AnswerFact: Fact Checking in Product Question Answering*

Wenxuan Zhang[†], **Yang Deng**[†], **Jing Ma**[‡] **and Wai Lam**[†]
[†]The Chinese University of Hong Kong
[‡]Hong Kong Baptist University
{wxzhang,ydeng,wlam}@se.cuhk.edu.hk
majing@comp.hkbu.edu.hk

Abstract

Product-related question answering platforms nowadays are widely employed in many E-commerce sites, providing a convenient way for potential customers to address their concerns during online shopping. However, the misinformation in the answers on those platforms poses unprecedented challenges for users to obtain reliable and truthful product information, which may even cause a commercial loss in E-commerce business. To tackle this issue, we investigate to predict the veracity of answers in this paper and introduce AnswerFact, a large scale fact checking dataset from product question answering forums. Each answer is accompanied by its veracity label and associated evidence sentences, providing a valuable testbed for evidence-based fact checking tasks in QA settings. We further propose a novel neural model with tailored evidence ranking components to handle the concerned answer veracity prediction problem. Extensive experiments are conducted with our proposed model and various existing fact checking methods, showing that our method outperforms all baselines on this task.

1 Introduction

The ability to ask questions during online shopping is found to be a key factor for customers to make purchase decisions (Smith and Anderson, 2016). To this end, product-related community question answering (PQA) platforms have emerged in many E-commerce sites such as *Amazon* and *Taobao*, allowing users to pose their concerns as questions and receive answers from fellow users to obtain useful product information. However, similar to other community question answering (CQA) platforms, the user-provided answers on PQA platforms vary significantly on their qualities (Zhang et al., 2020b), and more seriously, their veracity due to the lack of systematic quality control (Mihaylova et al., 2018). Those untruthful answers may attribute to multiple factors such as misunderstandings of the question, improper expressions during writing, and even intentionally malicious attacks from the competitors (Carmel et al., 2018). Therefore, automatically verifying the answer veracity is becoming a demanding need, which can offer a more reliable online shopping environment, for example, by triggering a double-check on the detected doubtful answers.

Question: Is this egg coker automatic shut off?
Answer (Claim): Yes there's an automatic shut-off when the cooking cycle is finished. (`Verdict : FALSE`)
Evidence:
s_1: A buzzer sounds to let you know the eggs are done, I wish it would just shut off instead.
s_2: When the alarm sounds you need to turn it off and open it.
s_3: I would have liked the cooker to turn off automatically but instead a bell rings until you turn if off.
s_4: Also, by the time the timer goes off, the hot pan has a burning smell.
s_5: And it turns off itself after the bell rings.
. . .

Table 1: An example instance in AnswerFact, where the answer is the claim to be verified. The relevant product information are provided as evidence sentences.

Fact checking aiming at verifying the truthfulness of a given claim (Thorne and Vlachos, 2018; Sharma et al., 2019) can be a promising direction to tackle the concerned problem. However, the claim on which existing fact checking methods mainly focus is usually a standalone text snippet such as news (Wang, 2017; Popat et al., 2018; Ma et al., 2019) or twitter posts (Derczynski et al., 2017; Wei et al., 2019). To predict the veracity of an answer in the QA settings, one can notice that it is insufficient to consider the answer alone since the question text

* The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719).

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2407–2417,
November 16–20, 2020.

also carries important semantic information for the prediction. Thus, we need to appropriately leverage the question text into the verification process.

In the context of CQA problems, most existing studies focus on measuring the semantic relevance of a candidate answer to the given question (Tay et al., 2017; Yang et al., 2019b) or ranking available answers for a given question (Zhang et al., 2020a). However, the notion of veracity poses a more rigorous requirement of an answer where it needs to be factually correct. For example, the given answer in Table 1 will be labeled as positive from the perspective of the typical CQA task (Nakov et al., 2017) since it is topically relevant to the question. But its verdict is indeed false which can be verified from the product description. Recently, a new shared task, namely SemEval-2019 Task 8 (Mihaylova et al., 2019) investigates the fact checking problem in question answering scenario, requiring a system to classify the veracity of answers in a web forum. However, only QA pairs are given in this task, making it less practical since most of the predictions require extra knowledge from external sources. Moreover, with only hundreds of QA pairs provided, such limited number of samples precludes its use to develop powerful machine learning based fact checking models.

