Attention Please: Your Attention Check Questions in Survey Studies Can Be Automatically Answered

Weiping Pei
Colorado School of Mines, Golden, Colorado
weipingpei@mines.edu

Arthur Mayer*
Colorado School of Mines, Golden, Colorado
arthurmayer@mines.edu

Kaylynn Tu*
Colorado School of Mines, Golden, Colorado
kaylynntu@mines.edu

Chuan Yue
Colorado School of Mines, Golden, Colorado
chuanyue@mines.edu

ABSTRACT

Attention check questions have become commonly used in online surveys published on popular crowdsourcing platforms as a key mechanism to filter out inattentive respondents and improve data quality. However, little research considers the vulnerabilities of this important quality control mechanism that can allow attackers including irresponsible and malicious respondents to automatically answer attention check questions for efficiently achieving their goals. In this paper, we perform the first study to investigate such vulnerabilities, and demonstrate that attackers can leverage deep learning techniques to pass attention check questions automatically. We propose AC-EasyPass, an attack framework with a concrete model, that combines convolutional neural network and weighted feature reconstruction to easily pass attention check questions. We construct the first attention check question dataset that consists of both original and augmented questions, and demonstrate the effectiveness of AC-EasyPass. We explore two simple defense methods, adding adversarial sentences and adding typos, for survey designers to mitigate the risks posed by AC-EasyPass; however, these methods are fragile due to their limitations from both technical and usability perspectives, underlining the challenging nature of defense. We hope our work will raise sufficient attention of the research community towards developing more robust attention check mechanisms. More broadly, our work intends to prompt the research community to seriously consider the emerging risks posed by the malicious use of machine learning techniques to the quality, validity, and trustworthiness of crowdsourcing and social computing.

CCS CONCEPTS

• **Human-centered computing** → **Collaborative and social computing**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

KEYWORDS

Crowdsourcing, Online Survey, Attention Check, Automatic Answer Generation, Deep Learning

ACM Reference Format:
Weiping Pei, Arthur Mayer, Kaylynn Tu, and Chuan Yue. 2020. Attention Please: Your Attention Check Questions in Survey Studies Can Be Automatically Answered. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3366423.3380195

1 INTRODUCTION

Survey is widely used by researchers and decision makers to access vital information such as thoughts, opinions, and feelings from a certain population. For example, psychologists and sociologists make extensive use of surveys to drive their important studies. Market research companies, which globally produce $45 billion in revenue each year [35], leverage surveys as a key quantitative technique to obtain valuable feedback from customers for business strategies. Government agencies, politicians, and news media conduct public opinion polls to derive new policies or make important predictions. The growth and the vast accessibility of the Web have significantly facilitated the popularity of online surveys over the years. Different from traditional surveys, online surveys could easily access to a diverse population and greatly reduce the time and cost of collecting data [50]. Online surveys are usually published on crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) [41].

As online surveys play an important role in research and decision-making, the quality of survey data becomes a crucial concern for crowdsourcing service providers and researchers. Some crowdsourcing service providers do have relevant mechanisms for helping survey requesters improve the data quality. For example, MTurk uses a qualification mechanism to keep tabs on HIT workers and allow survey requesters to pre-select qualified workers [52]. However, service-side support alone is far from enough, and it is still essential for requesters to evaluate the data quality [18, 26] and filter out poor quality data [21, 23, 36] for individual surveys.

Poor data quality could be caused by either legitimate respondents or adversaries [23]. For legitimate respondents, they may submit survey answers in an irresponsible manner. For example, MTurk workers who value money over responsibility would find the fastest and easiest way to complete more HITs with less efforts. For adversaries, they may complete surveys with malicious purposes such as simply polluting the data or purposefully injecting false information. In August 2018, MTurk had a "bot" scare [3]:

*Undergraduate research assistants who contributed to the data collection.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380195

psychology researchers have noticed a spike in poor quality survey responses collected on MTurk, which created a "bot" panic about data quality and potential attacks on crowdsourcing platforms. Data collected from either irresponsible or malicious respondents would have a significant impact on the reliability of the survey results, and cause a variety of severe consequences such as polluting important research studies and misleading decision makings.

Several approaches have been proposed to identify low-quality survey data and filter them out. One approach relies on the respondents' response patterns [21]. For example, those respondents who rush through a survey by selecting the first option in the entire survey would be considered unqualified due to this homogeneous pattern. However, this approach is not dependable because respondents may randomly select survey options without showing suspicious patterns. Another approach is time-based. It assumes that unreliable respondents will spend less effort and thus time than reliable respondents to complete a survey [62]. However, this approach is not dependable either because respondents could be very diverse in their behaviors, and may even switch to other tasks [17].

The third approach is *attention checking*, which embeds in a survey with the attention check questions that have obvious correct answers to identify inattentive respondents. For example, in a multiple choice question "We want to test your attention, so please click on the answer Agree", the correct answer is explicit and those respondents who select other options will be considered as inattentive. Because they are specially designed for attention checking and they are easy to be deployed simply by requesters, attention check questions have become commonly used in online surveys as a key mechanism to filter out inattentive respondents and improve the data quality [2, 16, 18, 51].

However, little research considers the vulnerabilities of this important quality control mechanism that can allow attackers including irresponsible and malicious respondents to automatically answer attention check questions for efficiently achieving their goals. We consider both types of respondents as attackers because their data are not reliable (e.g., inaccurate, invalid, untrustworthy, or even harmful) to survey requesters, and they are essentially compromising the integrity of the corresponding studies.

In this paper, we propose AC-EasyPass, an attack framework with a concrete model, to easily answer attention check questions. We construct the first attention check question dataset that consists of both original and augmented questions. We demonstrate that AC-EasyPass achieves 84.42% mean average precision (MAP), 84.83% mean reciprocal rank (MRR), and 75.65% accuracy on the original attention check questions. It achieves 79.69% MAP, 79.87% MRR, and 68.10% accuracy as well as 86.03% MAP, 86.61% MRR, and 78.54% accuracy on two sets of augmented questions, respectively. Finally, we explore two simple defense methods, adding adversarial sentences and adding typos, for survey designers to mitigate the risks posed by AC-EasyPass. However, these defense methods are fragile due to their limitations from both technical and usability perspectives, underlining the challenging nature of defense.

Overall, our work makes the following contributions: (1) We perform the first study to investigate the vulnerabilities of the attention check mechanism in online surveys, and highlight that they can allow attackers including irresponsible and malicious respondents to automatically answer attention check questions. (2) We propose and design AC-EasyPass, an attack framework with a concrete model, that combines convolutional neural network and weighted feature reconstruction to easily pass attention check questions. We evaluate and analyze the effectiveness of AC-EasyPass. (3) We construct the first attention check question dataset. (4) We explore and evaluate two simple defense methods, adding adversarial sentences and adding typos, but further show that both methods are fragile and defense remains a challenging task. (5) More broadly, we intend to prompt the research community to more seriously consider the emerging security risks posed by the malicious use of artificial intelligence techniques to the quality, validity, and trustworthiness of crowdsourcing and social computing.

2 BACKGROUND AND RELATED WORK

2.1 Background

2.1.1 Quality Control in Crowdsourcing Services. With the increasing popularity of crowdsourcing services, quality control becomes a critical challenge because HIT workers are very diverse in abilities, skills, interests, personal objectives, and technological resources [9]. Researchers have proposed several quality control approaches which mainly fall into two categories: "up-front task design" and "post-hoc result analysis" [29]. The former focuses on preparing well-designed tasks that are resistant to low-quality workers [7, 11]. The latter improves data quality by evaluating results and filtering out those results of low quality, for example, based on gold questions [73], based on consistency on the same questions [54, 61], or based on consensus labels that are inferred by using aggregation methods [58] such as the majority voting model [32] and the Dawid-Skene model [10].

