Research Article

DPAEG: A Dependency Parse-Based Adversarial Examples Generation Method for Intelligent Q&A Robots

Mingfu Xue, Chengxiang Yuan, Jian Wang, and Weiqiang Liu

1College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
2Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China
3College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Correspondence should be addressed to Mingfu Xue; mingfu.xue@nuaa.edu.cn

Received 1 June 2019; Revised 23 December 2019; Accepted 26 December 2019; Published 13 January 2020

Copyright © 2020 Mingfu Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, the natural language processing-based intelligent question and answer (Q&A) robots have been used ubiquitously. However, the robustness and security of current Q&A robots are still unsatisfactory, e.g., a slight typo in the user’s question may cause the Q&A robot unable to return the correct answer. In this paper, we propose a fast and automatic test dataset generation method for the robustness and security evaluation of current Q&A robots, which can work in black-box scenarios and thus can be applied to a variety of different Q&A robots. Specifically, we propose a dependency parse-based adversarial examples generation (DPAEG) method for Q&A robots. DPAEG first uses the proposed dependency parse-based keywords extraction algorithm to extract keywords from a question. Then, the proposed algorithm generates adversarial words according to the extracted keywords, which include typos and words that are spelled similarly to the keywords. Finally, these adversarial words are used to generate a large number of adversarial questions. The generated adversarial questions which are similar to the original questions do not affect human’s understanding, but the Q&A robots cannot answer these adversarial questions correctly. Moreover, the proposed method works in a black-box scenario, which means it does not need the knowledge of the target Q&A robots. Experiment results show that the generated adversarial examples have a high success rate on two state-of-the-art Q&A robots, DrQA and Google Assistant. In addition, the generated adversarial examples not only affect the correct answer (top-1) returned by DrQA but also affect the top-k candidate answers returned by DrQA. The adversarial examples make the top-k candidate answers contain fewer correct answers and make the correct answers rank lower in the top-k candidate answers. The human evaluation results show that participants with different genders, ages, and mother tongues can understand the meaning of most of the generated adversarial examples, which means that the generated adversarial examples do not affect human’s understanding.

1. Introduction

In recent years, artificial intelligence (AI) has developed rapidly both in its techniques and applications. A typical application of AI is the natural language processing-based (NLP-) intelligent question and answer (Q&A) robots [1], which are not only used in general applications but also in professional business or government applications. Recently, many companies have developed their Q&A robots and put them on the market, such as Google Assistant [2], Cortana [3], Siri [4], Alexa [5], and Watson [6]. Unlike search engines (e.g., Google and Baidu) that provide a ranked list of relevant web documents to the user, the task of an intelligent Q&A robot is to give the user a precise and concise answer in several interactions with the user [7]. In general, the Q&A robot has the following two features: (1) users can query the Q&A robot in natural language and (2) the answer returned by the Q&A robot is directly the answer that the user needs, instead of a ranked list of relevant documents.

Many current Q&A robots use NLP models to understand user’s questions and return answers [8]. However, the NLP models still have some shortcomings. For example, studies show that NLP models are not robust enough [9] that a small typo in the user’s input may cause the NLP models to
fail to process the question. Besides, the NLP models used in Q&A robots may not truly understand the semantics of the user’s question [10], which cause the Q&A robots to give irrelevant answers. Moreover, NLP models are also vulnerable to the adversarial examples attacks [11]. These shortcomings of the NLP models will affect the robustness and security of current Q&A robots, which will lead to a very bad user experience.

To date, there are some research studies on the robustness and security of machine learning models, such as [12–16], but little research has been done on the robustness and security issues of these ubiquitous Q&A robots. Motivated by these issues, in this paper, we propose a fast and automatic test dataset generation method to evaluate the robustness of Q&A robots by crafting adversarial questions. Although the proposed method only makes minor modifications to the original questions, these carefully constructed adversarial questions can easily make state-of-the-art Q&A robots answer incorrectly. Moreover, these generated adversarial questions are quite similar to the original questions and thus do not affect human’s understanding of these adversarial questions.

There are some adversarial example generation methods for text classifiers in the literature, such as [17–22]. However, these adversarial example generation methods for text classifiers are not suitable for Q&A robots. The reasons are as follows. (1) The application scenarios of text classifiers and Q&A robots are different. Text classifiers are applied to spam filtering, sentiment analysis, fake news detection, and so on. Q&A robots are applied to intelligent customer service, smart home service, professional question and answer, information query, and so on. (2) The techniques adopted by text classifiers and Q&A robots are also different. Text classifiers use a single NLP model to perform classification tasks. However, since the tasks performed by Q&A robots are more complex, Q&A robots use multiple different NLP models in the entire process of understanding questions and searching for answers. (3) The adversarial example generation methods for text classifiers are based on a single target model, and some of these generation methods require specific knowledge of the target model, such as [17, 18, 21]. However, for Q&A robots, attackers cannot obtain the specific knowledge of the Q&A robots in most cases. Therefore, it is more difficult to generate adversarial examples for Q&A robots than for text classifiers. In this paper, the proposed method determines the important words of the questions and modifies them slightly, which does not require the design information of the target Q&A robot and thus has a strong universality over a wide range of Q&A robots.

The proposed method first exploits a dependency parser to extract keywords from the original question. Then, the adversarial words of keywords are generated. The adversarial words contain three types of words: typos of keywords, words that are spelled similarly to the keywords, typos of these spell-similar words. The spell-similar words are obtained by searching for words in the English dictionary that satisfy the proposed three constraints. Typos of keywords and typos of spell-similar words are common misspelled words, which are obtained by querying the typos corpus and discarding those typos that have a large edit distance from the keywords. Finally, the keywords in the original question are replaced by adversarial words to generate large number of adversarial questions. In the experiment, the adversarial examples are generated from WebQuestionsSP, CuratedTREC, and WikiMovies Q&A datasets. Two state-of-the-art Q&A robots, DrQA and Google Assistant, are used to evaluate the success rate of the proposed method. The experiment results on DrQA and Google Assistant show that the generated adversarial examples can make the Q&A robot go wrong with a high success rate. The experimental results in terms of recall @k (R_n@k) [23], mean reciprocal rank (MRR) [24], and mean average precision (MAP) [24] further show that the generated adversarial examples also affect the top-k candidate answers returned by DrQA. The adversarial examples result in fewer correct answers in the top-k candidate answers and make the correct answers rank lower in the top-k candidate answers. Besides, we invite participants with different genders, ages, and mother tongues to evaluate the quality of the generated adversarial examples. The human evaluation results show that different participants can understand the meaning of most of the adversarial examples generated by the proposed method.