To tackle the aforementioned issues, we introduce a large scale fact checking dataset called AnswerFact for investigating the answer veracity in product question answering forums. An instance of the dataset is shown in Table 1. It consists of 60,864 answer claims, each with its veracity label derived from the community votes. Moreover, the relevant product information from product descriptions and user reviews are retrieved as evidence sentences providing the external knowledge for judging the answer veracity. Compared with existing works (Thorne et al., 2018a; Mihaylova et al., 2018), AnswerFact exhibits some unique characteristics: Firstly, different from a typical single text claim, a sentence pair (i.e., QA pair) is given in AnswerFact, indicating that the rich interaction information between the question and answer text needs to be explored and utilized. Secondly, since part of the evidence sentences come from user reviews written by ordinary users, the potential unreliability of some evidence sentences needs to be investigated and the consistency among evidence needs to be verified to uncover the common judgement towards the answer for the prediction.

We further propose AVER, an **A**nswer **V**eracity prediction model with tailored **E**vidence **R**anking modules to predict the answer veracity in PQA forums. AVER first utilizes the information from both the question and answer text to rank the evidence sentences with different gating mechanisms. An agreement-matching strategy is then employed to model the self-coherence of the evidence sentences for obtaining reliable combined evidence embeddings to verify the answer verdict. To summarize, our main contributions are as follows:

- We study the fact checking problem in product question answering. To our best knowledge, this is the first work to investigate the truthfulness of answers in E-commerce QA platforms.
- We introduce AnswerFact, a large dataset consisting of 60,864 answer claims across five product domains. Each claim comes with its veracity label and associated evidence sentences.
- We propose a novel neural model with tailored evidence ranking module to tackle the answer veracity prediction problem, which shows to outperforms all established baselines.

2 Related Work

2.1 Community Question Answering

Existing methods in community question answering (CQA) mainly focus on the answer selection task (Nakov et al., 2017; Tay et al., 2017; Yang et al., 2019b; Deng et al., 2020), where an answer is considered to be positive if it is semantically relevant to the question regardless of its veracity. Some studies further measure the quality of answers, trying to predict the answer helpfulness in PQA platforms (Zhang et al., 2020b) or ranking all available answers for a given question (Zhang et al., 2020a). One closely related work in the CQA context is a recent attempt of investigating the fact checking task in QA settings (Mihaylova et al., 2018), which was later adopted as the SemEval-2019 Task 8 (Mihaylova et al., 2019). Its goal is to classify an answer in the Qatar forum[1] into true, false or non-factual. However, only QA pairs are given in this shared task to predict the answer veracity, making it less practical due to the lack of evidence sources. Moreover, the small number of training data consisting only 495 QA pairs restricts the possibility of trying some powerful machine learning models such as deep neural networks.

[1] http://www.qatarliving.com

As pointed out in Mihaylova et al. (2019), verifying the verdict of answers in CQA requires using rich world knowledge. However, gathering relevant information as evidence can be difficult due to the open-domain nature of those questions. Compared with general CQA forums, PQA provides product-specific forums, making the evidence collection process more realistic and controllable. Also, as will be described in Section 3, the high proportion of factual type QA pairs also makes it suitable for studying the fact checking problem on PQA.

2.2 Fact Checking Datasets & Methods

Automatically predicting the veracity of claims has been extensively studied in recent years and various fact checking datasets have been released (Thorne et al., 2018a; Sharma et al., 2019; Augenstein et al., 2019). Typically, the data are collected from news checking websites such as Politifact and Snopes, where the evidence is either not given (Rashkin et al., 2017; Pérez-Rosas et al., 2018) or provided as an external URL link containing machine-unreadable format ranging from statistical tables to PDF reports (Wang, 2017). One recent trend is that evidence-based fact checking has gained more attention where datasets with well-formatted claims and evidence are adopted (Thorne et al., 2018a; Popat et al., 2018; Chen et al., 2020).

Fact checking methods are mostly tailored to specific types of datasets. Methods involving small datasets often use hand-crafted features to represent the claim (Mihaylova et al., 2018). These features are then fed into a SVM or MLP classifier to make the prediction (Baly et al., 2018). Deep learning based methods are also proposed given the existence of large datasets. The claim and evidence representations can be learned with neural networks such as recurrent neural networks (RNNs) (Rashkin et al., 2017) or convolutional neural networks (CNNs) (Wang, 2017).

However, none of these work conducts fact checking problem in QA settings with associated well-formatted evidence sentences.

