Beyond Relevance: Trustworthy Answer Selection via Consensus Verification

Lixin Su, Ruqing Zhang, Jiafeng Guo, Yixing Fan, Jiangui Chen, Yanyan Lan, and Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{sulixin17b,zhangruqing,guojiafeng,fanyixing,chenjiangui18z,lanyanyan,cxq}@ict.ac.cn

ABSTRACT

Community Question Answering (CQA) sites such as Yahoo! Answers and Baidu Knows have emerged as rich knowledge resources for information seekers. However, answers posted to CQA sites often vary a lot in their qualities. User votes from the community may partially reflect the overall quality of the answer, but they are often missing. Hence, automatic selection of "good" answers becomes a practical research problem that will help us manage the quality of accumulated knowledge. Without loss of generality, a good answer should deliver not only relevant but also trustworthy information that can help resolve the information needs of the posted question, but the latter has received less investigation in the past. In this paper, we propose a novel matching-verification framework for automatic answer selection. The matching component assesses the relevance of a candidate answer to a given question as conventional QA methods. The major enhancement is the verification component, which aims to leverage the wisdom of crowds, e.g., some big information repository, for trustworthiness measurement. Given a question, we take the top retrieved results from the information repository as the supporting evidences to distill the consensus representation. A major challenge is that there is no guarantee that one can always obtain reliable consensus from the wisdom of crowds for a question due to the noisy nature and the limitation of the existing search technology.Therefore, we decompose the trustworthiness measurement into two parts, i.e., a verification score which measures the consistency between a candidate answer and the consensus representation, and a confidence score which measures the reliability of the consensus itself. Empirical studies on three real-world CQA data collections, i.e. YahooQA, QuoraQA and AmazonQA, show that our approach can significantly outperform the state-of-the-art methods on the answer selection task.

CCS CONCEPTS

• **Information systems → Question answering**.

KEYWORDS

Trustworthy, Answer Verification, Answer Ranking

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441781

ACM Reference Format:
Lixin Su, Ruqing Zhang, Jiafeng Guo, Yixing Fan, Jiangui Chen, Yanyan Lan, and Xueqi Cheng. 2021. Beyond Relevance: Trustworthy Answer Selection via Consensus Verification. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441781

1 INTRODUCTION

Community question answering (CQA) sites, e.g., Quara[1], Amazon product CQA[2] and Yahoo! Answers[3], can utilize the power of the community to provide timely and personalized service to information seekers, and thus have merged as very rich knowledge resources for both general and specific/vertical topics. Unfortunately, answers posted to CQA sites vary a lot in their qualities since they are mostly written by ordinary users instead of professionals. Hence, they inevitably suffer from issues such as incomplete, redundancy, and even malicious content [21].

To identify high-quality answers, many CQA platforms allow the community to vote whether they like the answer or not. Such votes can indicate the overall quality of the answer. However, in practice, many answers do not get any votes. For instance, there are about 40% of answers without any vote at all in the Quora and Stack Overflow sites [34]. Hence, automatic selection of "good" answers for a posted question becomes a practical research problem that will help us manage the quality of accumulated knowledge.

A basic question here is the definition and modeling of the goodness of an answer. There have been many existing works on answer selection that formulated the task as an information retrieval (IR) problem, which ranked the answers with respect to their topical relevance to the posted question [32, 37]. Under this setting, both traditional machine learning techniques [13] and modern deep learning models [10] have been adopted, which leverage a variety of syntactic or semantic matching patterns [8] between a question-answer pair for topical relevance estimation. However, a good answer should deliver not only relevant but also trustworthy information that can help resolve the information needs of the posted question. For example, both answers in Figure 1 are topically relevant to the question, but not all of them are trustworthy as shown in the votes they got. A good answer is inherently relevant but not vice versa. To identify good answers in CQA, we need to extend the quality measurement of an answer from topical relevance to trustworthiness.

[1] https://www.quora.com
[2] https://nijianmo.github.io/amazon/index.html
[3] https://answers.yahoo.com/

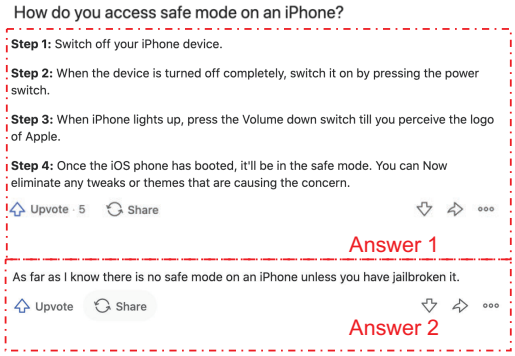

Figure 1: An example of trustworthy answer and relevant answer to a question.