However, those approaches are often inappropriate for subjective tasks such as collecting opinions in surveys because ground-truth is not available. For most survey tasks, whether respondents answer the survey questions attentively is a critical factor for evaluating the validity of the survey results. Based on such nature of surveys, attention checking has become a popular mechanism for improving the quality of survey results. It embeds attention check questions that are easy for attentive respondents to answer but are the "traps" for careless respondents. It has been widely used by researchers for quality control in important online surveys [2, 8, 20].

2.1.2 Two Forms of Attention Check Questions. Two major forms of attention check questions exist [30]:

Instructional Manipulation Checks (IMCs) [42] were first proposed in 2009 and have been widely employed in online surveys since then [16, 18, 43]. IMC is elaborated as a "trick" question with a large block of text. Figure 1 shows an example of IMC [42]. The large block of text describes the purpose of the current question and it ends with a straightforward question: "Please check all words that describe how you are currently feeling". However, the lengthy description of the purpose instructs respondents to ignore this question by clicking the "none of the above" option to pass this IMC. Careless respondents may miss the cue or instruction hidden in the lengthy description and fail this attention check.

Instructed-response Items [4, 25, 60] are designed to also trap respondents who are rushing through a survey. Those items are embedded into the survey and require a specific answer, such as "We

Recent research on decision making shows that choices are affected by context. Differences in how people feel, their previous knowledge and experience, and their environment can affect choices. To help us understand how people make decisions, we are interested in information about you. Specifically, we are interested in whether you actually take the time to read the directions; if not, some results may tell us very much about decision making in the real world. To show that you have read the instructions, please ignore the question below about how you are feeling and instead check only the none of the above option as your answer. Thank you very much.

Please check all words that describe how you are currently feeling.

A. Excited **B.** Afraid **C.** Scared **D.** None of the above

Figure 1: An Example of Instructional Manipulation Check.

We want to test your attention, so please click on the answer Agree.

A. Strongly disagree **B.** Disagree **C.** Neutral **D.** Agree
E. Strongly agree

Figure 2: An Example of Instructed-response Item.

want to test your attention, so please click on the answer Agree" as shown in Figure 2. Different from IMCs, instructed-response items are simpler and require less effort since they do not have the lengthy description to trick respondents. Meanwhile, they often have the similar format as survey questions, while IMCs may sometimes stand out from the typical survey questions [30].

Our constructed dataset (Section 4.1.1) contains both forms of questions. Note that attention check questions are different from gold questions, which are questions with the ground-truth answers provided by domain experts. A survey designer would filter out workers who cannot correctly answer gold questions because they may not have sufficient background knowledge or capability to take some specific-topic surveys. Gold questions were rarely observed in our dataset, and are not what our AC-EasyPass aims to answer.

2.2 Related Work

2.2.1 Research on Attention Check Questions. Attention check questions are common in online surveys and their usefulness has been studied by many researchers especially psychologists. Oppenheimer et al. [42] first proposed IMCs to identify inattentive respondents, and their study demonstrated that the inclusion of an IMC could increase the reliability of the collected data. Since then, attention checking has been considered as a desirable feature in online surveys [39, 43, 49, 49]. Berinsky et al. [2] further discussed the power of the attention check questions and demonstrated that it is more desirable to use multiple attention check questions than using a single one. Gould et al. [17] found that multitasking is a potential source of inattentiveness in crowd-working settings, so certain intervention can help reduce the frequency of task-switching and also improve on existing attention checks. To investigate whether attention check questions would be a threat to scale validity in psychological testing, Kung et al. [30] conducted two studies and found no evidence about the threat.

2.2.2 Related Attacks and Malicious Surveys. Crowdsourcing enables the solving of many important problems by gathering the crowd's intelligence and wisdom. Correspondingly, attackers are

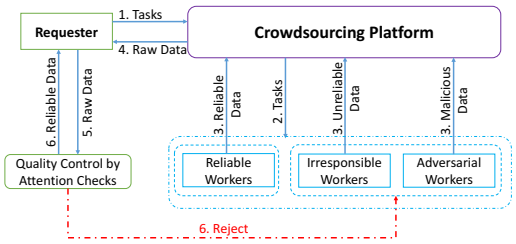


Figure 3: Crowdsourcing Platform with Quality Control.

very interested in compromising or manipulating crowdsourcing services. Besides, even legitimate workers may exhibit malicious behaviors in crowdsourcing [13]. Yao et al. [67] identified crowdturfing attacks which can generate fake comments automatically in online review systems. Our work differs from theirs from multiple perspectives such as the attack goals and attack techniques. Mao et al. [37] designed attacks targeting at the aforementioned Dawid-Skene model [10], which is an important aggregation method used for deriving the true labels. Unlike ours, their attacks target at crowdsourcing tasks that have the ground-truth for each item.

Checco et al. [6] proposed an attack for identifying gold questions by using a group of colluding crowd workers. They focused on identifying gold questions without providing correct answers, while our work focuses on attention check questions and aims to automatically answer them. Kharraz et al. [27] proposed Surveylance to identify websites involved in survey scam services. In this case, the attacks are triggered by malicious survey designers, while in our work the attacks are triggered by malicious workers.

3 DESIGN OF AC-EASYPASS

3.1 Threat Model

Crowdsourcing lets requesters leverage the crowd's intelligence and wisdom to solve problems. Figure 3 illustrates how a typical crowdsourcing platform with quality control works. First, requesters would create and publish their tasks on the crowdsourcing platform. Then the crowdsourcing platform would distribute the tasks to workers and aggregate the data provided by the workers who completed the tasks. Finally, the aggregated data would be sent back to the requesters for them to filter out unreliable data by using some quality control mechanism.

Let's consider an example scenario in which a requester posts a survey task on the MTurk crowdsourcing platform and some workers (respondents) take this survey. The posted survey is to collect public opinions on a specific political topic, and all its questions are Likert-scale statements on which workers are asked to indicate their agreement or disagreement levels. To increase the quality of the survey, the requester randomly embeds an attention check question with the instruction "Please select Completely Agree" into the sequence of the questions presented to each worker.

Reliable workers would pay attention to the survey and pass this attention check question easily. However, irresponsible workers and adversarial workers are likely to fail this attention check. Irresponsible workers want to rush through the survey to get paid quickly and get more pay by completing more tasks, so they may randomly select their responses to all the questions. Adversarial

workers may always select negative options such as "Strongly Disagree" or "Disagree" for questions in this survey aiming to influence the potential political decision. After the requester gathers all the raw survey data from workers, unreliable responses (i.e., those without selecting "Completely Agree" for the attention check question) will be filtered out. Those workers who selected other options for this attention check question would further get their submissions rejected, get their rewards to this task denied, and get their qualification deteriorated (e.g., with the increased rejection rate).

Therefore, both irresponsible workers and adversarial workers have the strong desire to pass attention check questions, so that they can be continuously qualified and can continuously achieve their financial or political goals. In this paper, we consider both irresponsible workers and adversarial workers as *attackers* because their data are not reliable (e.g., inaccurate, invalid, untrustworthy, or even harmful) to survey requesters, and they are essentially compromising the integrity of the corresponding studies. They can of course manually answer those attention check questions; however, *taking an automated approach will enable attackers to maximize their gains while still maintaining undetected.* This will be especially true if other survey questions are largely subjective and do not have the ground-truth answers.

3.2 Automated Answer Selection Approaches

In this paper, we focus on investigating attention check questions that provide multiple choices, among which one is typically the correct answer. Attackers aim to automatically analyze an attention check question and derive the correct answer. We formulate this problem as an *answer selection problem*.

One important characteristic of attention check questions is that the correct answer is hidden in the question (Section 2). A simple approach to selecting the correct answer is by word matching. For example, Word Count and Weighted Word Count are two representative matching methods [66]. However, these methods do not work well for attention check questions that do not have the option words appearing in the question, such as "Which of the following is a vegetable?" with options "Egg", "Steak", "Peach", and "Asparagus".