3 AnswerFact Dataset Construction

We build our dataset upon a large QA collection (Wan and McAuley, 2016) crawled from *Amazon*. Five product domains with the largest number of QA pairs are selected, namely, *Electronics*, *Home and Kitchen*, *Sports and Outdoors*, *Health and Personal Care*, and *Cell Phones and Accessories*, constituting around 2.7 million QA pairs in total.

Labels	Community Votes
TRUE	$n_{up} = n_{total}$
PARTTRUE	$n_{down} < n_{up} < n_{total}$
UNSURE	$n_{down} = n_{up}$
PARTFALSE	$n_{up} < n_{down} < n_{total}$
FALSE	$n_{down} = n_{total}$

Table 2: Veracity labels from community votes. n_{up}, n_{down}, n_{total} refers to the number of upvotes, downvotes and total votes of the answer respectively.

3.1 Factual QA Pairs Filtering

The raw data collection contains various questions spanning from questions asking for product details to personal user experience. Since it can be difficult to verify the truthfulness of answers to subjective questions given the diversity of user experience, we focus on factual QA pairs to investigate the answer veracity. We begin by manually labeling the factual types of two thousand randomly sampled questions, judging whether the answer will vary from user to user. For example, questions asking for product attributes are judged as FACTUAL since the answers are objective facts. Questions looking for personal experience are treated as NONFACTUAL since their answers depend on users' own experience and vary from person to person. Each question is labeled by two annotators and the disagreements are settled by discussions. From the annotation, we found that factual questions are actually the dominant type in PQA forums where around 71% of the annotated questions are factual ones.

Following the strategy in Syed et al. (2019) which ranked first for predicting the question type in SemEval-2019 Task 8, we applied the Universal Sentence representation (Cer et al., 2018) to encode question texts. While we found that the SVM classifier performs slightly better than the XGBoost (Chen and Guestrin, 2016) used in their work, achieving average 0.85 accuracy and 0.90 F1 score under the 5-fold cross validation. We then trained the SVM classifier on the whole 2k annotated questions for predicting the type of all questions. Note that since we can sacrifice some recall for the sake of precision to ensure that the questions we want are all factual ones, we discarded questions whose predicted scores are close to decision boundary. Finally, to measure the performance of such auto-filtering, we randomly sample 150 ques-

	Electronics	Home	Sports	Health	Phones	Total
# Answers per Label						
TRUE	13,054	10,592	4,539	6,879	2,467	37,531
PARTTRUE	1,737	1,297	581	1,035	336	4,986
UNSURE	3,116	2,228	1,134	1,782	738	8,998
PARTFALSE	822	683	308	564	151	2,528
FALSE	2,491	1,797	897	1211	415	6,821
# Answers	21,220	16,597	7,459	11,481	4,107	**60,864**
# Questions	11,554	8,210	3,918	5,816	2,245	**31,743**

Table 3: Summary statistics of the AnswerFact dataset

tions with their predicted types and annotate their question types again. The results showed that the precision score reached 0.99 on this set.

3.2 Veracity Labels from Community Votes

To obtain the veracity label of each answer, an intuitive way is to manually digest relevant product information to make the annotation. However, since the annotators may not be familiar with the concerned product, their annotations might be influenced by the surface level of the answer such as its writing style instead of its actual correctness. Such labeling process can also be time-consuming and difficult to collect large amounts of data. On the other hand, we observe that the community votes of each answer can be a valuable numerical indicator reflecting its veracity. Specifically, in PQA forums, each answer can receive upvotes and downvotes from the former buyers. For factual type QA pairs, such community votes reflect users' stance towards the statement claimed in the answer, indicating the overall veracity judgement given by the entire community. It is not surprising that some answers may not have any vote in practice. But those answers with votes can provide precious labeled data for our investigation in PQA forums.

To ensure the quality of labels, we first filter out answers with total votes (including upvotes and downvotes) less than 2. Then following typical settings in fact checking datasets (Vlachos and Riedel, 2014; Wang, 2017; Augenstein et al., 2019), we consider the problem as a multi-class classification task and divide answers into five types according to their community votes as shown in Table 2. The rationality is that fully objective truth is often elusive and ill-defined as pointed out in Popat et al. (2018). For example, an answer may contain partially true information for the question. Thus, such veracity label partition can also be interpreted as measuring the answer credibility or reliability in multiple scales.