The main contributions of this paper are as follows:

(i) Many previous text adversarial example generation methods require the knowledge of the target model to determine important parts of a text sequence which are further modified to generate adversarial examples. However, our proposed keywords extraction algorithm can determine important parts of a question without the knowledge of the design information of the Q&A robots. Therefore, the proposed method can work under black-box situations. Moreover, to the best of the authors’ knowledge, this is the first adversarial examples generation method for intelligent Q&A robots and also the first automatic test dataset generation method for the robustness and security evaluation of Q&A robots.

(ii) The proposed algorithm first extracts keywords from a given question and then generates adversarial words that are similar to the extracted keywords. These words are used to replace the corresponding words in the original question to generate large number of adversarial examples. Since the differences between generated adversarial questions and the original question are inconspicuous, humans are not aware of these adversarial words when reading the question. Human evaluation of participants with different genders, ages, and mother tongues shows that they have no trouble understanding the generated adversarial questions.
But, the state-of-the-art Q&A robots cannot answer the adversarial questions correctly.

(iii) The proposed adversarial question generation method can provide a fast and automatic test dataset generation method for the robustness and security evaluation of current Q&A robots in black-box scenarios. Besides, the proposed method has strong universality and can be used to evaluate the robustness of a wide range of different Q&A robots.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. Section 3 elaborates the proposed adversarial examples generation algorithm for Q&A robots. Experimental results are presented in Section 4. Finally, conclusions are given in Section 5.

2. Related Work

Generally, there are three kinds of working mechanisms used by current Q&A robots: using the knowledge base (KB), using information retrieval (IR), and using both KB and IR. The KB-based Q&A robots transform a question into a standard structured query through semantic parsing and then get answer from the KB [25]. The key step of this type of Q&A robots is to transform the user’s natural language questions into standard structured query languages [25]. Currently, many Q&A robots use machine learning techniques to understand the semantics of the questions, such as [25–27]. In [25], Yih et al. used an entity linking system and a deep convolutional neural network model for question answering. Yin et al. [26] proposed an end-to-end neural network model to generate answers. For IR-based Q&A robots, such as [28–30], they retrieve unstructured text documents and extract relevant answers from these documents. The DrQA, which is developed by Facebook [31], is a Q&A model for answering questions by retrieving and reading unstructured knowledge. DrQA uses Wikipedia as the unique knowledge source and uses recurrent neural network (RNN) model to extract answers from relevant articles [28]. Some Q&A robots, such as YodaQA [32], QuASE [33], and Watson [34], combine KB and IR techniques to get answers of the questions. Baudiš [32] proposed a Q&A framework, named YodaQA. YodaQA searches unstructured and structured knowledge and then uses a classifier to determine the best matching answer. Sun et al. [33] proposed a QuASE system for open-domain question answering, which searches for answers directly from the web and uses a knowledge base to further improve the accuracy of answering questions.

Although different Q&A robots have different working mechanisms, many current Q&A robots use NLP models when processing users’ questions and searching for the correct answers [35]. Unfortunately, the NLP models are vulnerable to adversarial examples, which are carefully designed inputs by an attacker to cause the model to produce erroneous outputs [36]. Recently, there are some adversarial example generation methods in NLP tasks, including text classification, machine translation, and reading comprehension. For instance, in [17–19], the authors search for the most important part of a text sequence for the text classifier and then make slight modifications to this part to generate adversarial examples. These modifications include insertion, substitution, removal, etc. When targeting machine translation model, Ebrahimi et al. and Belinkov and Bisk [37, 38] used noisy texts to generate adversarial examples, which make the machine translation results change greatly. When targeting reading comprehension systems, Jia and Liang [10] added irrelevant sentences to the input to fool the reading comprehension system. To the best of the authors’ knowledge, there is no research on adversarial examples generation for intelligent Q&A robots. However, since many NLP models are applied to Q&A robots, Q&A robots also face the threat of adversarial examples in practice. For example, when interacting with a Q&A robot, the user often misspells the words in a question, which will cause the Q&A robot to return a wrong answer or an irrelevant answer.

In this paper, the proposed adversarial examples generation method for Q&A robots is to modify an important part of a question slightly. Compared with other methods, the difference between the generated adversarial examples and the original question is more inconspicuous, and the proposed method can work in black-box scenarios. This minor modification hardly changes the semantics of the original question. Even if the semantics of a single word changes, people can still infer the semantics from the context of the question. Human evaluation experiments show that humans can understand the original meaning of the generated adversarial examples. In addition, the proposed method exploits a dependency parser to determine the important parts of a question without the knowledge of the design information of the Q&A robot. Therefore, it can be applied to various Q&A robots under black-box scenarios.

3. The Proposed DPAEG Method

3.1. Overall Procedure. In this section, we elaborate the proposed dependency parse-based adversarial examples generation (DPAEG) method. DPAEG replaces an important part of the original question with typos or words that are spelled similarly. The framework of the proposed adversarial examples generation method is shown in Figure 1. There are four stages in the proposed method to craft adversarial examples. First, the proposed method preprocesses the questions from Q&A datasets, which removes the original questions that the target Q&A robot cannot answer correctly. This means, in the Q&A dataset, only the original questions that the Q&A robot can answer correctly are retained for adversarial examples generation. Second, the proposed dependency parse-based keywords extraction algorithm is used to extract keywords from the original questions. Third, the proposed adversarial words generation algorithm is used to slightly modify keywords of a question, which includes three types of modifications, typos of keywords, spell-similar words, and typos of these spell-similar words. Specifically, by searching in a dictionary according to the proposed constraints, words that are spelled similarly to the keywords are determined. The typos of keywords and typos of these spell-similar words are determined from the typos corpus according to the edit distance settings. Finally,
the keywords in the original question are replaced by adversarial words to generate large number of adversarial questions. The detailed process of each stage is described in the following sections.

