edit distance, the more similar the texts are.

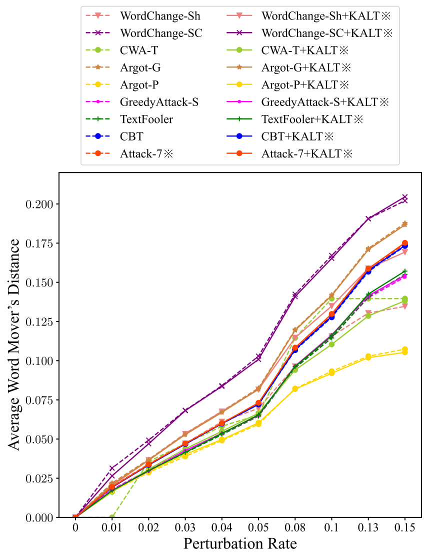
(4) Jaccard similarity coefficient: The Jaccard similarity coefficient measures the similarity between two sets of words. The closer the Jaccard similarity coefficient is to 1, the higher the similarity between the two sets of words. After processing two texts into set $\mathbb{A}$ and set $\mathbb{B}$ of words, the Jaccard similarity coefficient $J(\mathbb{A}, \mathbb{B})$ between $\mathbb{A}$ and $\mathbb{B}$ can be calculated using the following formula:

$$J(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|} = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A}| + |\mathbb{B}| - |\mathbb{A} \cap \mathbb{B}|}. \tag{12}$$

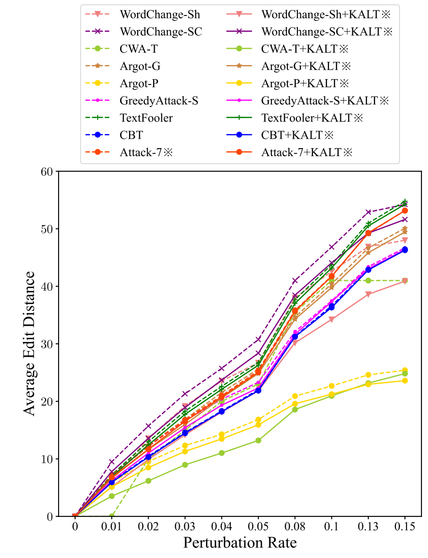
Figure 7 presents a comparison of the similarity between adversarial texts generated by baselines and our methods and the original texts across the aforementioned four evaluation metrics. As the perturbation rate increases, the similarity between adversarial texts generated by various methods and the original texts exhibits a decreasing trend. Simultaneously, as can be observed from Fig. 7, in terms of the WMD and edit distance, the adversarial texts generated by incorporating our proposed KALT exhibit significantly higher similarity to the original texts compared to those generated by methods without KALT. However,



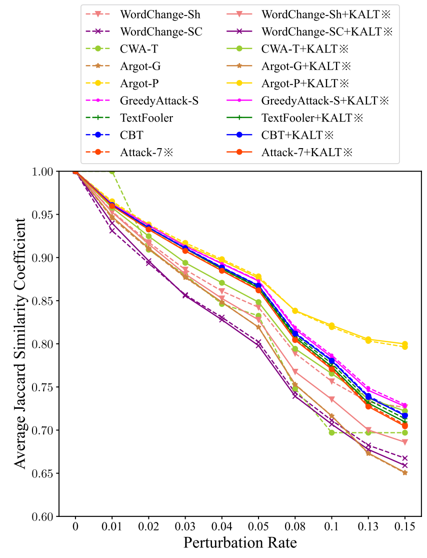
(a) Cosine similarity.



(b) Word mover's distance.



(c) Edit distance.



(d) Jaccard similarity coefficient.

Fig. 7 Text similarity evaluation

in terms of the cosine similarity and Jaccard similarity coefficient, the baselines and our methods perform similarly on text similarity. Particularly for the cosine similarity, the difference between the highest and lowest points on the vertical axis is only 0.014. This could be attributed to the predominance of longer texts in legal documents, where perturbed content is little within a perturbation rate of 0.15 or less.

5.2.3 Interpretability

In this work, the interpretability of an adversarial text generation method is concretely demonstrated through the proportion of the adversarial texts misclassified into other categories similar to the original one. The higher the similarity between the labels after misclassification and the original label, the stronger the interpretability of the adversarial text generation method. Figure 8 illustrates the three labels most similar to each original label. In this work, we posit that the greater the number of shared keywords, the higher the similarity between two labels. In Fig. 8, the symbol "/" denotes that these categories share the same number of keywords as the original category and are ranked equally.