3.3 Evidence Retrieval

We then use the question text to retrieve relevant product information as evidence for providing external information when predicting the answer veracity. In E-commerce scenario, product descriptions from the manufacture and user reviews from the former buyers contain rich product information, which can be treated as the candidate information pool for the retrieval process. Similar with Thorne et al. (2018a), we rank the evidence sentences by TF-IDF similarity to the question text. To further improve the accuracy of the retrieved evidence, we only use the TF-IDF similarity as an initial filtering step, then the pre-trained BERT (Devlin et al., 2019) is utilized as the sentence encoder to encode the filtered evidence sentences and question text. The k nearest evidence sentences using cosine similarity with the encoded question are kept as the evidence for veracity verification. The statistics of the entire dataset is reported in Table 3.

4 Answer Veracity Prediction

Problem Definition. Given an answer $\boldsymbol{a}$ to its corresponding question $\boldsymbol{q}$, our aim is to predict the answer veracity which falls into one of the predefined veracity type, with the help of k relevant evidence sentences $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_k$.

In this section, we describe our proposed model AVER for the Answer Veracity prediction task with tailored Evidence Ranking module. An overview of AVER is shown in Figure 1.

4.1 Attention-based Input Encoding

For each word in the given text sequences, which is either a question, an answer or an evidence sentence, we use an embedding matrix to map it into a vector representation. To capture the temporal

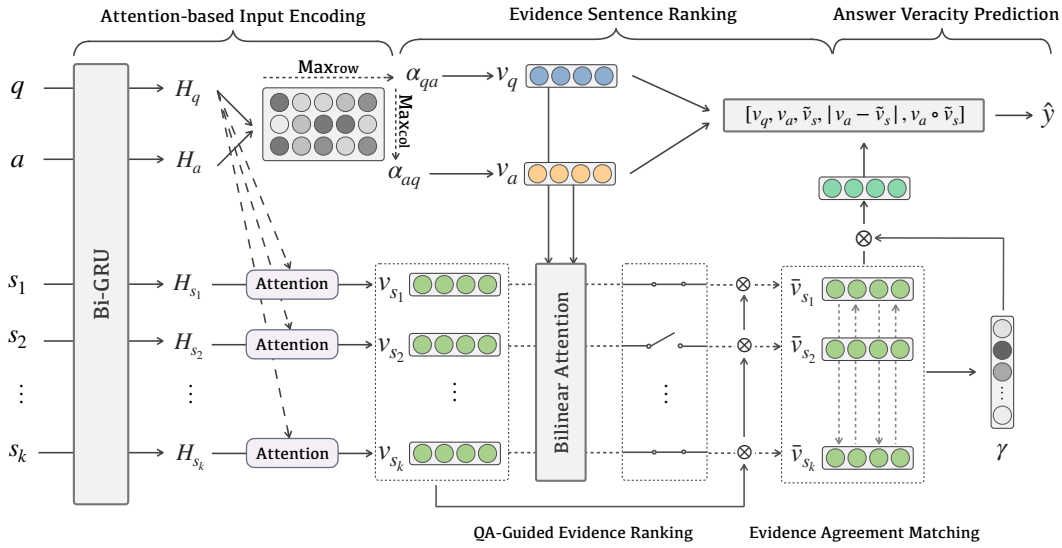

Figure 1: The architecture of our proposed AVER model

interactions between words, we employ a bidirectional GRU to transform the word embedding w_t to the context-aware representation h_t:

$$h_t^* = \text{Bi-GRU}(h_{t-1}^*, w_t), \quad * \in [q, a, s_i] \tag{1}$$

where $h_t^* \in \mathbb{R}^{d_h}$ is the hidden state at the t-th time step for the corresponding text sequence, d_h is the dimension of the hidden state. We denote the whole sequence as $H_* = [h_1^*, h_2^*, \ldots, h_l^*] \in \mathbb{R}^{l_* \times d_h}$ where l is the corresponding sequence length.

For predicting the answer veracity, one can note that rich semantic information is implicitly contained in the question text, indicating the importance of capturing the interrelations between the QA pair when encoding them. We thus employ a dual attention mechanism to encode the question and answer text with attention from each other:

$$S = H_q \cdot H_a^T \in \mathbb{R}^{l_q \times l_a} \tag{2}$$

where each item S_{ij} in the alignment matrix S denotes the alignment score between the i-th word in H_q and the j-th word in H_a. Next we can compute the dual attention weight for the question and answer respectively as follows:

$$\alpha_{qa} = \text{softmax}(\max_{\text{row}}(S)) \tag{3}$$

$$\alpha_{aq} = \text{softmax}(\max_{\text{row}}(S^T)) \tag{4}$$

$$v_q = H_q^T \cdot \alpha_{qa}, \quad v_a = H_a^T \cdot \alpha_{aq} \tag{5}$$

where $\max_{\text{row}}()$ denotes row-wise max-pooling operation. We can then obtain the encoded question embedding v_q and answer embedding v_a as the weighted sum of the context-aware representations of each word in the corresponding sequence.