3.2. Preprocess: Questions Filtering. For any given question, the proposed method is able to generate a large number of adversarial questions. In the experiment, three standard Q&A datasets (WebQuestionsSP [39], CuratedTREC [40], and WikiMovies [41]) are used to provide original questions. Any other questions are also feasible. Since the target Q&A robot cannot correctly answer all the original questions in these three datasets, it is meaningless to generate adversarial examples with those original questions that the Q&A robot cannot answer. Therefore, a preprocessing operation is applied to the Q&A datasets, and the original questions that the target Q&A robot cannot answer correctly are removed. The remaining questions that the target Q&A robot can answer correctly are used to generate adversarial examples.

3.3. Keywords Extraction Based on Dependency Parse. The proposed method extracts keywords according to the importance of words in a question. Generally, if modifying or removing a word in a question causes a significant change in the answer given by a Q&A robot, it indicates that this word is important for the Q&A robot to correctly understand and answer the question. However, since the Q&A robot is a black box for attackers, it is difficult to determine the important part of a question according to the Q&A robot except for continuous interactions with the Q&A robot. To solve this problem, the proposed keywords extraction algorithm identifies the important parts of a question according to the dependency relation of the question and thus can work in a black-box scenario without interactions with Q&A robots. Note that, the extracted keywords are determined by the dependencies between the words in the current sentence. If the same word has different dependencies in different sentences, the importance of the word in different sentences will be different.

The dependency relation is a method of describing the grammatical structure of a sentence, which represents the grammatical relation between words in a sentence [42]. Generally, a dependency parser converts a sentence into a dependency tree. The root of the tree is called the head of the sentence, which does not modify any word [42]. An example of a dependency parse for the sentence “Who played the voice of Aladdin” is shown in Figure 2. The root of the dependency tree is “played.” The arrow represents the dependency relation between two parts. For instance, the dependency relation between “Who” and “played” is the nsubj relation, which means that “Who” is the nominal subject (nsubj) of “played.” Similarly, “the voice” is the direct object (dobj) of “played,” “of” is the prepositional modifier (prep) of “the voice,” and “Aladdin” is the prepositional object (pobj) of “of”.

The dependent relations of a sentence can be divided into aux (auxiliary), arg (argument), and mod (modifier) [43]. These relations can be further divided into 48 different grammatical relations. In order to extract the important parts of an input question, the proposed keywords extraction method only focuses on words that satisfy the following rule: the dependent relations between the word and the head of the sentence is in the arg relation set (Rarg). The dependent relations contained in the arg relation set are shown in Figure 3 [43].

The proposed keywords extraction algorithm is shown in Algorithm 1. Firstly, a dependency parser is used to extract a dependency tree from the question qori. The dependency parser used in this method is a dependency parser provided by spaCy (https://spacy.io), which is a natural language processing tool. spaCy uses a transition-based parser to extract dependencies [44], and the process of extracting the dependency relation of a question is recapitulated as follows. Initially, the parser has an empty stack and a buffer, where the original question is in the buffer [44]. Then, the parser uses the shift and reduce operations to control the state of the

![Figure 1: Framework of the proposed adversarial questions generation method.](image1)

![Figure 2: An example of a dependency parse for the sentence “Who played the voice of Aladdin.”](image2)
stack and the buffer [44]. The shift operation moves the word in the buffer to the top of the stack, while the reduce operation pops top two words in the stack and determines the dependency relation between these two words [44]. The shift and reduce operations are repeated until the stack and buffer are empty. As a result, the dependency relation of the question is obtained, which is represented as a dependency tree. All nodes on the dependency tree are denoted by \( T = \langle w_i, h_i, r_i \rangle \mid w_i, h_i \in q_{ori} \), where \( w_i \) is the word on the \( i \)th node of the tree, \( h_i \) is the word on the parent node of the \( i \)th node, and \( r_i \) is the dependent relation between \( w_i \) and \( h_i \). Then, if the root of the dependency tree is a content word, the root is added to the keyword set. For each child node of the root, if the child node satisfies the following two conditions, the word on the child node is added to the keyword set. The two conditions are (1) the dependent relation between the child node and the root is in the arg relation set \( R_{arg} \) and (2) the word on the child node is a content word. Finally, if the question \( q_{ori} \) contains a clause, the root of the dependency tree is first replaced by the head of the clause. Then, the keywords are extracted in the same way for the clause. After extracting the keywords, the important parts of the question \( q_{ori} \) are determined. These extracted keywords are denoted as \( K = \{ k_1, k_2, k_3, \ldots, k_p \} \), where \( p \) is the number of keywords.

Since the function words in a question have little effect on the answer returned by the Q&A robot, the function words are not used as keywords in the proposed method. Compared with using all the content words as keywords, the proposed algorithm only uses content words that have a greater influence on the Q&A robot for returning the correct answer. In Section 4.4, we will compare the performance of the proposed keywords extraction method with content words extraction method that selects all content words from the question as the keywords.
3.4. Adversarial Words Generation Based on Extracted Keywords. To mislead a Q&A robot, the input questions are slightly modified to generate adversarial examples. The difference between the original question and the modified question should be as small as possible so that humans have no trouble understanding the modified questions. To this end, the proposed method generates adversarial words that are similar to the extracted keywords. These adversarial words are used to modify the corresponding keywords in the original question. The proposed adversarial word generation method is shown in Algorithm 2, which generates three types of adversarial words: typos of keywords, words that are spelled similarly to the keyword, and typos of these spell-similar words.

We describe Algorithm 2 in detail as follows. The algorithm determines typos of the keyword \( k \) from the typos corpus. If the edit distance between the keyword \( k \) and typos of the keyword \( k \) is less than or equal to 2, the typos of the keyword \( k \) are added to the adversarial words set \( W_{adv} \). The adopted typos corpus is publicly available in [45], which contains the Birkbeck typos corpus [46], the Holbrook typos corpus [47], the Aspell typos corpus [48], and Wikipedia typos corpus [49].