While there have been some prior works on predicting content trustworthiness such as [9, 11, 19], the answer trustworthiness prediction in CQA scenarios has not been well studied before. There are other works estimating the overall answer quality in CQA based on user votes, user profiles, or social network information [14, 45, 46]. However, in this work, we tackle the answer quality estimation problem where user information is not available [34, 43].

So the follow-up question is how we can actually assess the trustworthiness of an answer to a posted question. According to the consensus theory of truth [6, 7, 12], trustworthiness means that the statement is generally agreed upon by the crowds. For the example in Table 1, user vote is a type of agreement. When votes are missing, people often resort to the wisdom of crowds either offline (e.g., asking a set of experts) or online (e.g., searching the available information) to find the consensus for trustworthiness verification. This inspires us to propose a similar verification process for answer selection. Specifically, given a question, we search some big information repository, such as Web or user-generated content (UGC), to find supporting evidences. We then distill the consensus from those supporting evidences for trustworthiness verification. However, due to the noisy nature of the information and the limitation of the existing search technology, there is no guarantee that one can always obtain reliable consensus from the wisdom of crowds. Therefore, the critical challenge here is how to distill reliable consensus from the noisy evidences for trustworthy computation, which is the major contribution of this work.

To achieve the above goal, we propose a novel Matching-Verification (MV) framework for answer selection. The framework includes two major components, a *matching component* and a *verification component*. The matching component is a basic unit, which assesses the relevance of a candidate answer to a given question as conventional QA methods. In our work, we adopt some advanced neural matching model to implement this component. The major enhancement is the verification component, which aims to distill the consensus from the retrieved supporting evidences for trustworthiness measurement. Specifically, we decompose the trustworthiness measurement into two parts, i.e., a verification score and a confidence score, to take the uncertainty of the available information into account. Specifically, we first build an interaction matrix over the supporting evidences which captures the similarity between them. Based on this interaction matrix, 1) We employ a pivoted attention mechanism to obtain the consensus representation, and compare it with the candidate answer to produce the verification score which measures the answer-consensus consistency; 2) We employ a self-interaction network to produce a confidence score over the interaction matrix, which measures the consistency among the supporting evidences themselves. The final trustworthiness measurement is the product of two scores.

We test our approach on three answer selection datasets which are collected from real-world CQA applications, i.e., Yahoo! Answers, QuoraQA, and AmazonQA. We take several state-of-the-art answer selection models as our baselines, including those based on question-answer matching models and those incorporating external knowledge resources, e.g., knowledge graphs or pre-training models. Experimental results demonstrate that our approach can significantly outperform the state-of-the-art methods on the answer selection task. Besides, we provide a detailed analysis of the proposed model to gain a better understanding of the trustworthiness measurement.

2 RELATED WORK

Answer selection is a long-term research challenge in QA field, which has attracted substantial attention from both academic and industrial communities. To facilitate the study and evaluation of the answer selection task, several answer ranking datasets have been proposed, e.g., WikiQA [39] dataset for the WebQA scenario, and Yahoo! Answer, QuoraQA and AmazonQA for the CQA scenario. In this paper, we consider related works that rank answers based on textual evidences. Without loss of generality, existing methods based on textual clues could be categorized into two categories according to whether the external knowledge is leveraged or not. We will briefly review these studies as follows.

2.1 Models without External Knowledge

Models without external knowledge, by its name, only take the question and answer as input and then predict a score indicating the goodness of the answer. Most methods in this category are motivated by the previous studies in conventional information retrieval, which model the relevance between a query-document pair.

Early models mainly use shallow methods such as human-crafted rules and patterns to extract various features from question-answer pairs, and then apply a learnable function to predict the answer goodness. Severyn et al. [22] introduced a model which encodes syntactic and shallow semantic properties of question/answer pairs and conducts classification using structural kernel method. Wang et al. [35] proposed a statistical syntax-based model that softly aligns a question sentence with a candidate answer sentence.

In recent years, with the development of deep neural networks, many neural matching models have been proposed for better capturing semantic information by distributed representations. Those models usually employ CNN-based, RNN-based, or attention-based structures to combine and transform the representations of the question and the answer. Shen et al. [25] proposed a CNN-based model which encodes the question and the answer using CNN layer respectively and calculates a matching score by dot product between these two representations. Wang and Nyberg [33] applied

a stacked bidirectional LSTM to sequentially read words from the question and the answer to generate representations for them, and then predicted the goodness by their encoded vectors. Tan et al. [27] developed hybrid models that process the question and the answer using both CNN and RNN to combine merits of both structures. There are more related models combine LSTM/CNN and attention mechanism to perform multi-turn interaction [38].