Since the evidence sentences, either reviews or product descriptions are not written specifically for answering the question, we utilize the question text to highlight the important units in the evidence sentences during their encoding process. Therefore, we can obtain the encoded vector representation v_{s_i} for the i-th evidence sentence as follows:

$$T = H_{s_i} W_1 H_q^T \in \mathbb{R}^{l_{s_i} \times l_q} \tag{6}$$

$$v_{s_i} = H_{s_i}^T \cdot \text{softmax}(\max_{\text{row}}(T)) \tag{7}$$

where $W_1 \in \mathbb{R}^{d_h \times d_h}$ is a trainable weight matrix of the bilinear attention module to incorporate the different writing styles between q and s_i. We denote the encoded representations for all evidence sentences as $v_s = [v_{s_1}, v_{s_2}, \ldots, v_{s_k}] \in \mathbb{R}^{k \times d_h}$.

4.2 Evidence Sentence Ranking

One characteristic of our problem setting is that not all evidence sentences are equally useful and reliable. For example, some user reviews can be misleading and even conflicting with other evidences, requiring the model to take such imperfectness of the evidence sentences into consideration. To this end, we design an evidence sentence ranking module to capture the importance of each sentence.

4.2.1 QA-guided Evidence Ranking

We first use the question and answer to measure the usefulness of each evidence sentence:

$$\beta = f(v_s W_2 v_q + v_s W_3 v_a) \tag{8}$$

where $\beta \in \mathbb{R}^k$ denotes the weights for each evidence sentence, W_2 and W_3 are trainable parameters. The function $f()$ acts as a gate, which can be $sigmoid()$ or $softmax()$ function, corresponding to two different gating strategies:

Hard Gate. When the $sigmoid()$ function is applied element-wise for each sentence, the network will tend to assign weights closing to 0 or 1 to each evidence sentence. Thus such process will be similar as an evidence selection process, where only the useful evidence sentences will be "activated" to play the role in verifying the claim.

Soft Gate. The $softmax()$ function on the other hand will normalize the score for each evidence sentence. Thus, more important evidence can have larger weight and attach more importance in the subsequent prediction process.

After obtaining a score for each sentence, we then apply an element-wise product to obtain a new representation for each evidence sentence s_i:

$$\bar{v}_s = \beta \otimes v_s \in \mathbb{R}^{k \times d_h} \tag{9}$$

4.2.2 Evidence Agreement Matching

One remaining issue is that not all evidence sentences are always reliable. For example, s_5 in Table 1 contains opposite opinions with other evidence sentences and can mislead the veracity prediction process. To tackle this issue, we conduct an agreement matching process among the evidences to cross-check their internal coherence:

$$\gamma = \text{softmax}(w_4^T \tanh(W_5 \cdot \bar{v}_s^T)) \tag{10}$$

where $w_4 \in \mathbb{R}^{d_a}$ and $W_5 \in \mathbb{R}^{d_a \times d_h}$ are trainable parameters, $\gamma \in \mathbb{R}^k$ denotes the coherence weight for each evidence sentence. As discussed in Lin et al. (2017), such vector representation usually focuses on one specific aspect among the sentences. To capture multiple factual aspects involved in the verification process, we extend Equation 10 to a multi-view agreement matching as follows:

$$\Gamma = W_4 \cdot \tanh(W_5 \cdot \bar{v}_s^T) \tag{11}$$

$$\gamma' = \text{softmax}(\max_{\text{col}}(\Gamma)) \in \mathbb{R}^k \tag{12}$$

where $W_4 \in \mathbb{R}^{n_a \times d_a}$ and $W_5 \in \mathbb{R}^{d_a \times d_h}$ are trainable parameters, Γ is the multi-view agreement matching matrix. We then conduct a max-pooling on such matrix and take the softmax operation on the resulting vector to obtain the weight vector γ'. Then a combined evidence embedding denoting the most related evidence information from all evidence sentences can be calculated as follows:

$$\tilde{v}_s = \sum_{i=1}^{k} \bar{v}_{s_i} \cdot \gamma'_i \tag{13}$$

Note that the evidence embedding $\bar{v}_s$ is obtained by scaling with the importance weights of each evidence sentence if the soft gate is utilized. Thus we will substitute $\bar{v}_s$ by v_s in Equation 13 if the soft gate is utilized, which is also empirically better on our held-out validation set.