Words that are spelled similarly to the keyword are determined by searching in a dictionary according to proposed constraints. The dictionary contains common English words [50], which are divided into 26 subdictionaries according to the initial letters. Firstly, the subdictionary \( SD_i \) is determined according to the initials of the keyword, in which the initials of all words in the \( SD_i \) are the same as the initials of the keyword. Then, if the word \( w \) in the subdictionary \( SD_i \) satisfies the proposed constraints, the word \( w \) is added to the corresponding similar word set \( KS \). The proposed constraints are as follows:

(i) The edit distance between the word \( w \) and the keyword \( k \) is less than or equal to a predefined edit distance \( d \).

(ii) The part of speech (POS) of the word \( w \) is the same as the POS of the keyword \( k \).

(iii) The first letter of the word \( w \) is the same as the first letter of the keyword \( k \). Similarly, the last letter of the word \( w \) is the same as the last letter of the keyword \( k \).

The first constraint can identify words that are spelled similarly to the keyword. The purpose of the second constraint is to increase the success rate of adversarial attacks. The effect of the second constraint on the success rate of the generated adversarial examples is demonstrated in Section 4.3.1. The reasons behind the third constraint are as follows. On the one hand, inspired by [38], keeping the first and last letters of the word unchanged makes it easier for humans to recognize the original form of the modified word. On the other hand, sufficient similar words can be searched in a subdictionary. Hence, it is unnecessary to spend more time searching for more similar words from other subdictionaries. This constraint can make the algorithm only search one of the 26 subdictionaries, which can effectively reduce the number of searches and thus improve the search efficiency.

The Damerau–Levenshtein distance [51, 52] is used to evaluate the edit distance between two words. For the keyword \( k \) and a word \( w \) in the dictionary, the Damerau–Levenshtein distance between them (\( \text{dis}(k, w) \)) is the minimum number of character operations required to convert the keyword \( k \) to the word \( w \). Character operations include inserting, deleting, replacing a single character, or transposing two adjacent characters [53]. In order to search for appropriate similar words, we set different predefined edit distances according to the POS of the keyword. The value of \( d \) is determined by the following rule:

\[
d = \begin{cases} 
1, & \text{if } \text{POS of } k \text{ is verb,} \\
\max\left\{1, \left\lfloor \frac{\text{length}(k) - 2}{2} \right\rfloor \right\}, & \text{otherwise,}
\end{cases}
\]

where \( \text{length}(k) \) is the length of the keyword \( k \) and the max function ensures that the distance \( d \) is not less than 1. If the POS of the keyword \( k \) is a verb, the distance \( d \) is set to 1. Otherwise, the distance \( d \) is set to be \( \max\{1, \lfloor (\text{length}(k) - 2)/2 \rfloor \} \). The reason for setting different predefined edit distances \( d \) for the verb and other words in a sentence is as follows. A verb is an important part of a sentence. If the difference between the verb of the modified sentence and the original sentence is too large, it may affect human understanding of the modified sentence. Hence, such predefined distance settings can ensure that the edit distance between the verb of the adversarial example and the verb of the original question is small so that people have no difficulty in understanding the generated adversarial examples.

After searching for words that are similar to the keyword, the adversarial words are generated based on these similar words. For each keyword \( k_s \), in the similar word set \( KS \), if the edit distance between keyword \( k \) and \( k_s \) is less than or equal to 2, the word \( k_s \) is added directly to the adversarial word set \( W_{adv} \). Otherwise, the algorithm searches for typos of the word \( k_s \). If there are typos of \( k_s \) in the typos corpus and the edit distances between these typos and the keyword are less than or equal to 2, the typos of \( k_s \) are added to the adversarial words set \( W_{adv} \). Finally, for each keyword in the question, a corresponding adversarial word set is obtained.

3.5. Adversarial Questions Generation. For each keyword, the corresponding adversarial words are generated. These adversarial words are used to replace the corresponding keywords in the original question to generate multiple adversarial questions. However, if too many keywords are replaced in the original question, humans cannot infer the semantics from the context of the question and may have trouble understanding the generated adversarial examples. Hence, in order to prevent too many keywords from being modified in the original question, the following criterion is applied to select the appropriate adversarial questions:
The adversarial question set is generated. Finally, for each original question, a corresponding adversarial question is added to the adversarial questions set, defined as:

\[ q_{\text{ori}} \] and \[ q_{\text{adv}} \] are the original question and the generated adversarial question, respectively. The algorithm is designed to add typos of spell-similar words to generate adversarial examples in a constant time. To limit the degree of modification, an edit distance constraint is used, defined as:

\[ \text{dis}(q_{\text{ori}}, q_{\text{adv}}) \leq \epsilon \]

where \( q_{\text{ori}} \) and \( q_{\text{adv}} \) are the original question and the generated adversarial question, \( \epsilon \) is a predefined threshold that represents the maximum number of typos in a word. The algorithm first queries the typos of the keyword in the typos corpus, and then adds typos to the generated adversarial question. If the generated adversarial question satisfies the edit distance constraint, it is added to the adversarial questions set. Otherwise, the adversarial question is discarded. Finally, for each original question, a corresponding adversarial question set is generated.