Compared with traditional heuristic methods, deep models can better circumvent the lexical gap by employing the distributed semantic representation and the non-linear transformation. In spite of the effectiveness of neural matching models, they only consider the relevance matching between the question, so they cannot achieve modeling trustworthiness.

2.2 Models with External Knowledge

External knowledge can provide rich information for answer trustworthiness and there have been many models proposed to incorporate various external knowledge. Previous work has shown that redundancy of a large collection can be used for answer validation. Knowledge bases can also be used to enhance the representation of the question and the answer.

Early works focused on feature engineering, external information can be utilized when constructing features. For example, Jeongwoo et al. [13] used logistic regression to estimate the goodness of an answer based on answer relevance features which are extracted from WordNet and Wikipedia to tackle the lexical gap. Clarke et al. [4] considered the co-occurrence information of the question-answer pair from retrieved results as a feature for judging the goodness. Surdeanu et al. [26] proposed to incorporate web correlation features which are the number of pages from a search engine, using the question and the answer as the search query. Riezler et al. [20] leveraged the statistical machine translation technique to learn a translation model from question-answer pairs and then used the transition probability of words as features.

However, these traditional methods are restricted to short answers in WebQA scenarios. When the answers get longer in CQA scenarios, answers contain much noise information rather than short fact entities and these methods degrade significantly based on our preliminary studies. Many neural models have been proposed to incorporate the external knowledge, including structured knowledge graph (e.g., WordNet, ConceptNet or Freebase), pre-trained model parameters on unstructured text, and user information.

- For the structured knowledge graph, Yih et al. [41] used lexical semantic resource, i.e., WordNet, to extract word-pair relation to enhance semantic features. Wu et al. [36] incorporated question topic words as prior knowledge and combined original word embeddings. In this way, prior knowledge can guide their model to focus on the important parts of long answers. Shen et al. [23] proposed a knowledge-aware attention mechanism to effectively incorporate external knowledge from the knowledge graph into sentence representational learning. In summary, the information from the structured knowledge graph usually helps to better understand individual concepts or connections between concepts.
- Pre-trained language models, trained on a large corpus, such as BERT [5] and XLNet [40], have also been proposed to facilitate NLP tasks. Pre-trained language models can dig the linguistic knowledge in the text and encode knowledge into model parameters via various pre-training objectives.BERT [5] is pre-trained with the mask language model and next sentence prediction based on Transformer structure. XLNet [40] incorporated autoregressive pre-training objective to eliminate a pretrain-finetune discrepancy.
- Note a couple of models have been proposed to leverage user information in CQA sites [15], which is complementary to external textual information, so we skip this series of works.

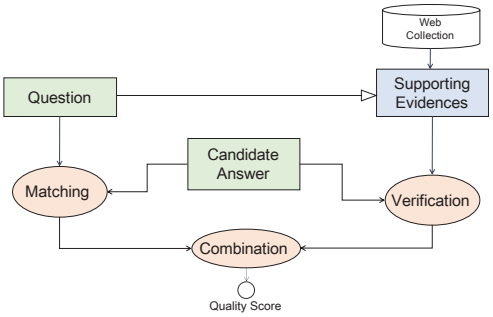

Figure 2: An Overview of the Matching-Verification (MV) Framework.

Knowledge bases contain accurate knowledge, however, they are sparse and incomplete compared to unstructured information. Pre-training methods on unstructured corpus only improve semantic representation. Different from previous work, we locate question-related evidences from the external big information repository from distill the consensus for answer verification. Similar to our work, Zhang et al. [44] used retrieved product reviews to verify answer helpfulness. However, the connection between reviews and quality of reviews is not considered.

3 OUR APPROACH

In this section, we introduce our proposed Matching-Verification (MV) framework for answer selection. We first introduce our key idea on the design of our framework. We then describe each component of the MV framework in detail as well as the learning procedure.

3.1 Key Idea

We propose the MV framework to assess the goodness of an answer to a given question for answer selection. In this work, we consider a good answer as both relevant and trustworthy. The MV framework is thus designed based on this key idea which is depicted in Figure 2.

- **Relevance Matching**: We consider relevance as a foundation of goodness, which could be assessed by the semantic matching between a question-answer pair. Therefore, in the MV framework, we propose a *matching component* which employs some state-of-the-art neural matching model to measure the relevance between the question and the answer.
- **Consensus Verification**: A good answer should also deliver trustworthy information that can help resolve the information needs of the given question. To take into consideration trustworthiness beyond relevance, we leverage the wisdom of crowds as a way to achieve this goal based on the consensus theory of truth

[7]. In the MV framework, we employ a *verification component* to take the top retrieved results from an external information repository as supporting evidences to distill the consensus representation for trustworthiness measurement. Consequently, the verification component assesses the answer-consensus consistency via the attention mechanism. However, the search results are usually uncertain, due to the noisy nature of online information and the limitation of existing search technology.