4.3 Answer Veracity Prediction

After obtaining the combined evidence embedding $\tilde{v}_s$, we utilize it to verify the answer claim. Following (Mou et al., 2016; Yang et al., 2019a) for strengthening the inference relations between the evidence and answer claim, we integrate the answer claim embedding v_a, evidence embedding $\tilde{v}_s$, their absolute difference $|v_a - \tilde{v}_s|$, and the element-wise product $v_a \otimes \tilde{v}_s$ into a prediction vector. Moreover, since the question text also implicitly contains useful semantic information, we also concatenate the question embedding v_q to the prediction vector. It is then fed to a MLP layer to make the prediction:

$$\hat{y} = \text{MLP}\left([v_q, v_a, \tilde{v}_s, |v_a - \tilde{v}_s|; v_a \otimes \tilde{v}_s]\right) \tag{14}$$

The entire model can then be trained end-to-end by computing the cross-entropy loss between the prediction $\hat{y}$ and the ground-truth label y.

5 Experiments

5.1 Experimental Setup

Dataset As introduced in Section 3, AnswerFact has 60,864 QA pairs in total [2]. We randomly split them into a training set and a test set with the ratio being 90:10. In addition, we set aside 10% data from the training set as the validation set to tune hyper-parameters during training.

Following previous work (Rashkin et al., 2017; Ma et al., 2019), we conduct experiments in two label settings, one considering all five classes introduced in Table 2, another merging the middle three classes, i.e., PARTTRUE, UNSURE and PARTFALSE as the class MIXED similarly with Ma et al. (2019). Such different label granularities can provide us a more practical and comprehensive understanding of our concerned task.

[2]The dataset can be found at https://isakzhang.github.io/.

Model	3-CLASS					5-CLASS	
	Mac-F1	Mic-F1	F_{TRUE}	F_{MIXED}	F_{FALSE}	Mac-F1	Mic-F1
CNN-claim	0.442	0.648	0.791	0.144	0.390	0.249	0.649
LSTM-claim	0.492	0.649	0.785	0.302	0.390	0.253	0.653
DeClarE	0.450	0.635	0.785	0.153	0.413	0.243	0.635
NSMN	0.504	0.663	0.799	0.284	0.429	0.279	0.651
MultiFC	0.513	0.655	0.787	0.300	0.453	0.299	0.655
AVER-w/o gate	0.516	0.661	0.798	0.296	0.453	0.305	0.657
AVER-hard gate	0.526	**0.674**	**0.804**	0.306	0.467	0.326	0.662
AVER-soft gate	**0.534**	0.673	0.802	**0.314**	**0.486**	**0.330**	**0.665**

Table 4: Performance of various methods for answer veracity predictions on AnswerFact dataset. F_{TRUE}, F_{MIXED} and F_{FALSE} denotes the F1 scores for TRUE, MIXED and FALSE class respectively.

Experimental Details We utilize the pre-trained 300D GloVe word vectors (Pennington et al., 2014) to initialize the embedding matrix and fine-tune it during training. k in Section 3 is set to 5. The hidden dimension of the Bi-GRU is set to be 256 with dropout of 0.4. For the evidence agreement matching module, we perform a grid search over n_a and d_a with the following hyperparameters where the final setting is underlined: $n_a = [2, \underline{3}, 4]$ and $d_a = [64, \underline{128}, 256]$. ReLU is used as the activation function in the MLP layer. We assemble batches of answers with similar length together with the batch size being 64. We use the Adam optimiser with learning rate of 0.0005 and train all models on two Tesla K80 GPUs. To avoid overfitting, we conduct early stopping on the validation set with a patience being 5 and add a L2 regularization with the weight of 0.002.

Evaluation Metrics We use macro and micro averaged F1 score, as well as class-specific F1 score as the evaluation metrics.

5.2 Baseline Models

We compare our proposed model with the following baseline and state-of-the-art models: 1) **CNN-claim** and 2) **LSTM-claim**: Two claim-focused fact checking models based on CNN (Rashkin et al., 2017) and LSTM (Rashkin et al., 2017) for obtaining claim representations respectively. Both of them exploit the claim text solely without considering any external evidence. 3) **DeClarE** (Popat et al., 2018): An evidence-aware neural fact checking model of textual claims. It utilizes a word-level attention for highlighting important units in evidence sentences. 4) **NSMN** (Nie et al., 2019): A pipeline-based system which ranked first in the FEVER shared task (Thorne et al., 2018b). We use its claim verification module for our task. 5) **MultiFC** (Augenstein et al., 2019): An evidence-based fact checking model which jointly rank evidence pages and conduct veracity predictions. Since the answer itself often does not contain enough information for the veracity prediction as discussed before, we concatenate the question and answer text as the "claim" for these fact checking models facilitating a more fair comparison.