To more intuitively demonstrate the interpretability of each method, we propose an interpretability score formula for quantifying interpretability. When calculating the interpretability scores, we introduce two parameters: a grading coefficient, η, and a reduction coefficient, λ. The former controls the grading system used for scoring, while the latter assigns different weights to labels with varying degrees of similarity to the original label. This work employs the 100-point scale, and we set η to 100. Additionally, we set λ to 2, meaning that the weight assigned to the label most similar to the original label is twice that of the second most similar label, and so on. We categorize the labels similar to the original label into different levels based on varying degrees of similarity. The number of levels is denoted by v. The higher the level, the lower the similarity to the original label. In this work, the value of v is set to 3, indicating that we only focus on the categories in the top three levels of similarity to the original category. In contrast, other categories are assigned a weight of 0. If there are multiple labels in a level, these labels equally share the weight allocated to the level. Assuming there are m categories, the interpretability score I_k for the kth category ($k = 1, 2, ..., m$) can be calculated using the following formula:

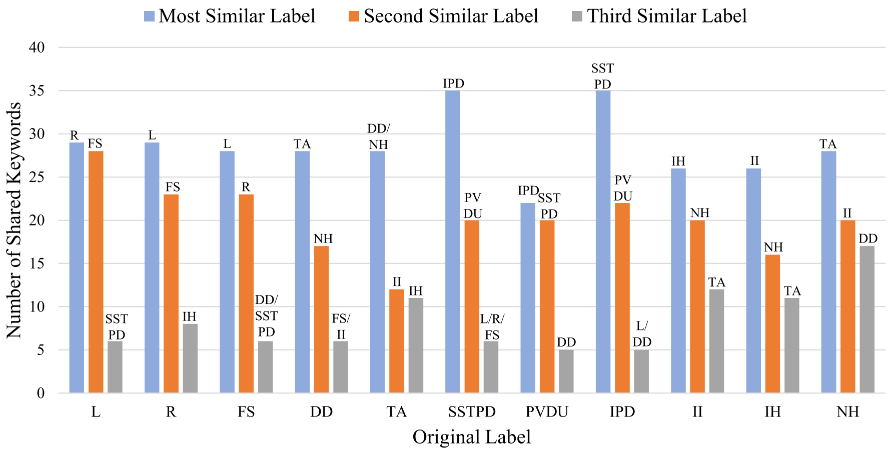


Fig. 8 The three labels most similar to the original label. The symbol "/" denotes an equal number of shared keywords among multiple labels

Table 2 Percentage (%) of misclassified adversarial texts for each category after attacks by WordChange-SC

	L	R	FS	DD	TA	SSTPD	PVDU	IPD	II	IH	NH	Score
L	–	**66.67**	33.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	83.34
R	**50.00**	–	0.00	0.00	0.00	0.00	0.00	0.00	14.29	35.71	0.00	58.93
FS	**33.33**	16.67	–	0.00	0.00	16.67	0.00	0.00	0.00	**33.33**	0.00	43.75
DD	**35.00**	15.00	0.00	–	25.00	0.00	0.00	5.00	0.00	5.00	15.00	32.50
TA	12.50	12.50	0.00	0.00	–	0.00	0.00	0.00	12.50	12.50	**50.00**	34.38
SSTPD	0.00	0.00	0.00	0.00	0.00	–	**50.00**	**50.00**	0.00	0.00	0.00	75.00
PVDU	0.00	0.00	0.00	0.00	0.00	**66.67**	–	33.33	0.00	0.00	0.00	66.66
IPD	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**	–	0.00	0.00	0.00	50.00
II	0.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	–	**80.00**	10.00	85.00
IH	16.67	16.67	0.00	0.00	0.00	0.00	0.00	0.00	**33.33**	–	**33.33**	49.99
NH	0.00	0.00	0.00	0.00	20.00	0.00	0.00	0.00	0.00	**80.00**	–	20.00
Average interpretability score												54.50

Table 3 Percentage (%) of misclassified adversarial texts for each category after attacks by WordChange-SC+KALT