- **Confidence-based Combination**: The consensus verification may not always be available in practice, e.g., there might be no consensus existing in the information repository or the search system cannot rank those supporting evidences as the top results. To handle these situations, we measure the consistency among the supporting evidences via a self-interaction network to assess the confidence of the consensus. The relevance matching component and the consensus verification component are combined based on the above confidence to measure the goodness of a candidate answer. The underlying idea is that if the consensus is of low confidence, we will degrade our framework to relevance matching for basic performance guarantee.

3.2 Matching Component

The goal of the matching component is to assess the relevance of an answer to a given question. It takes the question and the answer as inputs, and estimates a matching score of how relevant a candidate answer is to a posted question. As discussed in Section 2, the relevance matching can be modeled in different models. Here, we adopt the advanced pre-trained model BERT, which has shown impressive performance on many NLP tasks [31].

Firstly, the question q and the answer a are concatenated as the input to BERT with special tokens delimiting them, i.e.,[CLS] and [SEP]. Specifically, each word in the concatenated sequence is represented by summing its distributed, segmentation, and positional embeddings. Then, the representation $\mathbf{H}^{[CLS]}$ of the special token [CLS] and the representation $\mathbf{H}^{qa}$ of the concatenated question and answer are obtained by,

$$\mathbf{H}^{[CLS]}, \mathbf{H}^{qa} = BERT_{QA}([CLS]; q; [SEP]; a; [SEP]).$$

Finally, to obtain the relevance score s_r of the answer to a posted question, we apply a sigmoid function over the representation $\mathbf{H}^{[CLS]}$ of [CLS] following previous studies [5], i.e.,

$$s_r = sigmoid(\mathbf{W}_r \mathbf{H}^{[CLS]}),$$

where $\mathbf{W}_r$ is a learnable parameter. Note we start training from a pre-trained BERT model and then fine-tune it on CQA data collections.

3.3 Verification Component

The goal of the verification component is to harness reliable consensus from an external information repository for trustworthiness measurement. Basically, the verification component contains the following two steps: 1) Consensus Representation Learning: to distill the consensus representation from the supporting evidences; 2) Answer-consensus verification: to produce the verification score of a candidate answer by measuring the consistency between the answer and the consensus; The overall architecture of the verification component is depicted in Figure 3 and we will detail our model as follows.

3.3.1 Consensus Representation Learning. The consensus representation learning aims to distill the consensus representation from the retrieved supporting evidences. The key idea for the distillation adopts a way like an expectation-maximization process, where we first vote up a pivoted supporting evidence (the E-step), and then we obtain the consensus representation collectively using a new pivoted attention mechanism (the M-step). The detailed steps are as follows.

- **Supporting Evidences.** Formally, given a question q, we take the top K retrieved results $\{f_1, \ldots, f_K\}$ from the information repository as the supporting evidences. Here, we consider two scenarios including open-domain CQA and vertical domain CQA, and leverage Web and user-generated content as the information repository respectively.
- **Representations of Supporting Evidences.** We employ the pre-trained model BERT to encode the supporting evidences. Given a supporting evidence f_i,where $i \in [1, K]$, and a candidate answer a, the answer representation $\mathbf{A}$ and the supporting evidence representation $\mathbf{F}^i$ are defined as:

$$\mathbf{A} = BERT_A(a) \in \mathcal{R}^{l_a * h},$$

$$\mathbf{F}^i = BERT_F(f_i) \in \mathcal{R}^{l_i * h},$$

  where l_i is the length of the supporting evidence f_i, l_a is the length of the answer a, and h is the size of the hidden representation in BERT. $BERT_F$ is the shared BERT encoder for evidences. Note that we pad evidences for the question to the same length for simplified computation.
- **Pivot Supporting Evidence.** Based on the representations of the supporting evidences, we extract the pivot supporting evidence with the highest importance under the assumption that a supporting evidence is important if it is highly related to many important supporting evidences.
  Based on the representations of the supporting evidences, we firstly build an interaction matrix $\mathbf{E} \in \mathcal{R}^{K * K * l_i * l_j}$, where l_i and l_j denotes the length of the supporting evidence f_i and f_j respectively. Concretely, the element $\mathbf{E}_{i,j}$ in $\mathbf{E}$ is defined as follows:

$$\mathbf{E}_{ij} = \mathbf{F}^i \mathbf{F}^j,$$

  where $\mathbf{F}^i$ and $\mathbf{F}^j$ denotes the representation of the supporting evidence f_i and f_j respectively.
  Then, we add a softmax function to normalize all the elements in the interaction matrix $\mathbf{E}$ along l_j dimension,

$$\mathbf{E}_{i,j,m,n} = \frac{\exp(\mathbf{E}_{i,j,m,n})}{\sum'_n \exp \mathbf{E}_{i,j,m,n'}}.$$