For our proposed model, we report its performance with no gate mechanism involved ("**AVER-w/o gate**"), with hard gate ("**AVER-hard gate**") and soft gate ("**AVER-soft gate**") respectively as introduced in Section 4.2.1.

5.3 Veracity Prediction Results

Table 4 shows the results of different methods for predicting the answer veracity on the AnswerFact dataset with two label settings. It can be observed that models considering evidence information (e.g., MultiFC and AVER model) consistently achieve better results than those relying on claim text only (e.g., CNN-claim model). An exception is the DeClarE model which only obtains similar performance with the CNN-claim method. We conjecture that DeClarE treats each claim-evidence pair as one training instance without considering the relations between evidence sentences. Thus the model can be misled by conflicting evidence sentences and makes random predictions. This further indicates the necessity of selecting and ranking the evidence sentences by their importance for the prediction.

For our proposed model, we can find that AVER without any gate can already achieve better results

	3-CLASS	5-CLASS
QA (claim) only	0.507	0.253
+ avg evidence embed	0.514	0.313
+ fc evidence ranking	0.511	0.264
+ hard evidence ranking	0.526	0.326
+ soft evidence ranking	0.534	0.330

Table 5: Comparison of different evidence ranking strategies, Macro-F1 scores are reported.

	3-CLASS		5-CLASS	
	Mac	Mic	Mac	Mic
AVER-soft gate	**0.534**	**0.673**	**0.330**	**0.665**
- w/o QA-guided	0.516	0.661	0.305	0.657
- w/o agree-match	0.514	0.659	0.298	0.656
- w/o multi-view	0.522	0.669	0.315	0.656

Table 6: Ablation studies on AVER. Mac/Mic refer to Macro/Micro F1 scores respectively.

than most baseline models, showing the effectiveness of the agreement matching mechanism among evidence sentences for cross-checking their coherence. With the guide from question and answer information, the model with either soft or hard gate mechanism consistently outperforms all baseline methods on two label settings. This result suggests that the attention information from the QA pair is very important for ranking the evidence sentences and highlighting those more helpful sentences for assisting the prediction. Moreover, we can notice that the model with soft gate obtains better results than the model with hard gate in general, suggesting that measuring the importance of each evidence sentence with a soft weight is better than aggressively determining whether to "select" an evidence sentence or not in the hard gate mechanism for our concerned problem.

5.4 Analysis and Discussion

In this section, we conduct detailed analysis on our proposed evidence ranking module, which plays an important role for finding out more helpful and reliable evidence sentences for the subsequent veracity prediction.

Impact of Evidence Ranking Strategies

To investigate the effectiveness of our proposed evidence ranking strategy, we substitute it with two possible alternatives and present the results in Table 5. Specifically, we first report the results with QA pair only ("QA only") as a base model. Then we use the average sentence embedding $\frac{1}{k}\sum_{i=1}^{k} v_{s_i}$ to replace $\tilde{v}_s$ in Eq.13 to examine what if we do not consider the relations among the evidence sentences ("avg evidence embed"). We also create another model by utilizing a fully-connected layer to capture the relation of each evidence sentence with the answer and then concatenate these predictions to make the final judgement ("fc evidence ranking") as proposed in Augenstein et al. (2019).

We can see that our proposed model is superior than both alternatives since it carefully ranks the evidence sentences with both information from QA pair and agreement matching. It can be noticed that the model with fully connected evidence ranking performs even worse than averaging the evidence embeddings. This is likely due to the fact that it would be difficult for the model to implicitly learn the relations for each claim-evidence pair given only the veracity label. We alleviate this issue by conducting an agreement matching among the sentences first and then calculating a combined evidence embeddings to assist the prediction.