More advanced answer selection approaches exist. For example, the lexical semantic approach [69] finds the answer option by pairing the words (in questions and answers) that are semantically related. This approach is not ideal for answering attention check questions either. First, attention check questions are often diverse in length. For example, an instructed-response item question may have less than 10 words, while an IMC question may have more than 100 words. So it would be burdensome and error-prone for this approach to analyze each sentence in a lengthy question. Second, most attention check questions provide words or phrases instead of sentences as their candidate answers. So it is difficult for this approach to extract lexical semantics from short candidate answers.

3.3 Proposed AC-EasyPass Model

To build an automatic answer selection model that is appropriate for attention check questions, we take the deep learning approach. One significant advantage of deep learning approach over the traditional machine learning approach is that it works directly on raw data to eliminate the need of tremendous manual feature extraction

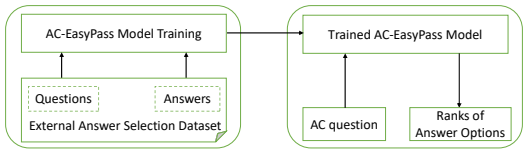


Figure 4: AC-EasyPass Framework.

effort. We propose AC-EasyPass, an attack framework with a concrete model, to easily answer attention check questions. Figure 4 illustrates the high-level structure of AC-EasyPass. In the left-side training component, a concrete AC-EasyPass machine learning model is trained for automatically extracting features from questions and answers as well as ranking candidate answer options based on the similarities between questions and answers. In the right-side testing component, the trained model analyzes each question and selects its top-ranked answer option as the final answer.

However, the biggest challenge to our approach and the AC-EasyPass framework is the lack of a large attention check question dataset for training. We address this challenge by ***training the AC-EasyPass model using an external answer selection dataset*** WikiQA [66], and more specifically, we only use its training dataset which includes 2,118 questions and 20,360 sentences. In our evaluation of the trained AC-EasyPass model, we test it on our constructed attention check question dataset as detailed in Section 4.1.1.

3.3.1 Training Dataset. Several answer selection related benchmark datasets exist such as WikiQA [66], MCTest [47], MovieQA [56], and InsuranceQA [12]. We choose WikiQA as our training dataset for three reasons. First, WikiQA is an open domain answer selection dataset while other datasets are very domain specific. Second, similar to attention check questions, WikiQA questions often contain the correct answers in short sentences, while other datasets may have very lengthy paragraphs (much longer and complicated than the description in IMCs) to contain the answers. Note that in this work, we consider the description in an IMC as a part of the question. Third, the size of the WikiQA dataset is large enough to train our deep learning model and make it converge.

3.3.2 AC-EasyPass Model. Recently, there have been significant advances in answer selection tasks [31, 69, 72]. Especially, researchers have leveraged convolutional neural networks (CNNs) to build machine comprehension models, and shown that CNNs are more effective and efficient than recurrent neural networks (RNNs) in answer selection tasks [59, 70, 71]. Yin et al. [70] proposed the BCNN and ABCNN models that analyze sentence pairs to rank answer options for the answer selection problem. BCNN constructs the representations for different levels of the sentences by using convolution and pooling operations. Compared with BCNN, ABCNN adopts an attention architecture to reweight the representations of sentences. The key idea of their attention architecture is to leverage learnable attention matrices to reweight the representations. Their attention matrices have fixed shapes, thus are more suitable for modeling the sentences with the similar length. Note that in this section, attention architecture and matrices are related to the attention mechanism in neural networks [1] that helps improve the model accuracy; they are not about the "attention" check questions.

Inspired by the BCNN and ABCNN models, we propose our AC-EasyPass model to also leverage CNNs for extracting features from questions and answers as shown in Figure 5. Our AC-EasyPass

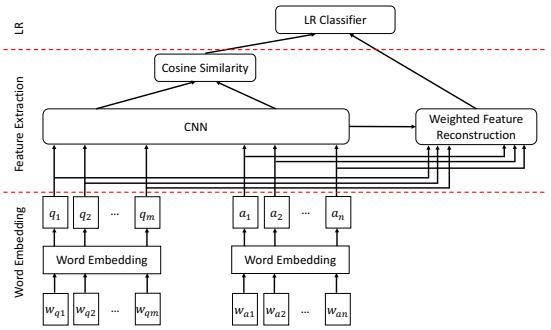


Figure 5: AC-EasyPass Model.

model has three layers: word embedding layer, feature extraction layer, and classifier layer. Words of questions and answers are mapped to vectors of real numbers in the word embedding layer. Then the feature extraction layer extracts features from both a CNN and a weighted feature reconstruction component. Based on the features derived from the feature extraction layer, the final logistic regression (LR) layer identifies the correct answer.

There are two main differences between our proposed model and the BCNN/ABCNN models. First, we do not include the sentence lengths (either the question length or the answer length) information as our additional features because attention check questions have a large range of length values (i.e., too distinct to be useful) and answer options are largely words or phrases with very similar length values (i.e., too resembling to be useful). Our experimental results confirmed that adding sentence length information as additional features makes no improvement in performance. Second, we propose a *Weighted Feature Reconstruction* component to extract additional features by reconstructing the sentence representations. Many attention check questions are lengthy (especially in IMCs), and it is difficult for a model to concentrate on the key information of a lengthy question. To capture the most relevant information in a question, we reconstruct the sentence representations based on the distance between the question and each answer option. Our weighted feature reconstruction component also leverages attention matrices to reweight the representations of sentences. However, different from that in ABCNN, we do not need to train those attention matrices, and the shapes of our matrices are not fixed but varying for different questions.

Let a question be a sequence $(w_{q1}, w_{q2}, w_{q3}, ..., w_{qm})$ of m words, and a candidate answer be a sequence $(w_{a1}, w_{a2}, w_{a3}, ..., w_{an})$ of n words. We now describe how our AC-EasyPass model automatically processes a question and a candidate answer in three layers.

Word Embedding Layer. In this layer, words of a question and a candidate answer are converted to informative vectors. We use the pre-trained *word2vec* embeddings [38] to represent each word w_i as a d_0-dimension vector, where $d_0 = 300$ is the value chosen in [38]. As a result, we can use $\mathbf{Q} = (q_1, q_2, q_3, ..., q_m) \in R^{d_0 \times m}$ and $\mathbf{A} = (a_1, a_2, a_3, ..., a_n) \in R^{d_0 \times n}$ to formally represent the question and the candidate answer, respectively.

Feature Extraction Layer. In this layer, we extract features from both a CNN and the weighted feature reconstruction component. In the CNN, we utilize convolution operations to model the representations of local phrases. Let $\mathbf{c_i} \in R^{wd_0}$ be the concatenated embeddings of w consecutive words, i.e., either $q_{i-w+1}, ..., q_i$ in Q or $a_{i-w+1}, ..., a_i$ in A. We generate the phrase representation $\mathbf{p_i} \in R^{d_1}$ for $\mathbf{c_i}$ in the convolution layer based on Formula (1):

$$\mathbf{p_i} = tanh(\mathbf{W}\mathbf{c_i} + \mathbf{b}) \tag{1}$$

where $\mathbf{W} \in R^{d_1 \times wd_0}$ is the learnable weights, $\mathbf{b} \in R^{d_1}$ is the bias, and d_1 is the number of filters in the CNN (which is 50 in our experiments). Then we apply the w average pooling (*w-ap*) and *all* average pooling (*all-ap*) for $\mathbf{p}$. Here *w-ap* is the average pooling with the filter width w, while *all-ap* uses the length of the sentence as the filter width. The *w-ap* models the question phrase representations $Q_{w-ap} \in R^{d_1 \times m}$ and the answer phrase representations $A_{w-ap} \in R^{d_1 \times n}$, which will be used as the input to the next convolution layer. The *all-ap* is used to generate two representation vectors $Q_{all-ap} \in R^{d_1}$ and $A_{all-ap} \in R^{d_1}$ for the question and the candidate answer, respectively. The cosine similarity between these two representation vectors, denoted as F_{cnn} shown in Formula (2), will be used as the feature for the final layer:

$$\mathbf{F_{cnn}} = cos\_sim(Q_{all-ap}, A_{all-ap}) \tag{2}$$

For the weighted feature reconstruction, we reconstruct a sentence vector for the question and the candidate answer, respectively. We generate the attention matrix based on the Euclidean distance between two words. We calculate the distance-based attention matrix $\mathbf{M} \in R^{m \times n}$ as shown in Formula (3):

$$\mathbf{M_{ij}} = \frac{1}{1 + ||q_i - a_j||} \tag{3}$$

where $||\cdot||$ is the Euclidean distance, and $\mathbf{M_{ij}}$ represents the attention value between the ith word in Q and the jth word in A. The smaller the Euclidean distance between two words, the larger the attention value. We then reconstruct Q and A based on the attention matrix M as shown in Formulas (4) and (5):

$$\mathbf{Q'} = \mathbf{A}f(\mathbf{M^T}) \tag{4}$$

$$\mathbf{A'} = \mathbf{Q}f(\mathbf{M}) \tag{5}$$

where $f(\cdot)$ is the column-wise softmax function. In this step, we reconstruct the question representation $\mathbf{Q'} \in R^{d_0 \times m}$ in the answer's space, and reconstruct the answer representation $\mathbf{A'} \in R^{d_0 \times n}$ in the question's space.

Finally, we apply max pooling for each constructed representation, and calculate the similarity between the question and the candidate answer in the same space as shown in Formulas (6) and (7):

$$\mathbf{F_Q} = cos\_sim(max\_pool(\mathbf{Q}), max\_pool(\mathbf{A'})) \tag{6}$$

$$\mathbf{F_A} = cos\_sim(max\_pool(\mathbf{Q'}), max\_pool(\mathbf{A})) \tag{7}$$

where $\mathbf{F_Q}$ represents the cosine similarity between the question and the candidate answer in the question's space, while $\mathbf{F_A}$ represents the similarity in the answer's space. With the max pooling operation instead of average pooling, we would filter out the less significant information and retain the more important information.

Logistic Regression (LR) Classifier Layer. All the features obtained from the previous feature extraction layer will be the inputs to the LR classifier layer. All the candidate answers will be ranked based on their probability to be the correct answer.

4 EVALUATION OF AC-EASYPASS

4.1 Setup of the Experiments

4.1.1 Datasets. Our dataset consists of three sub-datasets: AC-Original, Ans-Augmented, and Ques-Augmented. AC-Original consists of 115 original attention check questions that we collected largely from MTurk. To more comprehensively evaluate AC-EasyPass, we further propose two methods to augment AC-Original, and construct Ans-Augmented and Ques-Augmented. We use lowercasing and stop word removal to preprocess our training dataset and these testing datasets.

AC-Original dataset. The AC-Original dataset is constructed mainly by logging into MTurk as a worker and searching for survey HITs periodically with about two times per week. Two native English speakers who are not involved in the design of AC-EasyPass performed this search and then manually browsed over 1,000 HITs. They read all the questions in each survey to identify unique attention check questions. Roughly 25% of the surveys contain attention check questions, and they are largely related to important studies such as psychological, political, and marketing research. We notice that the diversity of the attention check question in practice is limited. Between July 2018 and January 2019, 91 unique attention check questions are collected from MTurk. We further included another 24 attention check questions mentioned in literature [18, 22, 42] and used in online forums like Reddit [46]. And to avoid overfitting, those representative and effective attention check questions are only used for testing (and the model is trained on a general answer selection dataset, the WikiQA). In total, AC-Original consists of 115 unique questions, among which 34 are IMCs and 81 are instructed-response items. The average length of IMCs is 102 words, while that of instructed-response items is only 12 words. Some questions have multiple correct answers.

Ans-Augmented dataset. This dataset is constructed by using our answer-based augmentation method that creates new questions based on the diversity of the answer options. In this method, we replace the correct answer that is mentioned in the question with another candidate answer. For example, for the attention check question "Please answer Rarely to this question" with the answer options ("Never", "Rarely", "Occasionally", "Almost every time"), we could replace the correct answer "Rarely" in the question with another answer option "Never" to derive a new question "Please answer Never to this question". Depending on the number of answer options in each original question, we can derive multiple new questions. Based on the AC-Original dataset, we eventually derived 442 new attention check questions in the Ans-Augmented dataset.

Ques-Augmented dataset. This dataset is constructed by using our question-based augmentation method that creates new questions by paraphrasing the original questions. We adopt the backtranslation method [71] which leverages neural machine translation (NMT) techniques to paraphrase questions. The basic idea of this augmentation is to use two translation models, one from English to German and the other from German to English, to obtain paraphrases of the questions. The publicly available codebase provided by Luong et al. [34] has replicated Google's NMT (GNMT) system [64]. So we utilize a pretrained 4-layer GNMT model provided by this codebase on 4.5 million English-German sentence pairs [63] to paraphrase questions. We obtain four augmented questions for each original question from the AC-Original dataset, and combine those augmented questions with the original answer options as our Ques-Augmented dataset. Considering the fact that backtranslation is not perfect and sometimes causes grammatical errors or information absence, we revise the augmented questions to ensure that their expressions are correct and their correct answers can be easily identified by us. To simplify our augmentation and guarantee its precision, we do not use those questions with the length greater than 160. Finally, we derived 424 new attention check questions in the Ques-Augmented dataset.

4.1.2 Reference Methods for Comparison. We compare our proposed AC-EasyPass model with two baseline methods *Baseline_fixed* and *Baseline_rand* as well as a reference method BCNN.

In the *Baseline_fixed* method, a survey respondent simply selects a specific such as the first option for all questions. This method requires the least effort from attackers and is easy to implement by them. Attackers may also adopt this basic method to achieve some specific purposes. For example, they may select the first answer option such as "Strongly Disagree" for all the questions to damage the reputation of a company or product. In our experiments, we implement this method by selecting the first option as the answer.

In the *Baseline_rand* method, a survey respondent simply selects a random option for each question. This is a common method for respondents who want to rush through a survey with less effort and want to avoid being detected as abnormal. Attackers can easily implement and use it to manipulate the overall survey results.

We also implement the BCNN model proposed by Yin et al. [70] as a reference method for comparison. Our implemented BCNN model achieves an almost identical performance as theirs on the WikiQA dataset. We could not implement ABCNN as a reference because its parameters are not described in their paper and their code is not publicly available.

4.1.3 Metrics for Evaluation. For evaluation, we rank answer options based on their predicted scores for being the correct answer. We leverage *mean average precision (MAP)* and *mean reciprocal rank (MRR)* as the metrics for evaluation, which are commonly used in the answer selection related research [45, 57, 65]. The definition of MRR is shown in Formula (8):

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{8}$$

where $|Q|$ is the number of the questions and $rank_i$ is the rank position of the first correct answer for the i-th question. MRR is based on the rank position of the first correct answer, thus it is suitable for questions with only one correct answer. However, we have questions that have more than one correct answer, so we also consider MAP as shown in Formula (9):

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}AP(i) \tag{9}$$

where $AP(i)$ is the average precision similarly defined as that in [28] for the i-th question. MAP takes all correct answers into account, thus it is suitable for all multiple-choice questions.

We also report the results using the traditional *accuracy* metric, which is defined as the percentage of attention check questions that a model ranks the correct answer as the top option.

4.2 Effectiveness of AC-EasyPass

To evaluate our AC-EasyPass model, we compare it with the two baseline methods and the BCNN model on three datasets: AC-Original, Ans-Augmented, and Ques-Augmented. Table 1 provides the detailed evaluation results.