  Afterwards, we directly average the 2-th, 3-th and 4-th dimension of $\mathbf{E}_{i,j,m,n}$ to obtain the importance score $\mathbf{p}_i$ of each supporting evidence f_i,

$$\mathbf{p}_i = Avg(\mathbf{E}_{i,*,*,*}).$$

  Finally, we obtain the pivot supporting evidence $\mathbf{F}^p$ as:

$$\mathbf{F}^p = \arg\max_i p_i.$$

- **Consensus Representation.** We distill the consensus representation based on the representations of the supporting evidences

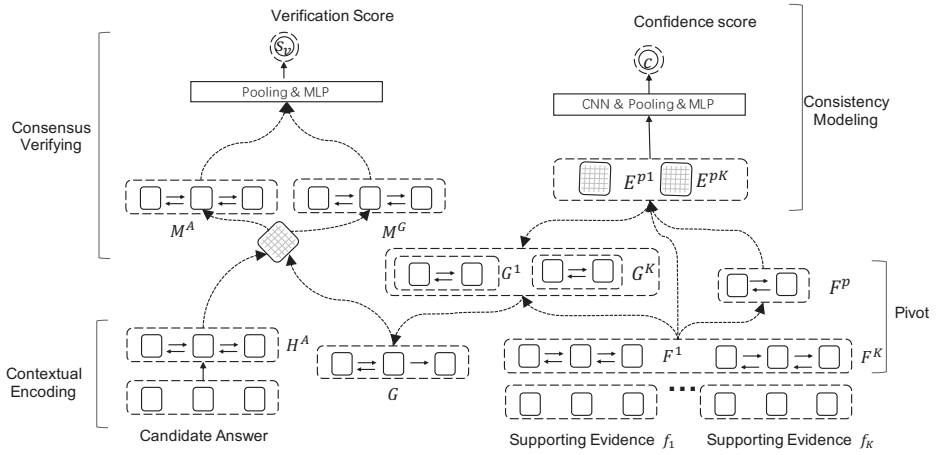

Figure 3: The model architecture of the verification component.

and the pivot evidence. To achieve this purpose, we introduce a *pivoted attention mechanism* to attend the pivot evidence to all supporting evidences $\{\mathbf{F}^j\}_{j=1}^K$. The key idea is that a supporting evidence is important if it is highly related to the pivot evidence. Specifically, the aligned evidence representations $\mathbf{G}^j$ of each supporting evidence f_j is obtained by

$$\mathbf{G}^j = \alpha_j \cdot \mathbf{F}^j,$$

where α_j indicates the similarity between the pivot evidence and each supporting evidence f_j, and is defined as,

$$\alpha_j = softmax_{col}(\mathbf{E}^{pj}),$$

$$\mathbf{E}^{pj} = \mathbf{F}^p \mathbf{F}^{jT},$$

where $softmax_{col}(\cdot)$ denotes the column-wise softmax normalization.

Finally, by concatenating K aligned evidence representations and feeding it to a shared fully-connected layer, we obtain the consensus representation $\mathbf{G}$ as follows:

$$\mathbf{G} = FC([\mathbf{G}^1; \mathbf{G}^2; \ldots; \mathbf{G}^K]) \in \mathcal{R}^{l_p * h},$$

where $FC(x) = tanh(x\mathbf{v}_i + b_i)\mathbf{v}_h + b_h$ denotes a feed-forward layer.

3.3.2 Answer-consensus Verification. Based on the answer representation and the consensus representation, the answer-consensus verification aims to produce the verification score s_v for each candidate answer to measure the consistency between the answer and the consensus.

Firstly, we compute the alignment matrix $\mathbf{E}^{AG}$ between the answer and the consensus to capture the semantic interaction information, i.e., $\mathbf{E}^{AG} = \mathbf{A}\mathbf{G}^T$. Then, we get the aligned answer representation $\hat{\mathbf{A}}$ and the aligned consensus representation $\hat{\mathbf{G}}$,

$$\hat{\mathbf{A}} = softmax_{col}(\mathbf{E}^{AG})\mathbf{G}, \hat{\mathbf{G}} = softmax_{row}(\mathbf{E}^{AG})\mathbf{A},$$

where $\hat{\mathbf{A}} \in \mathcal{R}^{l_a * d}$ and $\hat{\mathbf{G}} \in \mathcal{R}^{l_p * d}$.