The time complexity of the proposed adversarial examples generation algorithm is analyzed as follows. The proposed adversarial examples generation algorithm consists of three parts: keywords extraction, adversarial words generation, and adversarial questions generation. Assume that there are \( n \) words in the input question. The time complexity of keywords extraction and adversarial questions generation is \( \Theta(n) \). For adversarial words generation, the main time overhead is to search for similar words. If the time cost of each comparison is \( c_1 \) and the time cost of determining the typos of the word is \( c_2 \), the runtime of the adversarial words generation algorithm is approximately:

\[ T = c_1 \cdot m + c_2, \]

where \( m \) is the number of words in the input question. The algorithm is efficient for large datasets, with good scalability and can generate adversarial examples efficiently.
WikiMovies [41], are used to generate adversarial questions. The information of the three datasets are as follows:

(i) WebQuestionsSP: this dataset, which is created by Yih et al. [39], contains semantic parses for the questions from the WebQuestions dataset. There are 4737 questions in the WebQuestionsSP dataset.
(ii) CuratedTREC: this dataset is collected by Baudiš and Šedivý [40] based on the Text Retrieval Conference (TREC) [54] corpus, which consists of 2180 questions extracted from TREC1999, TREC2000, TREC2001, and TREC2002 dataset.
(iii) WikiMovies: this dataset is constructed by Miller et al. [41], which consists of question-answer pairs in the field of movies. The WikiMovies dataset contains training set, development set, and test set. The three sets contain 96k, 10k, and 10k examples, respectively [41]. In the experiment, we use the test set to generate the adversarial examples.

The adversarial examples generated from these standard Q&A datasets can form a new adversarial questions dataset. Unlike these standard Q&A datasets which are used to evaluate the ability of Q&A robot to answer questions, the generated adversarial questions dataset is used to evaluate the robustness of Q&A robots when facing typos and misspellings and evaluate Q&A robots’ understanding of sentence semantics. In other words, if the Q&A robot cannot answer the question in the standard Q&A datasets, it means that the Q&A robot does not have the answer to the question. Unlike this, if the Q&A robot cannot answer the adversarial question in the generated adversarial questions dataset, it indicates that the Q&A robot has the answer to the original question, but it cannot process the perturbation in the adversarial question.

4.1.2. Target Q&A Robots. To illustrate the feasibility of the proposed method, the success rate of the generated adversarial examples on two top Q&A robots, DrQA [28] and Google Assistant [2], is calculated. The information of the two target Q&A robots are as follows:

(i) DrQA is an open-domain question answering system based on Wikipedia, which consists of two components [28]: the document retriever module and the document reader module. The document retriever module searches for articles related to the question from the Wikipedia database, and then the document reader module uses RNN model to extract answers from the relevant articles. DrQA has good performance on multiple Q&A datasets. Therefore, DrQA is a good baseline to evaluate the performance of the proposed adversarial examples generation method.

(ii) Google Assistant [2] is an intelligent personal assistant designed by Google, which provides question and answer service. Users can ask questions to Google Assistant by voice or text. If Google Assistant can correctly answer the user’s question, it will directly return the corresponding answer. Otherwise, it will return web search results related to the question [2]. In the experiment, we send questions to it in plain text and record the answers returned by Google Assistant. If the answer returned by Google Assistant is a web search result, we consider that Google Assistant cannot answer this question correctly.

4.1.3. Evaluation Metric. Success rate [37] is used as the metric to evaluate the adversarial examples generated by the proposed algorithm. The success rate is the ratio of questions that the Q&A robot answers incorrectly in all generated adversarial questions [37]. The higher the success rate of the generated adversarial examples is, the more effective the attack on the target Q&A robot is.

Besides, we use three other metrics, recall @k (R_k) [23], mean reciprocal rank (MRR) [24], and mean average precision (MAP) [24] to evaluate the impact of adversarial examples on the top-k candidate answers returned by the Q&A robot. R_k [23] reflects whether the correct answer exists in the top-k candidate answers returned by the Q&A robot, where n is the number of relevant documents retrieved by the Q&A robot. The same as [23, 55, 56], we use R_10@2 and R_10@5 as the evaluation metrics. MRR [24] reflects the position of the first correct answer in the top-k candidate answers returned by the Q&A robot. MAP [24] reflects the ranking of the correct answers in the top-k candidate answers returned by the Q&A robot. Note that, Google Assistant returns only one answer or some webpages. On the one hand, these top-k related metrics cannot be calculated based on only one answer returned by Google Assistant. On the other hand, since the returned webpages are not specific answers, we also cannot calculate these metrics based on returned webpages. Therefore, we cannot use these three metrics to evaluate the performance of Google Assistant answering adversarial questions. Hence, in this paper, these three top-k related metrics can only be used to evaluate DrQA.

To demonstrate that the adversarial questions generated by the proposed method does not affect human’s understanding, we invite a number of participants to evaluate whether they understand the meaning of the generated adversarial examples. We define a metric named comprehension rate. The comprehension rate of a participant is calculated as \( N_{\text{und}}/N_{\text{all}} \), where \( N_{\text{und}} \) is the number of adversarial examples that the participant can understand correctly and \( N_{\text{all}} \) is the number of all the evaluated adversarial examples.

4.2. Experimental Results

4.2.1. Experimental Results on Q&A Robots. Table 1 shows three samples generated by the proposed method. The underlined letters represent the difference between the generated adversarial example and the original question. In the first example, the keyword in the original question is replaced by a typo of the keyword. In the second example,
the keyword is replaced by a word that is spelled similarly to the keyword. In the third example, the keyword is replaced by a typo of a spell-similar word. It is shown that only one or two characters in the original question are modified, but the answers given by the Q&A robots are very different from the answer to the original question. Table 2 presents the success rate of the adversarial examples generated from the three datasets on the two target Q&A robots. The maximum edit distance $\epsilon$ is set to be 4 in this experiment. It is shown that the generated adversarial examples have a high success rate on DrQA. In other words, for most adversarial examples, DrQA cannot return the correct answers. Compared with DrQA, Google Assistant is more robust to the generated adversarial questions. However, there are still about half of the adversarial questions that Google Assistant cannot answer correctly. Therefore, the generated adversarial examples can mislead the target Q&A robot’s understanding of questions, resulting in a low accuracy of answering questions.