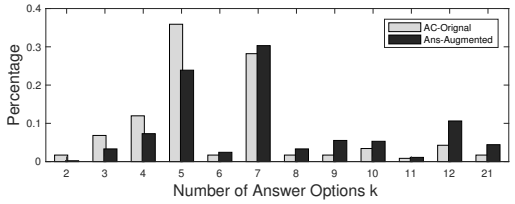
Overall Results and Analysis. We can see that on the AC-Original dataset, *Baseline_fixed* achieves the worst performance, which means that selecting the same option (the first one in our experiments) is not a good strategy to pass the attention check, with only 13.91% of attention check questions being passed. Although *Baseline_rand* achieves a better result in comparison with *Baseline_fixed*, its performance is still quite poor with around 0.42 in both MAP and MRR. BCNN achieves 0.7889 MAP and 0.7901 MRR on the AC-Original dataset, which confirms that it is feasible to leverage machine comprehension techniques to pass attention check questions with high accuracy. Our AC-EasyPass model achieves the best performance with 0.8442 MAP and 0.8483 MRR on the AC-Original dataset in which 75.65% of questions are passed, far surpassing the baseline methods and also outperforming the BCNN model. On the two augmented datasets constructed from the AC-Original dataset, AC-EasyPass still outperforms those three methods, which further validates the effectiveness of our approach.

Performance Discrepancy on the Ans-Augmented Dataset. *Baseline_rand*, BCNN, and AC-EasyPass all perform better on the AC-Original dataset and the Ques-Augmented dataset than on the Ans-Augmented dataset. This discrepancy is mainly caused by the unbalanced augmentation in the Ans-Augmented dataset, in which the number of the newly derived questions depends on the number of the answer options. An attention check question that contains only one correct answer with k answer options could be used to derive $k-1$ Ans-Augmented questions. The expectation of MAP for this question is shown in Formula (10):

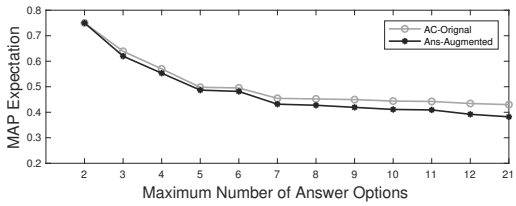
$$E_{MAP} = \frac{1}{k} \times (1 + \frac{1}{2} + ... + \frac{1}{k}) \tag{10}$$

A larger k would lead to a smaller expectation of MAP. Figure 6a shows the distribution of questions with k options in the AC-Original dataset and the Ans-Augmented dataset. We can see that the unbalanced augmentation method decreases the proportion of questions with less answer options ($k < 6$) and increases the proportion of questions with more options ($k \geq 6$). Figure 6b shows the expectation of MAP on questions with no larger than k options. This chart clearly illustrates that an unbalanced augmentation would decrease MAP because questions with larger k values would have a larger impact on the final MAP expectation result. Note that our answer-based augmentation slightly increased the percentage of the questions with the first option as the correct answer, thus the *Baseline_fixed* method which always selects the first option performs a little bit better on the Ans-Augmented dataset than on the other two datasets.

Analyzing Failure Cases. AC-EasyPass and BCNN both found the correct answers for most of attention check questions such as the aforementioned "Which of the following is a vegetable?" and "We want to test your attention, so please click on the answer Agree". We now focus on analyzing the types of the questions that AC-EasyPass and BCNN models failed to rank a ground-truth answer as the top option. On the AC-Original dataset that contains



(a) Distribution of questions with k options



(b) Expectation of MAP questions with no larger than k options

Figure 6: Comparison between the AC-Original and Ans-Augmented datasets from two different aspects: (a) distribution of questions, and (b) expectation of MAP.

115 attention check questions, our AC-EasyPass model fails on 28 (24.3%) questions. Among the 28 failure questions, 8 of them have answer options related to some number (e.g., the correct answer is the concatenated word "thirtyfour" and the option is "34"), 6 of them have lengthy descriptions (e.g., IMCs or reading comprehension given a story), 5 of them are more difficult questions that require some advanced comprehension ability (e.g., selecting the word start with the letter 'b'), 4 of them are caused by some very distracting information embedded in the questions, and the rest 5 are due to the limitations of the model on discerning subtle differences such as between "strongly agree" and "agree" sometimes. The BCNN model has 39 (33.9%) failure cases, among which 27 cases also fail AC-EasyPass. However, 12 failure questions for the BCNN model are correctly answered by our AC-EasyPass model. The BCNN model mainly focuses on learning phrase representations by leveraging the average pooling operation, which may make it fail to handle the answer options that are not phrases. For example, it fails on questions such as "please select 4 as your answer" with the options "1", "2", "3", and "4". Besides, due to the small edit distance between "agree" and "disagree", the BCNN model also fails to correctly answer questions such as "please choose the strongly disagree option for this question" with the options "strongly agree", "agree", "disagree", and "strongly disagree". Our AC-EasyPass model answers these questions correctly with its Weighted Feature Reconstruction in which the significant information would be extracted by leveraging the max pooling operation. These results validate the better performance of AC-EasyPass over BCNN.

4.3 Usefulness of Weighted Feature Reconstruction

Our proposed weighted feature reconstruction component aims to filter out less significant and retain more important information so that the ground-truth answer would be closer to the question than other wrong options. We now explore how this component contributes to the answer selection results. We first analyze the similarities between a question and the answer options before and

Table 1: AC-EasyPass Evaluation Results on Three Datasets.

	AC-Original			Ans-Augmented			Ques-Augmented		
Method	MAP	MRR	Accuracy	MAP	MRR	Accuracy	MAP	MRR	Accuracy
Baseline_fixed	0.3851	0.3877	0.1391	0.3979	0.4016	0.1719	0.3787	0.3861	0.1439
Baseline_rand	0.4231	0.4264	0.2043	0.3960	0.3978	0.1672	0.4146	0.4212	0.1995
BCNN	0.7889	0.7901	0.6609	0.7262	0.7270	0.5837	0.8078	0.8101	0.7028
AC-EasyPass	**0.8442**	**0.8483**	**0.7565**	**0.7969**	**0.7987**	**0.6810**	**0.8603**	**0.8661**	**0.7854**

after the feature reconstruction. Then we investigate the difference between the top-ranked option and the other options in BCNN and AC-EasyPass to compare the confidence of these two models on selecting the ground-truth answer.

Similarity rank improvement in reconstructed spaces. For a specific question, our model ranks the answer options based on their probabilities to be the ground-truth answer. The probability to be the ground-truth answer depends on the similarities between the answer options and the question. The higher the similarity value, the higher the probability for an answer option to be the ground-truth answer. We hypothesize that the reconstructed features would make the ground-truth answer closer to the question compared with other wrong options. Since we are concerned about the relative ranks of answer options for a given question, we compare the rank change of the ground-truth answer based on similarities in different spaces (original space and reconstructed spaces). We define a similarity rank improvement (SRI) metric as in Formula (11):

$$SRI = R_{original} - R_{reconstructed} \tag{11}$$

where $R_{original}$ is the rank of the ground-truth answer based on the similarity before reconstruction while $R_{reconstructed}$ is the rank after reconstruction. A positive SRI value implies that the reconstruction process makes the similarity between the question and the ground-truth answer higher than other options.

We analyze the SRI in the reconstructed question and answer spaces. In the question space, 51 (44.3%) questions keep the same similarity rank before and after the reconstruction process, 54 (47.0%) questions have a positive SRI, while 10 (8.7%) questions have a negative SRI. Among questions with a positive SRI, 27 questions rise by one in rank and 18 questions rise by two in rank. In the answer space, 52 (45.2%) questions have a positive SRI and 30 (26.1%) questions have a negative SRI. The average SRI in the question space and the answer space is 0.7217 and 0.4174, respectively, with the former greater than the latter. We used the paired sample t-test to compare the pairs of SRI in question space and answer space, and found this average SRI difference is not statistically significant. Therefore, the reconstructed features within the question and answer spaces improve the model performance almost equivalently, and we adopt both of them in our model.