Afterwards, we employ a fusion layer [38] over the answer representation $\mathbf{A}$ and the consensus representation $\mathbf{G}$ to obtain the fused answer representation $\mathbf{M}^A$ and the fused consensus representation $\mathbf{M}^G$, i.e.,

$$\mathbf{M}^A = Fusion(\mathbf{A}, \hat{\mathbf{A}}) \in \mathcal{R}^{l_a * d}, \mathbf{M}^G = Fusion(\mathbf{G}, \hat{\mathbf{G}}) \in \mathcal{R}^{l_p * d}.$$

Specificaly, the fusion layer, $Fusion(\mathbf{M}_1, \mathbf{M}_2)$, compares two representations in three perspectives and then fuse them together.

$$\mathbf{M}_{h1} = FC_1([\mathbf{M}_1; \mathbf{M}_2]),$$

$$\mathbf{M}_{h2} = FC_2([\mathbf{M}_1; \mathbf{M}_1 - \mathbf{M}_2]),$$

$$\mathbf{M}_{h3} = FC_3([\mathbf{M}_1; \mathbf{M}_1 \odot \mathbf{M}_2]),$$

$$Fusion(\mathbf{M}_1, \mathbf{M}_2) = FC_4([\mathbf{M}_{h1}; \mathbf{M}_{h2}; \mathbf{M}_{h3}]),$$

where FC_1, FC_2, FC_3 and FC_4 are single-layer feed forward networks with independent parameters.

The fused representations $\mathbf{M}^A$ and $\mathbf{M}^G$ are then transformed to fix-length vectors through a max pooling layer along the length dimension,

$$\mathbf{a} = Pooling(\mathbf{M}^A) \in \mathcal{R}^h,$$

$$\mathbf{g} = Pooling(\mathbf{M}^G) \in \mathcal{R}^h.$$

Finally, based on the vectors from the pooling layers, we obtain the verification score s_v through a feed-forward neural network and a sigmoid activation function, i.e.,

$$s_v = sigmoid(FC([\mathbf{a}; \mathbf{g}; \mathbf{a} - \mathbf{g}; \mathbf{a} \odot \mathbf{g}])).$$

3.4 Confidence-based Combination of Matching and Verification

To measure the consistency among the supporting evidences themselves, we employ a *self-interaction network* to produce a confidence score c.

Self-Interaction Network. In order to consider both interactions and semantic representations, we apply the convolution layer and pooling layer [17] over the interaction matrix $\{\mathbf{E}^{p1}, \ldots, \mathbf{E}^{pK}\}$ between the pivot evidence $\mathbf{F}^p$ and each supporting evidence $\mathbf{F}^k, k \in [1, K]$, to obtain the confidence score c, i.e.,

$$c = Conv\&Pooling\&MLP([\mathbf{E^{p1}}; \ldots; \mathbf{E^{pK}}]).$$

The consensus verification may not always be available in practice, and thus it is necessary to combine the relevance matching component and the consensus verification component to measure the goodness of a candidate answer. The final goodness score s is

obtained by using the sums of two scores,

$$s = s_r + cs_v. \quad (1)$$

If the consensus verification is of low confidence (i.e., the confidence score c is low), we will degrade our framework to relevance matching for performance guarantee.

3.5 Learning and Prediction

In the training phase, for each question q, we randomly select a good answer a^+ and a negative answer a^- to build a training sample (q, a^+, a^-). We train our framework by minimizing the following pairwise ranking loss $\mathcal{L}(\theta)$, i.e.,

$$\mathcal{L}(\theta) = \max(0, m - s^+ + s^-),$$

where $s^+ = MV(q, a^+, f), s^- = MV(q, a^-, f)$, f denotes the supporting evidences. m is a pre-defined margin to judge if a training triple will be terminated or not, emphasizing the selection of good answers. In this way, we demand the good answer to be assigned a higher score than the negative answer.

In the prediction phase, we apply the learned MV framework to predict the goodness score s for each candidate answer a of a given question q, i.e., $s_a = MV(q, a, f)$. All the candidate answers are ranked according to their probability of goodness to the question.

4 EXPERIMENTS

In this section, we conduct experiments to verify the effectiveness of our proposed model.

4.1 Dataset Description

To evaluate the performance of our model, we conduct experiments using three large answer selection datasets, i.e., QuoraQA [16], AmazonQA [42], and YahooQA [30].