	L	R	FS	DD	TA	SSTPD	PVDU	IPD	II	IH	NH	Score
L	–	**96.15**	3.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.08
R	**70.00**	–	0.00	0.00	0.00	0.00	0.00	0.00	10.00	20.00	0.00	75.00
FS	**52.63**	44.74	–	0.00	0.00	0.00	0.00	0.00	0.00	2.63	0.00	75.00
DD	**36.84**	15.79	0.00	–	26.32	0.00	0.00	5.26	0.00	5.26	10.53	31.59
TA	14.29	14.29	0.00	0.00	–	0.00	0.00	0.00	14.29	14.29	**42.86**	32.15
SSTPD	0.00	0.00	0.00	0.00	0.00	–	**66.67**	33.33	0.00	0.00	0.00	66.66
PVDU	0.00	0.00	0.00	0.00	0.00	**66.67**	–	33.33	0.00	0.00	0.00	66.66
IPD	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**	–	0.00	0.00	0.00	50.00
II	0.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	–	**80.00**	10.00	85.00
IH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**50.00**	–	**50.00**	75.00
NH	0.00	0.00	0.00	0.00	20.00	0.00	0.00	0.00	0.00	**80.00**	–	20.00
Average interpretability score												61.38

$$I_k = \eta \sum\nolimits_{i=1}^{v} \sum\nolimits_{j=1}^{u_i} \frac{p_{ij}}{\lambda^{i-1} u_i}, \tag{13}$$

where p_{ij} represents the percentage of misclassified adversarial texts corresponding to the jth category in the ith level, while u_i denotes the number of categories contained in the ith level. When calculating the interpretability score for a method, we compute the mean of the interpretability scores across all categories after attacks by the method. The average interpretability score $\bar{I}$ of the method can be computed employing the following formula:

$$\bar{I} = \frac{1}{m} \sum\nolimits_{k=1}^{m} I_k. \tag{14}$$

Table 4 Percentage (%) of misclassified adversarial texts for each category after attacks by GreedyAttack-S

	L	R	FS	DD	TA	SSTPD	PVDU	IPD	II	IH	NH	Score
L	–	**78.26**	21.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	89.13
R	4.76	–	2.38	0.00	0.00	0.00	0.00	0.00	0.00	**92.86**	0.00	29.16
FS	1.56	**71.88**	–	0.00	0.00	0.00	0.00	0.00	0.00	25.00	1.56	37.50
DD	7.14	**35.71**	0.00	–	**35.71**	0.00	0.00	0.00	0.00	7.14	14.29	42.86
TA	7.14	14.29	0.00	11.90	–	0.00	0.00	0.00	2.38	14.29	**50.00**	35.71
SSTPD	0.00	0.00	0.00	0.00	0.00	–	28.57	**57.14**	0.00	14.29	0.00	71.42
PVDU	0.00	12.50	0.00	0.00	0.00	25.00	–	**62.50**	0.00	0.00	0.00	75.00
IPD	0.00	0.00	0.00	0.00	0.00	33.33	**66.67**	–	0.00	0.00	0.00	66.66
II	0.00	5.26	0.00	0.00	0.00	0.00	0.00	0.00	–	**78.95**	15.79	86.84
IH	8.33	33.33	0.00	0.00	0.00	0.00	0.00	0.00	**50.00**	–	8.33	54.16
NH	3.85	0.00	0.00	11.54	23.08	0.00	0.00	0.00	7.69	**53.85**	–	29.81
Average interpretability score												56.20

Table 5 Percentage (%) of misclassified adversarial texts for each category after attacks by GreedyAttack-S+KALT

	L	R	FS	DD	TA	SSTPD	PVDU	IPD	II	IH	NH	Score
L	–	**87.50**	12.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.75
R	18.18	–	4.55	0.00	0.00	0.00	0.00	0.00	4.55	**72.73**	0.00	38.64
FS	21.21	**57.58**	–	0.00	0.00	0.00	0.00	0.00	0.00	19.70	1.52	50.00
DD	8.33	33.33	0.00	–	**41.67**	0.00	0.00	0.00	0.00	8.33	8.33	45.84
TA	7.14	14.29	0.00	11.90	–	0.00	0.00	0.00	2.38	14.29	**50.00**	35.71
SSTPD	0.00	0.00	0.00	0.00	0.00	–	28.57	**57.14**	0.00	14.29	0.00	71.42
PVDU	0.00	12.50	0.00	0.00	0.00	25.00	–	**62.50**	0.00	0.00	0.00	75.00
IPD	0.00	0.00	0.00	0.00	0.00	33.33	**66.67**	–	0.00	0.00	0.00	66.66
II	0.00	5.26	0.00	0.00	0.00	0.00	0.00	0.00	–	**78.95**	15.79	86.84
IH	9.09	36.36	0.00	0.00	0.00	0.00	0.00	0.00	**45.45**	–	9.09	50.00
NH	4.17	0.00	0.00	12.50	25.00	0.00	0.00	0.00	4.17	**54.17**	–	30.21
Average interpretability score												58.55