Besides, we use the metrics $R_{10}@2$ and $R_{10}@5$ [23], MRR [24], and MAP [24] to evaluate the impact of adversarial questions on the top-$k$ candidate answers returned by DrQA. Table 3 shows the performance of DrQA answering original questions and adversarial questions in terms of $R_{10}@2$, $R_{10}@5$, MRR, and MAP. Note that since we only use questions that DrQA can answer correctly to generate adversarial questions, we only use questions that DrQA can answer correctly to generate adversarial questions. We identify the adversarial questions not only affect the correct answer (top-1) returned by DrQA but also affect the top-$k$ candidate answers returned by DrQA. Specifically, $R_{10}@2$ and $R_{10}@5$ scores indicate that when DrQA answers adversarial questions, the number of correct answers in the top-$k$ answers returned by DrQA is less than that of DrQA when answering original questions. The MRR score indicates that the adversarial examples make the first correct answer rank lower in the top-$k$ candidate answers returned by DrQA. Compared with the MAP score of DrQA answering original questions, the MAP score of DrQA answering adversarial questions is much lower, which indicates that the generated adversarial questions can significantly make all correct answers rank lower in the top-$k$ candidate answers returned by DrQA.

In the practical use of Q&A robots, different users may use different expressions to describe the same meaning of a question. Therefore, we also evaluate the success rate of generating adversarial examples for questions with the same meaning but different expressions. We select 50 questions that the Q&A robot can answer correctly from the WebQuestionsSP dataset. Then, we rephrase these questions by restructuring these questions and replacing the words in the question with synonyms. The meaning of the restated question is consistent with the original question. Since not all the restated questions that the Q&A robot can answer correctly, it is meaningless to generate adversarial examples from those questions that the Q&A robot cannot answer. Therefore, we discard the restated questions that the Q&A robot cannot answer correctly and discard the corresponding original questions. Finally, for DrQA, we generate 257 and 263 adversarial examples from 41 original questions and 41 corresponding restated questions, respectively. For Google Assistant, we generate 289 and 277 adversarial examples from 44 original questions and 44 corresponding restated questions, respectively. Table 4 shows the success rate of the adversarial questions generated from the original questions and from the restated questions on the two target Q&A robots. The results show that the success rate of the adversarial examples generated using restated questions is similar to that of the adversarial examples generated using original questions. Therefore, the restated questions can also effectively generate adversarial examples. Two examples of adversarial questions are shown in Table 5, which are generated from the corresponding restated questions, respectively. The underlined letters represent the difference between the adversarial question and the original question or the restated question. It is shown that both the adversarial questions generated from the original questions

<table>
<thead>
<tr>
<th>Original question</th>
<th>Adversarial question</th>
<th>DrQA’s answer</th>
<th>Google Assistant’s answer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong>&lt;br&gt;Original question: Who was the leader of the USSR in 1948?&lt;br&gt;Adversarial question: Who was the leader of the USSR in 1948?&lt;br&gt;DrQA’s answer: Joseph Stalin&lt;br&gt;Google Assistant’s answer: Here’s what I found: <a href="https://www.bbc.com/news/world-europe-17858981">https://www.bbc.com/news/world-europe-17858981</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **3**<br>Original question: Who was the leader of the USSR in 1948?<br>Adversarial question: Who was the leader of the USSR in 1948?<br>DrQA’s answer: Cory Everson<br>Google Assistant’s answer: Here you go. The first result is from Wikipedia: https://en.m.wikipedia.org/wiki/Arucaria_arauca
Table 2: Success rate of the adversarial examples generated from the three datasets on the two target Q&A robots ($\epsilon = 4$).

<table>
<thead>
<tr>
<th>Target Q&amp;A robot</th>
<th>WebQuestionsSP (%)</th>
<th>CuratedTREC (%)</th>
<th>WikiMovies (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrQA</td>
<td>83.73</td>
<td>78.49</td>
<td>77.26</td>
</tr>
<tr>
<td>Google Assistant</td>
<td>52.47</td>
<td>45.34</td>
<td>46.33</td>
</tr>
</tbody>
</table>

Table 3: Performance of DrQA answering original questions and adversarial questions in terms of $R_{10}@2$, $R_{10}@5$, MRR, and MAP.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Dataset</th>
<th>$R_{10}@2$</th>
<th>$R_{10}@5$</th>
<th>MRR</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original questions</td>
<td>WebQuestionsSP</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.583</td>
</tr>
<tr>
<td></td>
<td>CuratedTREC</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.587</td>
</tr>
<tr>
<td></td>
<td>WikiMovies</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.576</td>
</tr>
<tr>
<td>Adversarial questions</td>
<td>WebQuestionsSP</td>
<td>0.208</td>
<td>0.255</td>
<td>0.220</td>
<td>0.094</td>
</tr>
<tr>
<td></td>
<td>CuratedTREC</td>
<td>0.287</td>
<td>0.383</td>
<td>0.282</td>
<td>0.073</td>
</tr>
<tr>
<td></td>
<td>WikiMovies</td>
<td>0.276</td>
<td>0.356</td>
<td>0.269</td>
<td>0.078</td>
</tr>
</tbody>
</table>

Since we only use the original questions that DrQA can answer correctly to generate adversarial questions, $R_{10}@2$, $R_{10}@5$, and MRR scores of DrQA answering original questions are 1.

Table 4: Success rates of the adversarial questions generated from the original questions and from the restated questions on the two target Q&A robots.

<table>
<thead>
<tr>
<th>Target Q&amp;A robot</th>
<th>Original questions (%)</th>
<th>Restated questions (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrQA</td>
<td>78.21</td>
<td>76.05</td>
</tr>
<tr>
<td>Google Assistant</td>
<td>49.48</td>
<td>48.73</td>
</tr>
</tbody>
</table>

Table 5: Examples of adversarial questions generated from the original questions and from the restated questions.