- **AmazonQA.** Following Zhang et al. [42], we collect questions and answers from Amazon product pages. Then, we collect reviews with respect to the product following [29]. Answers which have at least three up-votes are regarded as the good answers. Reviews are used as the external information repository for evidence retrieval.
- **YahooQA.** We collect the question-answer pairs from Yahoo! Answers following [30]. Each question is associated with multiple candidate answers and the best answer is selected by the human annotators.
- **QuoraQA.** Following lyu et al. [16], we leverage the Quora questions and answers and select the highest-upvoted answer as the best answer. We utilize web pages as the external information repository for YahooQA and QuoraQA.

In these three datasets, each question is associated with multiple candidate answers, which are labeled with binary labels indicating the answer is good or not. Table 1 shows the detailed statistics of the datasets.

4.2 Evaluation Methodologies

As for evaluation metrics, MRR and $P@1$ are used in our experiments, which are commonly used metrics in answer selection. Note although we consider trustworthiness beyond relevance, previous metrics are still applicable. The reason is that the golden label for answer selection indicates the goodness of an answer indeed and modeling goodness via relevance and trustworthiness is just a better solution for answer selection.

4.3 Baselines

We compare our MV framework with several state-of-the-art answer selection models, including models without external knowledge and models with external knowledge.

4.3.1 Models without external knowledge. We firstly employ traditional semantic matching models without external knowledge, including representation-based and interaction-based models.

- **Representation-based models** firstly encode the question and the answer separately, and then compute the similarity between the two representations. We compare our model with two representative models in this category, i.e., QA-CNN [27] and QA-LSTM [27], which employs a CNN-based layer and a bi-directional LSTM to obtain the representations respectively.
- **Interaction-based models** [10] first build local interactions (i.e., local matching signals) between two pieces of text using various attention mechanisms, and then leverage deep neural networks to learn hierarchical interaction patterns for matching. We choose two well-performed interaction-based models, i.e., RE2 [38] and ESIM [2]. Specifically, RE2 [38] highlights three key features, namely previously aligned features, original point-wise features, and contextual features for inter-sequence alignment. ESIM [2] is another powerful model that uses Bi-LSTM to encode texts and apply the attention and fusion layer over the representations to obtain the label.

4.3.2 Models with External knowledge. We also consider several answer ranking models that leverage external knowledge, e.g., knowledge graph and pre-trained model.

- **Knowledge Graph** (KG)-based models leverage information derived from knowledge graph [1] to enhance the answer ranking. We choose recently proposed KABLSTM [24], which leverages external knowledge from KG to enrich the representational learning of QA sentences, as our baseline.
- **Pre-training** models learn knowledge [3, 5] from large corpus, and then the learned parameters are fine-tuned for various downstream tasks. We compare our model with the well-known pre-trained model BERT [5]. The model structure of BERT is based on Transformer [28].
- **Review-guided** model RAHP [42] use retrieved product reviews to verify the answer helpfulness. RAHP individually compares each review with the answer and aggregate the score for answer helpfulness.

4.4 Implementation Details

For the QA-CNN, QA-LSTM, and ESIM baselines, we leverage the implementations in open source toolkit MatchZoo [4]. For the RE2 baseline, we use the code released by original authors [5]. For the BERT baseline, We use the pre-trained parameters of BERT-based-uncased version [6]. For the KABLSTM baseline, we use the

[4] https://github.com/NTMC-Community/MatchZoo
[5] https://github.com/alibaba-edu/simple-effective-text-matching
[6] https://github.com/huggingface/transformers

Table 1: Data Statistics. QL: the length of a question, AL: the length of an answer, PosRate: the rate of positive labels.

Dataset	#Questions (train/dev/test)	#Answers (train/dev/test)	#Avg QL	#Avg AL	#Avg EL	%PosRate
QuoraQA	34,125/3941/3893	136,948/15,567/15,569	14.4	203.6	49.1	30.3
YahooQA	10,657/833/865	88,000/6483/7097	10.2	46.6	52.5	16.1
AmazonQA	17,314/2278/2236	36,083/4525/4516	31.5	81.4	71.6	67.5

Table 2: Comparisons between our MV framework and baselines on three datasets.