We set the perturbation rate to 0.05 and exemplify interpretability using character-level WordChange-SC and the word-level GreedyAttack-S, presenting the results of the misclassification proportions and the interpretability scores before and after the incorporation of KALT for the two methods in Tables 2, 3, 4, 5, respectively. In Tables 2, 3, 4, 5, each row represents the original category, and each column indicates the category into which the adversarial texts are misclassified after attacks. We represent each category using the initials of the words that constitute the name of each charge. The bold values represent the proportion of adversarial texts corresponding to the misclassified labels with the highest number of adversarial texts.

Combining the analysis of Tables 2, 3, 4, 5 with Fig. 8, the following four conclusions can be drawn: (1) Even without incorporating our proposed KALT, the initial WordChange-SC and GreedyAttack-S tend to classify the original category into similar categories, indicating that the initial WordChange-SC and GreedyAttack-S possess a certain degree of interpretability; (2) After incorporating our proposed KALT, WordChange-SC and Greedy-Attack-S are more prone to misclassifying adversarial texts into the labels most similar to the original label. Compared to the initial methods without KALT, the methods with KALT exhibit higher average interpretability scores, implying that KALT is capable of enhancing interpretability; (3)The increase in interpretability scores for WordChange-SC is significantly greater than that for GreedyAttack-S before and after the incorporation of KALT, suggesting that the improvement in interpretability by KALT for character-level WordChange-SC is more significant than that for word-level GreedyAttack-S; (4) The interpretability scores for categories of larceny, robbery, and forcible seizure applicable to the SSL perturbation strategy are remarkably higher than those for other categories applicable to the WOL perturbation strategy, revealing that the SSL perturbation strategy enhances the interpretability of the methods more effectively than the WOL perturbation strategy.

5.2.4 High confidence percentage

The proportion of adversarial texts misclassified with high confidence can reflect the extent to which the target model is confused. The higher the proportion of adversarial texts misclassified with high confidence, the deeper the degree of confusion in the target model. The proportion of adversarial texts misclassified with high confidence can further reflect the interpretability of adversarial text generation methods. Adversarial texts generated by methods with high interpretability tend to be misclassified with high confidence. In this work, we set 0.8 as the threshold for high confidence, meaning that adversarial texts misclassified with confidence greater than 0.8 are considered misclassified with high confidence. Table 6 displays the percentage of adversarial texts misclassified with high confidence after attacks by baselines and our methods. In Table 6, "Initial" denotes the original method, while "+KALT" denotes the method combined with KALT. The bold values represent the proportion of adversarial texts misclassified with high confidence after the target model is attacked by methods that more effectively confuse the target model.

From Table 6, it is evident that incorporating our proposed KALT into all adversarial text generation methods leads to varying degrees of increase in the proportion of adversarial texts misclassified with high confidence. Combining Sect. 5.2.3, it becomes apparent that including KALT can guide the target model in misclassifying adversarial texts into categories similar to the original one with high confidence, further substantiating the ability of KALT to enhance the interpretability of adversarial text generation methods. Simultaneously, as can be seen from Table 6, KALT considerably elevates the proportion of adversarial texts misclassified with high confidence for all character-level methods and the word-level Argot-P, while the increase is less pronounced for other word-level methods and hybrid perturbation-based Attack-7.

5.2.5 Adversarial training

Adversarial training is one of the widely employed defense measures against adversarial attacks (Jin et al., 2020; Li et al., 2019, 2020, 2021). In this section, we adopt adversarial training to investigate the attack effectiveness of our proposed KALT in the face of defense mechanisms. We take the character-level WordChange-SC with KALT and the word-level TextFooler with KALT as examples to study this issue. We mix the original texts with the adversarial texts generated by the two methods and retrain the target models, respectively. Subsequently, we generate adversarial texts using texts not included in the training set to

Table 6 Percentage (%) of adversarial texts misclassified with high confidence after attacks by various methods