The probability gap between the top-ranked option and other wrong options. A larger probability gap implies that a model has higher confidence on choosing the top-ranked option. Instead of measuring all possible combinations between the top-ranked option and other wrong options, we only need to examine the probability gap between the top-ranked and the second-ranked options, which is the lower bound of the difference between the top-ranked option and the other options. We only consider the successful cases where the ground-truth answer is selected as the top-ranked answer in BCNN and AC-EasyPass models. We first normalize the probabilities of all options so that their sum becomes one. We then define the probability gap metric as in Formula (12):

$$probability\ gap = P_{top-ranked} - P_{second-ranked} \tag{12}$$

where $P_{top-ranked}$ and $P_{second-ranked}$ are the probabilities of the top-ranked option and the second-ranked option to be the ground-truth answer, respectively.

Figure 7 shows the distributions of probability gap in BCNN and AC-EasyPass models. On average, the probability gap is 9.9% in BCNN, while it increases to 14.8% in AC-EasyPass. We used the Wilcoxon signed-rank test to compare these two distributions, and found that their difference is statistically significant. This result demonstrates that the weighted feature reconstruction process even gives AC-EasyPass more confidence than BCNN on selecting a ground-truth answer.

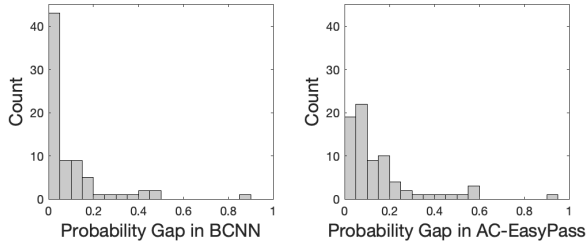


Figure 7: Probability gap of the top-ranked and second-ranked options in BCNN, and AC-EasyPass.

5 DEFENSES AGAINST AC-EASYPASS

We now explore potential defenses for improving the robustness of the attention check mechanism. That is, attackers would use our proposed AC-EasyPass or other answer selection models to automatically pass attention check questions, while our goal in this section is to make those attacks less effective in deriving correct answers. Especially, inspired by the recent research on fooling machine learning models, we explore and evaluate two simple defense methods: adding adversarial sentences and adding typos.

Recently, researchers found that machine comprehension models are vulnerable to adversarial examples [24, 33, 48]. In these research efforts, adversarial examples or sentences are created by attackers to fool the legitimate machine learning models. Inspired by them but starting from a defense perspective, we consider to defend against AC-EasyPass by adding adversarial sentences or phrases.

Besides, some researchers showed that attackers can use typos to fool machine comprehension models [19, 33, 48]. For example, "United States" in a sentence could be written as "Un1ted Sta8tes". Those words with typos are still easily recognizable by humans, but could be identified by machine comprehension models as unknown words because they are not in the dictionary and they would be mapped to random word vectors. Therefore, we consider to add

typos into attention check questions for disguising the meanings of words and decreasing the accuracy of attackers' models.

5.1 Adding Adversarial Phrases or Sentences

Jia et al. [24] demonstrated that on the SQuAD dataset [44], adversarial sentences can incur an over 50% decrease to the F-measure score of the machine comprehension models. Different from the questions in the SQuAD dataset, attention check questions do not have relevant (or relevant enough) sentences that could be useful for deriving the correct answers. In more details, instructed-response items do not have any relevant sentences, while IMC questions are spurious and their sentences cannot be used to derive the final correct answers as shown in the "Please check all words that describe how you are currently feeling" example in Figure 1.

Therefore, we cannot generate adversarial sentences based on relevant sentences as Jia et al. [24] did. We instead add perturbations such as some words, phrases, or sentences as noises to distract AC-EasyPass without distorting the semantics and purpose of an attention check question. So we define three rules to follow for adding perturbations: (1) the added perturbations should not be perceptible as irrelevant information to attentive respondents; (2) adding perturbations would not change the correct answer to be identified by attentive respondents; (3) adding perturbations would likely fool AC-EasyPass to select an incorrect answer.

Guided by these rules, we come up with a concrete strategy to add perturbations, i.e., carefully including some incorrect answer option(s) into the question to distract an attacker's model without adding too much outstanding information. This strategy is applicable to both IMCs and instructed-response items, but its implementation in them is different due to the differences between these two forms of attention check questions:

IMCs. A lengthy description exists in each IMC before its question as shown in Figure 1. Most of those lengthy descriptions would give a brief introduction about answer options and provide the cue or instruction to the correct answer. We can add perturbations by smoothly mentioning some wrong option(s) in the question. Figure 8 illustrates how we add an adversarial phrase (in bold font) to the IMC example shown in Figure 1. In this example, all answer options are about "words" that describe feelings, so we mention one of wrong answers "Excited" near "words". Mentioning an incorrect option as an example for explanation not only retains the semantics of the original question but also distracts the attacker's model.

> Recent research on decision making shows that choices are affected by context..... Thank you very much.
>
> Please check all words **below such as Excited** that describe how you are currently feeling.
>
> **A.** Excited **B.** Afraid **C.** Scared **D.** None of the above

Figure 8: Adding an Adversarial Phrase (in bold font) to the IMC example shown in Figure 1. Note the first paragraph is shrunk to save space, but is identical to that in Figure 1.

Instructed-response Items. Instructed-response items are more straightforward than IMCs. They do not have lengthy descriptions, so it is hard to find related information about options as in IMCs. In this case, we add a short descriptive sentence about how to complete a task. Figure 9 illustrates how we add an adversarial sentence (in bold font) to the instructed-response item example shown in Figure 2. The added sentence contains one of the incorrect answer options "Disagree" to distract the attacker's model.

> **Please click on one of options such as Disagree.** We want to test your attention, so please click on the answer Agree.
>
> **A.** Strongly disagree **B.** Disagree **C.** Neutral **D.** Agree
> **E.** Strongly agree

Figure 9: Adding an Adversarial Sentence (in bold font) to the Instructed-response Item example shown in Figure 2.

5.2 Adding Typos

Pre-trained word embeddings have been shown to boost the performance in natural language processing tasks. However, any typo would make a word be identified as an unknown word, and would fail the mapping from the word to a unique and meaningful word embedding. Our goal is to add simple typos to fool an attacker's model while still making it easy for humans to ignore the typos. Specifically, we add typos to some keywords of a question that are significant to the derivation of the correct answer; meanwhile, we only change one letter in each keyword by replacing it with a similar character. However, not all letters have similar characters, so we give a high priority to the letters that do have similar characters. Table 2 lists some of the high-priority letters that we identified.

Table 2: Some High-priority Letters and their Replacements.

Original Letter	Similar Character	Replacement Example
q	9	question → 9uestion
o	0	other → 0ther
z	2	zero → 2ero
l	1	select → se1ect
u	v	true → trve
s	5	classified → cla5sified

Besides adding a typo, we also add "(forgive the typos)" at the end of the question to help lessen the possible ambiguity introduced by the typo to a respondent. In comparison with adding adversarial sentences or phrases, adding typos requires less effort from a requester to design an attention check question, but may require more effort for a respondent to ignore the typos. So both methods have pros and cons.

5.3 Evaluation of the Two Defense Methods

We evaluate the two defense methods by comparing the performance of AC-EasyPass on the AC-Original dataset with that on two variations of the AC-Original dataset (one with an adversarial sentence or phrase added, called the AC-Original-Adversarial dataset, and the other with a typo added, called the AC-Original-Typos dataset, for each attention check question in the AC-Original dataset). Table 3 lists the evaluation results.