Ablation Study

We perform ablation studies by discarding some important components of AVER to investigate their effectiveness. For two evidence ranking modules, we discard QA-guided evidence ranking by directly replacing $\bar{v}_s$ in Eq.11 with v_s so as to neglect the QA information ("w/o QA-guided"). Then we create another variant by using the weight vector β in Eq.8 for calculating the combined evidence embedding in Eq.13 resulting in leaving out the evidence agreement matching component ("w/o agree-match"). As shown in Table 6, both modules contribute to the final veracity prediction performance in either label setting, indicating the importance of treating each evidence sentence differently for predicting the answer veracity. Moreover, we also replace the multi-view agreement matching with the single-view matching operation in Eq.10 ("w/o multi-view"), which leads to an inferior performance. This result indicates that cross-checking the coherence among the evidence from multiple perspectives can better measure the importance of each sentence, thus helping the final prediction.

Case Study

We present a sample case in Table 7 which is correctly predicted as false by AVER. The evidence

Question: Does this case fit the S4 with the inductive charging back? It is slightly thicker than the original back. ***Answer:*** No, it will not is only for the S2. ***Verdict:*** `FALSE`
s_1: Love these cases...they fit the Galaxy S4 so well, they even accommodate the wireless charger back plate. s_2: It fits the s4 perfect, the cut outs are perfect and its not bulky. s_3: I had a very similar case for my Galaxy S2, so I bought this one hoping it would hold up as well as the first. s_4: I wish it was available in more colors for the Galaxy S4. s_5: The case didn't work with extended battery and cover.

Table 7: A sample case of the prediction where the evidence sentences are ranked by their attention weights.

sentences are also shown, ranked by their weight γ_i' in Eq.12. We can observe that the top ranked evidences are highly topically relevant to the QA pair and coherent to other evidence sentences. Moreover, they contain essential information that can be directly used to infer the verdict of the answer. In contrast, the lower ranked sentences contain less relevant information which should play less important role during the prediction. This example indicates that different importance and usefulness of each evidence sentence need to be taken into consideration when predicting the answer verdict.

6 Conclusions

In this paper, we investigate the fact checking problem in product question answering forums, aiming to predict the answer veracity so as to provide more reliable online shopping environment. To this end, we introduce AnswerFact, an evidence-based fact checking datasets in QA settings. Further, we propose AVER model to predict answer veracity via tailored evidence ranking module. Extensive experiments show that our proposed method outperforms various established baselines.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 4684–4696.

Ramy Baly, Mitra Mohtarami, James R. Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 21–27.

David Carmel, Liane Lewin-Eytan, and Yoelle Maarek. 2018. Product question answering using customer generated content-research challenges. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1349–1350.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations*, pages 169–174.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations ICLR 2020*.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 7651–7658.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017*, pages 69–76.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017*.

Jing Ma, Wei Gao, Shafiq R. Joty, and Kam-Fai Wong. 2019. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 2561–2571.

Tsvetomila Mihaylova, Georgi Karadzhov, Pepa Atanasova, Ramy Baly, Mitra Mohtarami, and Preslav Nakov. 2019. Semeval-2019 task 8: Fact checking in community question answering forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, pages 860–869.

Tsvetomila Mihaylova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Mitra Mohtarami, Georgi Karadzhov, and James R. Glass. 2018. Fact checking in community forums. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 5309–5316.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017*, pages 27–48.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019*, pages 6859–6866.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 3391–3401.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2931–2937.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM TIST*, 10(3):21:1–21:42.

Aaron Smith and Monica Anderson. 2016. Online shopping and e-commerce.

Bakhtiyar Syed, Vijayasaradhi Indurthi, Manish Shrivastava, Manish Gupta, and Vasudeva Varma. 2019. Fermi at semeval-2019 task 8: An elementary but effective approach to question discernment in community QA forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, pages 1160–1164.

Yi Tay, Minh C. Phan, Anh Tuan Luu, and Siu Cheung Hui. 2017. Learning to rank question answer pairs with holographic dual LSTM architecture. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 3346–3359.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 809–819.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the Workshop on Language Technologies and Computational Social Science@ACL 2014*, pages 18–22.

Mengting Wan and Julian J. McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *IEEE 16th International Conference on Data Mining, ICDM 2016*, pages 489–498.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 422–426.

Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4787–4798.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019a. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709.

Xiao Yang, Madian Khabsa, Miaosen Wang, Wei Wang, Ahmed Hassan Awadallah, Daniel Kifer, and C. Lee Giles. 2019b. Adversarial training for community question answer selection based on multi-scale matching. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 395–402.

Wenxuan Zhang, Yang Deng, and Wai Lam. 2020a. Answer ranking for product-related questions via multiple semantic relations modeling. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 569–578.

Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. 2020b. Review-guided helpful answer identification in e-commerce. In *WWW '20: The Web Conference 2020*, pages 2620–2626.