Basic method	With KALT or not	Perturbation rate									
		0.00	0.01	0.02	0.03	0.04	0.05	0.08	0.10	0.13	0.15
WordChange-Sh	Initial	0.00	14.29	9.09	9.09	7.69	14.29	13.33	11.76	5.88	5.88
	+KALT∗	0.00	**86.79**	**83.33**	**84.13**	**82.35**	**79.45**	**78.67**	**78.95**	**78.75**	**79.75**
WordChange-SC	Initial	0.00	38.64	44.83	48.53	51.35	44.58	45.05	48.96	49.50	48.08
	+KALT∗	0.00	**67.06**	**67.00**	**67.24**	**67.74**	**63.91**	**63.64**	**65.10**	**67.32**	**63.35**
CWA-T	Initial	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+KALT∗	0.00	**93.18**	**92.16**	**91.07**	**91.53**	**91.53**	**92.06**	**92.19**	**89.55**	**88.24**
Argot-G	Initial	0.00	51.33	51.77	49.11	46.53	45.45	44.73	47.10	48.59	50.61
	+KALT∗	0.00	**62.86**	**62.35**	**58.21**	**57.39**	**56.92**	**55.97**	**58.65**	**57.49**	**59.18**
Argot-P	Initial	0.00	52.56	45.45	45.16	46.10	46.58	45.70	44.39	46.00	45.77
	+KALT∗	0.00	**69.17**	**62.89**	**58.10**	**60.00**	**58.80**	**55.93**	**54.69**	**55.73**	**55.73**
GreedyAttack-S	Initial	0.00	68.45	64.65	67.26	65.71	64.23	**62.50**	**59.88**	**60.61**	**61.31**
	+KALT∗	0.00	68.45	**65.97**	**67.71**	**65.98**	**66.93**	61.94	59.04	60.00	60.93
TextFooler	Initial	0.00	74.56	72.21	66.34	67.50	65.55	65.98	65.17	63.07	63.26
	+KALT∗	0.00	**76.15**	**73.27**	**66.93**	**69.31**	**67.28**	**66.53**	**66.16**	**64.30**	**63.78**
CBT	Initial	0.00	**68.26**	68.01	69.84	70.44	72.06	72.13	70.07	69.62	69.46
	+KALT∗	0.00	67.53	**68.34**	**69.85**	**70.54**	**73.45**	**73.23**	**70.66**	**70.56**	**69.93**
Attack-7∗	Initial	0.00	**71.11**	69.53	**70.53**	70.34	70.13	69.03	**67.62**	**68.16**	66.34
	+KALT	0.00	69.19	**69.71**	70.10	**70.53**	**71.21**	**69.06**	66.62	67.77	**67.04**

Table 7 Classification accuracy (%) before and after adversarial training

Method	Target model	Perturbation rate									
		0.00	0.01	0.02	0.03	0.04	0.05	0.08	0.10	0.13	0.15
WordChange-SC+KALT	Initial	90.36	82.64	81.27	79.82	79.09	78.27	77.36	76.82	76.45	75.73
	+ AT	88.27	85.18	84.55	83.64	83.00	82.36	81.00	80.55	80.18	80.00
TextFooler+KALT	Initial	90.36	66.73	61.45	56.27	53.64	50.64	45.00	42.82	40.45	38.91
	+ AT	84.82	65.00	57.45	51.73	48.09	44.09	38.64	35.91	32.82	30.82

attack the retrained target models. After being attacked by the two methods, the changes in the classification accuracy of the original target model and the retrained target models are shown in Table 7. In Table 7, “Initial” represents the original target model, and “+AT” denotes the target models after adversarial training.

From Table 7, it is evident that even after the adversarial training, the target models cannot defend against the attacks from WordChange-SC and TextFooler enhanced with KALT. Although adversarial training provides defense against attacks by WordChange-SC enhanced with KALT to some extent, the classification accuracy of the retrained target model decreases continuously with the increase in the perturbation rate. This indicates that WordChange-SC with KALT retains a degree of attack effectiveness on the retrained target model. Compared to WordChange-SC with KALT, adversarial training exhibits suboptimal defensive efficacy against attacks by TextFooler with KALT. After being attacked by TextFooler with KALT, the classification accuracy of the retrained target model is even lower than that of the original target model. This phenomenon can be attributed to two main reasons. On the one hand, mixing the original texts with adversarial texts inevitably introduces noise data, decreasing the classification accuracy of the target model that is not attacked. On the other hand, the attack effectiveness of TextFooler with KALT is excellent, indicating that it can successfully attack the target model despite the defensive measure of adversarial training.