Overall Results and Analysis. We can see that both methods can to some extent decrease the accuracy of our AC-EasyPass attacks. Adding adversarial sentences contributes to an over 10%

Table 3: Effectiveness of the Two Defense Methods on Decreasing AC-EasyPass Performance.

Dataset	MAP	MRR	Accuracy
AC-Original	0.8442	0.8483	0.7565
AC-Original-Adversarial	0.7144	0.7178	0.5478
AC-Original-Typos	0.5247	0.5326	0.2957

decrease in both MAP and MRR, while adding typos leads to a more than 30% decrease in both MAP and MRR. Adding typos outperforms adding adversarial sentences, and one reason is that typos directly affect the correct answer while adversarial sentences just distract AC-EasyPass from paying attention to the correct answer.

Analyzing Failure Cases. As mentioned in Section 4.2, there are 28 failure questions on the AC-Original dataset. It is not a surprise that 26 of those 28 failure questions are still failure questions after being added with adversarial sentences or typos. On the AC-Original-Adversarial dataset, AC-EasyPass now fails on 52 (45.2%) out of 115 questions. Excluding those 26 failure questions and for 17 of the rest 26 failure questions, AC-EasyPass selects the wrong option mentioned in an adversarial sentence as the top-ranked option. On the AC-Original-Typos dataset, AC-EasyPass now fails on 81 (70.4%) out of 115 questions. Excluding those 26 failure questions, 55 failure cases are caused by the added typos. Both defense methods can be useful in decreasing the accuracy of AC-EasyPass.

5.4 Limitations of the Two Defense Methods

Although adding adversarial sentences and adding typos could to some extent decrease the accuracy of our AC-EasyPass attacks, these two defense methods are fragile due to their limitations from both technical and usability perspectives.

For adding adversarial sentences, this defense method will become less effective if attackers include some adversarial sentences to train AC-EasyPass and improve its robustness. This is similar to the approach of adversarial training [15, 55], but is now performed by attackers instead of defenders. We conducted some preliminary experiments to evaluate the effectiveness of adversarial training. We first apply our adding adversarial sentences method to the Ques-Augmented dataset to generate an adversarial dataset, called the Ques-Augmented-Adversarial dataset. Then this dataset is added to the WikiQA training dataset for adversarial training. Our adversarially trained AC-EasyPass model can now obtain 75.41% MAP, 76.07% MRR, and 60.87% accuracy on the AC-Original-Adversarial dataset. These results show that even a simple adversarial training can help AC-EasyPass regain (to certain extent) the accuracy lost to the adding adversarial sentences defense method. Meanwhile, if not properly designed and tested, adversarial sentences may confuse humans and decrease the accuracy of attentive survey respondents on answering attention check questions.

For adding typos, attackers can leverage spelling check techniques to correct those typos and improve the robustness of AC-EasyPass. To evaluate the effectiveness of this idea, we used a context aware spelling check service provided by Microsoft Azure [53] to correct typos. Our experiments show that, 59.7% of the questions in the AC-Original-Typos dataset can be completely corrected by this single spelling check service, while 8.4% of the questions can be partially corrected; therefore, this second defense method becomes less effective too. Moreover, while our added typos could be easily ignored by humans, they are still perceptible to humans and may negatively affect the survey answering process. This is perhaps the biggest limitation of the adding typos defense method.

6 DISCUSSION

Based on the analysis of failure cases in Section 4.2, attackers may include more answer selection datasets besides WikiQA to train AC-EasyPass. This is because more comprehensive training data can often help improve the generalization ability of neural network models. For the same reason and based on the experiments in Sections 5.3 and 5.4, attackers can always perform adversarial training to improve the robustness of AC-EasyPass. In addition, some attention check questions repeatedly occur in different surveys as we observed in our data collection process; this characteristic can be leveraged by attackers to collectively identify attention check questions in multiple surveys similar to what Checco et al. did in identifying gold questions by using colluding crowd workers [6].

In terms of the defense, using CAPTCHAs as a solution is not desirable because less difficult CAPTCHAs could be easily compromised by advanced solvers [14, 40, 68] while more difficult ones would cause selection bias and incur usability concerns [5]. We may also ask if we can use questions that AC-EasyPass could not correctly answer as reliable attention check questions. This may not be a long-term solution because attackers can incorporate those questions to further train their models. On the other hand, non-intrusive bot detection techniques such as anomaly detection based on user behavior profiling could be a potential defense.

It is crucial to defend against AC-EasyPass or similar attack models to protect this important quality control mechanism, yet it seems that currently everything strongly favors attackers. Researchers need to design effective defense schemes to protect the existing attention checking mechanism, or perhaps need to design new attention checking mechanisms that are secure and usable.

7 CONCLUSION

In this paper, we performed the first study to investigate the vulnerabilities of the attention check mechanism. We proposed AC-EasyPass, an attack framework with a concrete model, that combines convolutional neural network and weighted feature reconstruction to easily pass attention check questions. We constructed the first attention check question dataset that consists of both original and augmented questions, and demonstrated that AC-EasyPass is effective on those questions. We also explored two simple defense methods, adding adversarial sentences and adding typos, for survey designers to mitigate the risks posed by AC-EasyPass. However, these two defense methods are fragile due to their limitations from both technical and usability perspectives, underlining the challenging nature of the defense task. We hope that our work will raise sufficient attention of the research community towards developing more robust attention check mechanisms. More broadly, our work intends to prompt the research community to seriously consider the emerging risks posed by the malicious use of machine learning techniques to the quality, validity, and trustworthiness of crowdsourcing and social computing.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their valuable suggestions. This research was supported in part by the NSF grant OIA-1936968.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[2] Adam J Berinsky, Michele F Margolis, and Michael W Sances. 2014. Separating the shirkers from the workers? Making sure respondents pay attention on self-administered surveys. *American Journal of Political Science* 58, 3 (2014), 739–753.

[3] BotHitMTurk 2018. A Bot Panic Hits Amazon's Mechanical Turk | WIRED. https://www.wired.com/story/amazon-mechanical-turk-bot-panic/.

[4] Nathan A Bowling, Jason L Huang, Caleb B Bragg, Steve Khazon, Mengqiao Liu, and Caitlin E Blackmore. 2016. Who cares and who is careless? Insufficient effort responding as a reflection of respondent personality. *Journal of Personality and Social Psychology* 111, 2 (2016), 218.

[5] Elie Bursztein, Steven Bethard, Celine Fabry, John C Mitchell, and Dan Jurafsky. 2010. How good are humans at solving CAPTCHAs? A large scale evaluation. In *2010 IEEE symposium on security and privacy*. IEEE, 399–413.

[6] Alessandro Checco, Jo Bates, and Gianluca Demartini. 2018. All That Glitters Is Gold?An Attack Scheme on Gold Questions in Crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*.

[7] Scott Clifford and Jennifer Jerit. 2015. Do attempts to improve respondent attention increase social desirability bias? *Public Opinion Quarterly* 79, 3 (2015), 790–802.

[8] Paul G Curran. 2016. Methods for the detection of carelessly invalid responses in survey data. *Journal of Experimental Social Psychology* 66 (2016), 4–19.

[9] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 7.

[10] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[11] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. 1013–1022.

[12] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. *arXiv preprint arXiv:1508.01585* (2015).

[13] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 2015. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the Annual ACM Conference on Human Factors in Computing Systems*. 1631–1640.

[14] Haichang Gao, Jeff Yan, Fang Cao, Zhengya Zhang, Lei Lei, Mengyun Tang, Ping Zhang, Xin Zhou, Xuqin Wang, and Jiawei Li. 2016. A Simple Generic Attack on Text Captchas.. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[16] Joseph K Goodman, Cynthia E Cryder, and Amar Cheema. 2013. Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples. *Journal of Behavioral Decision Making* 26, 3 (2013), 213–224.