### Q1
- **Original question**: What currency do Italy have?
- **Adversarial question**: What currency do Italy have?
- **Original answer**: Euro
- **DrQA’s answer**: Ethiopia
- **Google Assistant’s answer**: Wikipedia has this result: [http://en.m.wikipedia.org/wiki/Italian_lira](http://en.m.wikipedia.org/wiki/Italian_lira)

### Q1’
- **Restated question**: What is Italy’s currency?
- **Adversarial question**: What is Italy’s currency?
- **Original answer**: Euro
- **DrQA’s answer**: Ethiopia
- **Google Assistant’s answer**: Wikipedia has this result: [http://en.m.wikipedia.org/wiki/Italian_lira](http://en.m.wikipedia.org/wiki/Italian_lira)

### Q2
- **Original question**: Who is kobe bryant dad?
- **Adversarial question**: Who is kobe bryant dad?
- **Original answer**: Joe Bryant
- **DrQA’s answer**: Los Angeles Lakers
- **Google Assistant’s answer**: [http://m.youtube.com/watch?v=dYhEB2nfnSg](http://m.youtube.com/watch?v=dYhEB2nfnSg)

### Q2’
- **Restated question**: Who is kobe bryant’s father?
- **Adversarial question**: Who is kobe bryant’s father?
- **Original answer**: Joe Bryant
- **DrQA’s answer**: Los Angeles Lakers
- **Google Assistant’s answer**: [http://m.youtube.com/watch?v=dYhEB2nfnSg](http://m.youtube.com/watch?v=dYhEB2nfnSg)

and those generated from the restated questions can make DrQA and Google Assistant answer incorrectly.

4.2.2. Human Evaluation. In this section, we use the metric comprehension rate to evaluate different humans’ understanding on the generated adversarial examples. Besides, the effect of different maximum edit distance settings on the comprehension rate of humans is presented in Section 4.3. The effect of different keywords modification methods on the comprehension rate of humans is presented in Section 4.4.

To avoid human subjective factors affecting human evaluation results, we invite 10 different participants to evaluate the quality of the generated adversarial examples and determine whether the subjective factors (i.e., gender, age, and mother tongue) of the participants affect the human evaluation results. Specifically, the background of these 10 participants is as follows: (1) there are 5 male participants and 5 female participants; (2) 7 participants are 18–35 years old, and 3 participants are 36–50 years old; and (3) there are 8 participants whose mother tongue is Chinese and 2 participants whose mother tongue is English. Compared with automatic evaluation on the Q&A robots by programs and scripts, human evaluation is a time consuming process for the participants and therefore not suitable for evaluation with a large number of questions. Hence, in this experiment, we randomly select 50 adversarial questions generated from WebQuestionsSP dataset to perform human evaluation and calculate the comprehension rate of each participant.

Table 6 shows the minimum, maximum, and average comprehension rate of participants under different subjective factors. It is shown that participants can understand the meaning of most of the generated adversarial examples. In addition, under different subjective factors, the comprehension rate of each type of participants is similar. In other words, participants with different backgrounds have no difference in understanding the generated adversarial questions. Therefore, the participants’ gender, age, and their mother tongue hardly affect humans’ understanding on the generated adversarial questions, and humans can understand the meaning of the generated adversarial examples correctly.

4.3. Parameter Settings

4.3.1. Different POS Constraints of Similar Words. In the process of generating adversarial words (Section 3.4), the proposed method uses three constraints to search for words that are spelled similarly to the keyword. In order to verify that the second constraint can improve the success rate of the proposed adversarial examples, we compare the success rate of adversarial examples under the following three
settings: (1) the POS of the similar word is the same as the POS of the keyword; (2) the POS of the similar word is different from the POS of the keyword; and (3) there is no constraint on the POS of the similar word. We generate adversarial examples under these three different settings. Then, these generated adversarial examples are applied to the DrQA to calculate the success rate. The comparison results of the three different settings are shown in Figure 4. Obviously, the success rate of the adversarial examples generated in setting 1 is higher than that of the adversarial examples generated in setting 2 and setting 3. Therefore, when searching for the words that are spelled similarly to the keyword, keeping the POS of the similar word the same as the POS of the keyword can effectively improve the success rate of the adversarial examples.

4.3.2. Different Maximum Edit Distance $\varepsilon$. In the proposed method, different maximum edit distance settings not only affect the success rate of the generated adversarial examples on Q&A robots but also affect human’s understanding of the generated adversarial examples. In this section, under different maximum edit distance settings, we evaluate the success rate of the adversarial examples on the Q&A robots and evaluate the comprehension rate of humans of the adversarial examples. The adversarial examples are generated from the three datasets under different maximum edit distances, and DrQA is used to evaluate the success rate of these adversarial examples.

Figure 5 presents the success rate of adversarial examples generated under different maximum edit distance settings. It is shown that the larger the maximum edit distance is, the higher the success rate of the adversarial examples is. The reason behind this is that when the maximum editing distance is set to be large, the difference between the adversarial question and the original question will be large. Therefore, the probability of the Q&A robot answering the question correctly will be small, and thus the success rate of the adversarial examples will be high. However, a large maximum edit distance may make it difficult for humans to understand the generated adversarial questions.

We also invite 10 participants (as mentioned in Section 4.2) to evaluate the effect of maximum edit distance on the comprehension rate of humans. The adversarial questions are generated under the maximum edit distance $\varepsilon = 3, 4, 5$, respectively. For $\varepsilon = 4$, we have evaluated the comprehension rate of humans on the generated adversarial questions in Section 4.2. For $\varepsilon = 3$ and $\varepsilon = 5$, we randomly select 20 adversarial questions generated from the WebQuestionsSP dataset for evaluation, respectively (since human evaluation is a time consuming process for the participants, it is not suitable to evaluate using a large number of questions). Figure 6 shows the minimum, maximum, and average comprehension rate of humans on the generated adversarial

<table>
<thead>
<tr>
<th>Participants</th>
<th>Minimum (%)</th>
<th>Maximum (%)</th>
<th>Average (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>72</td>
<td>82</td>
<td>77.2</td>
</tr>
<tr>
<td>Female</td>
<td>70</td>
<td>82</td>
<td>75.2</td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18–35</td>
<td>72</td>
<td>82</td>
<td>77.7</td>
</tr>
<tr>
<td>36–50</td>
<td>70</td>
<td>76</td>
<td>72.6</td>
</tr>
<tr>
<td>Mother tongue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>70</td>
<td>82</td>
<td>75.7</td>
</tr>
<tr>
<td>English</td>
<td>76</td>
<td>80</td>
<td>78</td>
</tr>
<tr>
<td>All the participants</td>
<td>70</td>
<td>82</td>
<td>76.2</td>
</tr>
</tbody>
</table>

Figure 4: Success rate of adversarial examples generated under three different POS constraints on the DrQA.