6 Conclusion

We propose a novel adversarial text generation method incorporating legal knowledge, KALT, for the charge classification task in the legal domain. KALT leverages KeyBERT to extract legal knowledge and integrates the knowledge into our proposed perturbation strategies: SSL and WOL. By incorporating legal knowledge, KALT can guide the target model in misclassifying adversarial texts into categories similar to the original one, thereby enhancing the interpretability of the adversarial text generation process. To more vividly demonstrate the improvement in interpretability offered by KALT, we formulate a computational formula for the interpretability score to assess interpretability quantitatively. Furthermore, KALT can serve as a component to be integrated into any adversarial text generation method based on word importance, thereby effortlessly enhancing the adversarial attack performance of approaches designed for common domains when applied to the legal field. The attack performance of KALT is evaluated based on the attack effectiveness, textual similarity, interpretability, and the proportion of adversarial texts misclassified with high confidence. The experimental results indicate that the majority of adversarial text generation methods augmented with KALT outperform their original counterparts in terms of attack effectiveness. Concurrently, KALT also increases the similarity between adversarial and original texts. Additionally, the high interpretability often allows adversarial texts enhanced by KALT to mislead the target model with high confidence, inducing a deeper level of confusion for the target model than methods without KALT. In future research, we plan to extend our investigation to other verticals, aiming to devise adversarial text generation methods tailored to these areas, thereby laying a foundation for developing subsequent defensive measures.

Author contribution (1) Yunting Zhang: Writing - Original Draft, Methodology, Software, Validation, Formal analysis, Visualization. (2) Shang Li: Writing - Review & Editing, Conceptualization. (3) Lin Ye: Funding acquisition, Project administration. (4) Hongli Zhang: Resources, Supervision. (5) Zhe Chen: Investigation. (6) Binxing Fang: Supervision.

Funding This work was supported by the National Natural Science Foundation of China under Grant 61872111 and the Key R&D Program of Heilongjiang Province of China under Grant JD2023SJ07.

Data availability Data will be made available on request.

Code availability Code will be made available on request.

Declarations

Conflict of interest/Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors approved the final manuscript and the submission to this journal.

References

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B., Srivastava, M. B. & Chang, K. (2018). Generating natural language adversarial examples. In *Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, Belgium, October 31–November 4* (pp. 2890–2896). https://doi.org/10.18653/v1/d18-1316.

Cheng, N., Chang, G., Gao, H., Pei, G., & Zhang, Y. (2020). Wordchange: Adversarial examples generation approach for Chinese text classification. *IEEE Access, 8*, 79561–79572. https://doi.org/10.1109/ACCESS.2020.2988786

Chen, Y., Gao, H., Cui, G., Qi, F., Huang, L., Liu, Z. & Sun, M. (2022). Why should adversarial perturbations be imperceptible? Rethink the research paradigm in adversarial NLP. In *Proceedings of the 2022 conference on empirical methods in natural language processing, Abu Dhabi, United Arab Emirates* (pp. 11222–11237). https://doi.org/10.18653/v1/2022.emnlp-main.771.

Chen, Y., Gao, H., Cui, G., Yuan, L., Kong, D., Wu, H., Shi, N., Yuan, B., Huang, L., Xue, H., Liu, Z., Sun, M. & Ji, H. (2023). From adversarial arms race to model-centric evaluation: Motivating a unified automatic robustness evaluation framework. In *Findings of the association for computational linguistics: ACL 2023, Toronto, Canada* (pp. 9607–9632). https://doi.org/10.18653/v1/2023.findings-acl.611.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies* (pp. 4171–4186). https://doi.org/10.18653/v1/n19-1423.

Gao, J., Lanchantin, J., Soffa, M. L. & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE security and privacy workshops, SP workshops 2018, San Francisco, CA, USA, May 24* (pp. 50–56). https://doi.org/10.1109/SPW.2018.00016.

Garg, S. & Ramakrishnan, G. (2020). BAE: bert-based adversarial examples for text classification. In *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, November 16–20* (pp. 6174–6181). https://doi.org/10.18653/v1/2020.emnlp-main.498.

Goodfellow, I. J., Shlens, J. & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *3rd international conference on learning representations*.

Grootendorst, M. (2020). KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265.