[17] Sandy J. J. Gould, Anna L. Cox, and Duncan P. Brumby. 2016. Diminished Control in Crowdsourcing: An Investigation of Crowdworker Multitasking Behavior. *ACM Trans. Comput.-Hum. Interact.* 23, 3 (June 2016), 19:1–19:29.

[18] David J Hauser and Norbert Schwarz. 2016. Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior research methods* 48, 1 (2016), 400–407.

[19] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving Google's Perspective API Built for Detecting Toxic Comments. *arXiv preprint arXiv:1702.08138* (2017).

[20] Jason L Huang, Nathan A Bowling, Mengqiao Liu, and Yuhui Li. 2015. Detecting insufficient effort responding with an infrequency scale: Evaluating validity and participant reactions. *Journal of Business and Psychology* 30, 2 (2015), 299–311.

[21] Jason L Huang, Paul G Curran, Jessica Keeney, Elizabeth M Poposki, and Richard P DeShon. 2012. Detecting and deterring insufficient effort responding to surveys. *Journal of Business and Psychology* 27, 1 (2012), 99–114.

[22] Qatrunnada Ismail, Tousif Ahmed, Kelly Caine, Apu Kapadia, and Michael Reiter. 2017. To permit or not to permit, that is the usability question: Crowdsourcing mobile apps' privacy permission settings. *Proceedings on Privacy Enhancing Technologies* 2017, 4 (2017), 119–137.

[23] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. 2017. Identifying unreliable and adversarial workers in crowdsourced labeling tasks. *Journal of Machine Learning Research* 18, 1 (2017), 3233–3299.

[24] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2021–2031.

[25] Chester Chun Seng Kam and Gabriel Hoi-huen Chan. 2018. Examination of the validity of instructed response items in identifying careless respondents. *Personality and Individual Differences* 129 (2018), 83–87.

[26] Jeremy Kees, Christopher Berry, Scot Burton, and Kim Sheehan. 2017. An analysis of data quality: Professional panels, student subject pools, and Amazon's Mechanical Turk. *Journal of Advertising* 46, 1 (2017), 141–155.

[27] Amin Kharraz, William Robertson, and Engin Kirda. 2018. Surveylance: Automatically Detecting Online Survey Scams. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. 70–86.

[28] Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

[29] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. 1301–1318.

[30] Franki YH Kung, Navio Kwok, and Douglas J Brown. 2018. Are Attention Check Questions a Threat to Scale Validity? *Applied Psychology* 67, 2 (2018), 264–283.

[31] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. 1188–1196.

[32] Hongwei Li and Bin Yu. 2014. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086* (2014).

[33] J Li, S Ji, T Du, B Li, and T Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. In *26th Annual Network and Distributed System Security Symposium*.

[34] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural Machine Translation (seq2seq) Tutorial. *https://github.com/tensorflow/nmt* (2017).

[35] MarketResearch 2019. Market research industry - Current stats and future trends | QuestionPro. https://www.questionpro.com/blog/market-research-stats-and-trends/.

[36] Adam W Meade and S Bartholomew Craig. 2012. Identifying careless responses in survey data. *Psychological methods* 17, 3 (2012), 437.

[37] Chenglin Miao, Qi Li, Lu Su, Mengdi Huai, Wenjun Jiang, and Jing Gao. 2018. Attack under Disguise: An Intelligent Data Poisoning Attack Mechanism in Crowdsourcing. In *Proceedings of the World Wide Web Conference*. 13–22.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in neural information processing systems*. 3111–3119.

[39] Asako Miura and Tetsuro Kobayashi. 2016. Survey satisficing inflates stereotypical responses in online experiment: The case of immigration study. *Frontiers in psychology* 7 (2016), 1563.

[40] Greg Mori and Jitendra Malik. 2003. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 1. IEEE, I–I.

[41] MTurk 2018. Amazon Mechanical Turk (MTurk). https://www.mturk.com.

[42] Daniel M Oppenheimer, Tom Meyvis, and Nicolas Davidenko. 2009. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology* 45, 4 (2009), 867–872.

[43] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. Running experiments on amazon mechanical turk. *udgment and Decision Making* 5, 5 (2010), 411–419.

[44] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[45] Jinfeng Rao, Hua He, and Jimmy Lin. 2017. Experiments with convolutional neural network models for answer selection. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 1217–1220.

[46] Reddit 2019. Reddit. https://www.reddit.com/.

[47] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 193–203.

[48] Suranjana Samanta and Sameep Mehta. 2018. Generating adversarial text samples. In *Proceedings of the European Conference on Information Retrieval*. 744–749.

[49] Mario Schaarschmidt, Stefan Ivens, Dirk Homscheid, and Pascal Bilo. 2015. Crowdsourcing for Survey Research: where Amazon Mechanical Turks deviates from conventional survey methods. *Arbeitsberichte aus dem Fachbereich* (2015).

[50] Daniel J Simons and Christopher F Chabris. 2012. Common (mis) beliefs about memory: A replication and comparison of telephone and Mechanical Turk survey methods. *PloS one* 7, 12 (2012), e51876.

[51] Scott M Smith, Catherine A Roster, Linda L Golden, and Gerald S Albaum. 2016. A multi-group analysis of online survey respondent data quality: Comparing a regular USA consumer panel to MTurk samples. *Journal of Business Research* 69, 8 (2016), 3139–3148.

[52] Ianna Sodré and Francisco Brasileiro. 2017. An analysis of the use of qualifications on the Amazon mechanical Turk online labor market. *Computer Supported Cooperative Work* 26, 4-6 (2017), 837–872.

[53] SpellCheckMicrosoft 2018. Spell Check | Microsoft Azure. https://azure.microsoft.com/zh-cn/services/cognitive-services/spell-check/.

[54] Peng Sun and Kathryn T Stolee. 2016. Exploring crowd consistency in a mechanical turk survey. In *2016 IEEE/ACM 3rd International Workshop on CrowdSourcing in Software Engineering (CSI-SE)*. IEEE, 8–14.

[55] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[56] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding Stories in Movies through Question-Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[57] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Hyperbolic Representation Learning for Fast and Efficient Neural Question Answering. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. 583–591.

[58] Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. 2011. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*. 21–26.

[59] Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747* (2016).

[60] Mary Kathrine Ward and Samuel B Pond III. 2015. Using virtual presence and survey instructions to minimize careless responding on Internet-based surveys. *Computers in Human Behavior* 48 (2015), 554–568.

[61] Alex C Williams, Joslin Goh, Charlie G Willis, Aaron M Ellison, James H Brusuelas, Charles C Davis, and Edith Law. 2017. Deja vu: Characterizing worker reliability using task consistency. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

[62] Steven L Wise and Xiaojing Kong. 2005. Response time effort: A new measure of examinee motivation in computer-based tests. *Applied Measurement in Education* 18, 2 (2005), 163–183.

[63] WMT16 2016. ACL2016 First Conference on Machine Translation (WMT16). http://www.statmt.org/wmt16/.

[64] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[65] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the ACM International on Conference on Information and Knowledge Management*. 287–296.

[66] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2013–2018.

[67] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y Zhao. 2017. Automated crowdturfing attacks and defenses in online review systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1143–1158.

[68] Guixin Ye, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. 2018. Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 332–348.

[69] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 1744–1753.

[70] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association of Computational Linguistics* 4, 1 (2016), 259–272.

[71] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv preprint arXiv:1804.09541* (2018).

[72] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. In *Proceedings of the Deep Learning and Representation Learning Workshop*.

[73] Haijun Zhai, Todd Lingren, Louise Deleger, Qi Li, Megan Kaiser, Laura Stoutenborough, and Imre Solti. 2013. Web 2.0-based crowdsourcing for high-quality gold standard development in clinical natural language processing. *Journal of medical Internet research* 15, 4 (2013).