Figure 5: Success rate of adversarial examples generated under different maximum edit distance $\varepsilon$. 
questions under the maximum edit distance $\epsilon = 3, 4, 5$, respectively. It is shown that the larger the maximum edit distance is, the lower the comprehension rate of humans is. Therefore, the maximum edit distance is set to be 3~5 to ensure that the generated adversarial examples have a good success rate, while at the same time humans have no difficulty in understanding the generated adversarial examples.

4.4. Keywords Extraction and Keywords Modification Evaluation. In the proposed method, keywords extraction and keywords modification are two important steps of generating adversarial examples. Therefore, we also evaluate the performance of the proposed method from these two aspects.

4.4.1. Keywords Extraction Evaluation. To evaluate the performance of the proposed keywords extraction method, we implement two other keywords extraction methods for comparison. The random keywords extraction method is used as one baseline, which randomly selects one or more words from the question as the keywords. The content words extraction method is used as another baseline, which selects all content words from the question as the keywords. In the evaluation experiment, first, the random keywords extraction method, the content words extraction method, and the proposed keywords extraction method are used to extract keywords, respectively. Then, the extracted keywords are removed from the question to generate adversarial examples. Note that in this experiment, we removed the keywords directly instead of replacing them to evaluate the importance of the keywords found by these three methods. These adversarial examples are generated using the WebQuestionsSP dataset. Lastly, the generated adversarial examples are applied to the target Q&A robots, and the success rates are calculated.

Table 7 presents the success rates of the adversarial examples generated by different keywords extraction methods. Compared with the random keywords extraction method and the content words extraction method, the proposed keywords extraction method has a higher success rate on the DrQA and the Google Assistant. This indicates that the proposed keyword extraction method can effectively extract keywords which are important in the original question. If the keywords in a question change, DrQA and Google Assistant will not be able to answer the question. Therefore, the proposed keyword extraction method can effectively improve the success rate of the generated adversarial questions. It is also shown that the content keywords extraction has a higher success rate than the random keywords extraction method, which indicates that content words are important than function words.

4.4.2. Keywords Modification Evaluation. When evaluating the performance of the proposed keywords modification method, the random keywords modification method and the noisy texts method [38] are used as baselines. First, the proposed keywords extraction algorithm is used to extract keywords from the question. Then, the random keywords modification method, the noisy texts method, and the proposed keywords modification method are used to modify the keywords to generate three different types of adversarial examples, respectively. The random keywords modification method randomly replaces the characters in the keywords. The noisy texts method generates adversarial examples by modifying a word in the following five ways [38]: replacing a single letter, swapping the position of two letters, randomizing the order of letters in a word except the first and last letters, randomizing the order of all letters, and replacing letters with adjacent letters on the keyboard. Similarly, these adversarial examples are generated from the WebQuestionsSP dataset. The generated adversarial examples are applied to the target Q&A robots.

Table 8 presents the success rate of the adversarial examples generated by different keywords modification methods. It is shown that the adversarial examples generated by the proposed method have a higher success rate than the adversarial examples generated by the random keywords modification method. The success rate of the adversarial examples generated by the noisy texts method [38] is close to the success rate of the adversarial examples generated by the proposed method. Note that when targeting the Google Assistant, the success rate of the adversarial examples generated by the noisy texts method [38] is a little bit higher than that of the proposed method. The reason is that the average edit distance ($d = 5.2$) between the adversarial examples generated by the noisy texts method and the original question is larger than the average edit distance ($d = 3.7$) between the adversarial examples generated by the proposed method and the original question. However, larger edit distance makes it more difficult for humans to understand the meaning of the adversarial examples.
Besides, we also compare the impact of different key-words modification methods on the comprehension rate of humans. For the proposed keywords modification method, we have evaluated the comprehension rate of humans on the generated adversarial questions as discussed in Section 4.2. For the random keywords modification method and the noisy texts method [38], we randomly select 20 adversarial questions generated from the WebQuestionsSP dataset, respectively, and evaluate the comprehension rate of 10 participants on the generated adversarial questions.

Figure 7 shows the minimum, maximum, and average comprehension rate of participants under different key-words modification methods. It is shown that the comprehension rate of humans under the proposed keywords modification method is higher than the comprehension rate of humans under other keywords modification methods. In other words, compared with random keywords modification method and the noisy texts method [38], it is easier for humans to understand the meaning of the adversarial questions generated by the proposed keywords modification methods. Overall, the adversarial examples generated by the proposed method have a high success rate on DrQA and Google Assistant, and humans can easily understand the meaning of the generated adversarial examples.

5. Conclusion

In this paper, we propose a novel adversarial examples generation method for Q&A robots, which can be used as a fast and automatic test dataset generation method for the robustness and security evaluation of intelligent Q&A robots in black-box scenarios. The proposed method generates adversarial questions by modifying the important part of a question slightly, which is close to the practical use of Q&A robots, e.g., typos, spelling mistakes, and similar words. These generated adversarial questions can successfully make the Q&A robot answer incorrectly, while it ensures that the difference between the generated adversarial questions and the original question is so small that it does not affect human’s understanding of the question. In the experiment, two state-of-the-art Q&A robots, DrQA and Google Assistant (which are considered to be two top Q&A robots currently), are used to evaluate the success rate of the proposed method. Experimental results show that the generated adversarial examples have high success rates on DrQA and Google Assistant. The metrics $R_n@k$, MRR, and MAP on DrQA further indicate that the generated adversarial examples cause DrQA to return fewer correct answers in the top-$k$ candidate answers and cause the correct answers to rank lower in the top-$k$ candidate answers returned by DrQA. In addition, the human evaluation results demonstrate that even if the participants’ gender, age, and mother tongue are different, they have no difficulty in understanding the generated adversarial examples. This is the first adversarial examples generation method for intelligent Q&A robots and also the first automatic test dataset generation method for the robustness and security evaluation of Q&A robots. This paper can hopefully help evaluate and enhance the robustness of intelligent Q&A robots.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61602241) and the Natural Science Foundation of Jiangsu Province (no. BK20150758).
References