Jin, D., Jin, Z., Zhou, J. T. & Szolovits, P. (2020) 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, New York, NY, USA, February 7-12* (pp. 8018–8025).

Jin, G., Liu, C., & Chen, X. (2021). Adversarial network integrating dual attention and sparse representation for semi-supervised semantic segmentation. *Information Processing & Management, 58*(5), 102680.

Kusner, M., Sun, Y., Kolkin, N. & Weinberger, K. (2015). From word embeddings to document distances. In *Proceedings of the 32nd international conference on machine learning* (pp. 957–966).

Liu, H., Yin, Q. & Wang, W. Y. (2019). Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th conference of the association for computational linguistics* (pp. 5570–5581).

Li, J., Ji, S., Du, T., Li, B. & Wang, T. (2019). Textbugger: Generating adversarial text against real-world applications. In *26th annual network and distributed system security symposium, NDSS 2019, San Diego, California, USA* (February 24–27).

Li, S., Zhang, H., Ye, L., Guo, X., & Fang, B. (2019). MANN: A multichannel attentive neural network for legal judgment prediction. *IEEE Access, 7*, 151144–151155.

Li, L., Shao, Y., Song, D., Qiu, X. & Huang, X. (2020). Generating adversarial examples in chinese texts using sentence-pieces. CoRR abs/2012.14769. arXiv:2012.14769.

Li, L., Ma, R., Guo, Q., Xue, X. & Qiu, X. (2020). BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, Online, November 16–20* (pp. 6193–6202). https://doi.org/10.18653/v1/2020.emnlp-main.500.

Li, S., Zhang, H., Ye, L., Su, S., Guo, X., Yu, H., & Fang, B. (2020). Prison term prediction on criminal case description with deep learning. *CMC-Computers, Materials & Continua, 62*(3), 1217–1231.

Li, D., Zhang, Y., Peng, H., Chen, L., Brockett, C., Sun, M. & Dolan, B. (2021). Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2021, Online, June 6–11*, pp. 5053–5069. https://doi.org/10.18653/v1/2021.naacl-main.400.

Luo, P., Chu, J., & Yang, G. (2023). Ip packet-level encrypted traffic classification using machine learning with a light weight feature engineering method. *Journal of Information Security and Applications, 75*, 103519. https://doi.org/10.1016/j.jisa.2023.103519

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st international conference on learning representations*.

Ou, H., Yu, L., Tian, S., & Chen, X. (2022). Chinese adversarial examples generation approach with multi-strategy based on semantic. *Knowledge and Information Systems, 64*(4), 1101–1119. https://doi.org/10.1007/s10115-022-01652-1

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J. & Fergus, R. (2014). Intriguing properties of neural networks. In *2nd international conference on learning representations*.

Tong, X., Wang, L., Wang, R., & Wang, J. (2020). A generation method of word-level adversarial samples for chinese text classification. *Netinfo Security, 20*(9), 12–16. https://doi.org/10.3969/j.issn.1671-1122.2020.09.003

Wang, W., Wang, R., Wang, L. & Tang, B. (2019). Adversarial examples generation approach for tendency classification on chinese texts. *Journal of Software*, *30*(8), 2415–2427.https://doi.org/10.13328/j.cnki.jos.005765

Wu, B., Zou, F., Zhang, C., Yu, T., & Li, Y. (2023). Multi-field relation mining for malicious http traffic detection based on attention and cross network. *Journal of Information Security and Applications, 73*, 103411. https://doi.org/10.1016/j.jisa.2022.103411

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H. & Xu, J. (2018). CAIL2018: A large-scale legal dataset for judgment prediction. CoRR abs/1807.02478. arXiv:1807.02478.

Xu, J., & Du, Q. (2020). Adversarial attacks on text classification models using layer-wise relevance propagation. *International Journal of Intelligent Systems, 35*(9), 1397–1415. https://doi.org/10.1002/int.22260

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision - ECCV 2014 - 13th European conference* (Vol. 8689, pp. 818–833).

Zhang, Z., Liu, M., Zhang, C., Zhang, Y., Li, Z., Li, Q., Duan, H. & Sun, D. (2020). Argot: Generating adversarial readable Chinese texts. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI* (pp. 2533–2539). https://doi.org/10.24963/ijcai.2020/351.

Zhang, Y., Ye, L., Tang, H., Zhang, H. & Li, S. (2023). Chinese BERT attack method based on masked language model. *Journal of Software*https://doi.org/10.13328/j.cnki.jos.006932