	AmazonQA		QuoraQA		YahooQA	
Model	P@1	MRR	P@1	MRR	P@1	MRR
QA-CNN	0.6937	0.7398	0.4151	0.6409	0.335	0.5402
QA-LSTM	0.7073	0.7502	0.4246	0.6480	0.3387	0.5422
ESIM	0.7276	0.7632	0.4466	0.6675	0.3665	0.5635
RE2	0.7316	0.7652	0.4724	0.6857	0.3861	0.5867
KABLSTM	0.7202	0.7594	0.4754	0.6812	0.3742	0.5767
RAHP	0.7406	0.7683	0.479	0.692	0.3898	0.587
BERT	0.7421	0.773	0.4812	0.6981	0.4404	0.6153
MV	**0.7548**	**0.7929**	**0.496**	**0.7378**	**0.4629**	**0.6434**

original implementation [7] and use same hyper-parameters. For RAHP, we use the official implementation from the author [8]. All the hyper-parameters, such as learning rate and sequence truncated length, are tuned on the validation set. The batch size for all models except BERT is 64, and batch size for BERT is set to 16. We pad the sequence to batch-wise maximum length to save computation. We use the Glove embedding. We apply Adam to learn the model parameters.

For the collection of the supporting evidences, we leverage the public Bing search engine as the retriever over the web, and the top ten ranked Web snippets are used as the supporting evidences for QuoraQA and YahooQA. For AmazonQA, related product reviews are used as the information repository indexed by Elasticsearch[9] and top ten reviews ranked by BM25 similarity are used as the supporting evidences. The search query is based on the question with stop words and punctuations removed. To avoid data leakage, we remove the source pages from Yahoo! Answers or Quora website.

We implement our MV framework in PyTorch [18]. We fine-tune model from the bert-base-uncased checkpoint as the initial checkpoint. In the training phase, the batch size is 16, and the epoch is 6. We apply early stopping based on the validation set performance. The learning rate of Adam algorithm is set as 2e-5. We apply warm-up strategy and set the warm-up rate as 0.1. Dropout with probability 0.1 is applied to all feed-forward layers. All the hyper-parameters are tuned on the validation set. We select the model that achieves the best performance on the development set and report results on the test set.

4.5 Baseline Comparison

We compare the performance between our MV framework and the baselines on the CQA datasets, i.e., YahooQA, QuoraQA, and AmazonQA. As shown in Table 2, we can find that: (1) *QA-LSTM* performs better than *QA-CNN* on all three datasets, showing that LSTM is more responsible than CNN for question answer matching by capturing the long-term dependence information. (2) Interaction-based models (i.e., *ESIM* and *RE2*) perform better than representation-based models (i.e., *QA-LSTM* and *QA-CNN*). This is mainly because the interaction-based model is capable to capture more complex semantic interaction between sentences by learning from a matching matrix/histogram between a question and an answer. For example, the relative improvement of *RE2* over the *ESIM* is 2.6% in terms of MAP on the QuoraQA dataset. The reason might be that by stacking multiple interaction blocks *RE2* can better capture the interaction than *ESIM*, which only runs once interaction. (3) *KABLSTM* performs better than the interaction-based models (i.e., *ESIM* and *RE2*) on QuoraQA and slightly worse on AmazonQA. This indicates that the knowledge base is useful for answer ranking, while it is sparse for product related questions in AmazonQA. (4) RAHP performs better than ESIM and RE2, which demonstrates that retrieved evidences are informative for answer goodness assessment. (5) Pre-trained BERT is the best performing baseline model, which consists of multiple interactions and utilizes knowledge from an extremely large corpus. (6) By learning hierarchical interaction patterns for matching, and introducing the consensus from supporting evidences retrieved from the big information repository, our MV framework can achieve the best performance on the three CQA datasets with long and noisy non-factoid questions.

Table 3: Ablation analysis on AmazonQA and QuoraQA.

Model Ablation	AmazonQA		QuoraQA	
	P@1	MRR	P@1	MRR
MV	**0.7548**	**0.7929**	**0.496**	**0.7378**
$MV_{-confidence}$	0.7504	0.7851	0.4849	0.7275
MV_{-pivot}	0.741	0.7784	0.4873	0.7292
$MV_{-matching}$	0.7377	0.7715	0.4795	0.7184
$MV_{-verification}$	0.7421	0.773	0.4812	0.6981

4.6 Ablation Analysis

We conduct ablation analysis to investigate the effectiveness of key designs in our framework, i.e., pivot attention mechanism in Section3.3.1, self-interaction network in Section 3.4 of evidences, and the whole verification component in Section 3.3. Firstly, we replace the pivot attention mechanism with a heuristic mechanism, which selects the top one ranked supporting evidence as the pivot and use it to aggregate aligned representation from other evidences. We name this method as MV_{-pivot}. Then, we remove the self-interaction network for obtaining the confidence score and directly sum the matching score and verification score. We denote this method as $MV_{-confidence}$. Finally, we attempt to remove the

[7] https://github.com/dengyang17/kablstm
[8] https://github.com/isakzhang/answer-helpfulness-prediction
[9] https://www.elastic.co/cn/

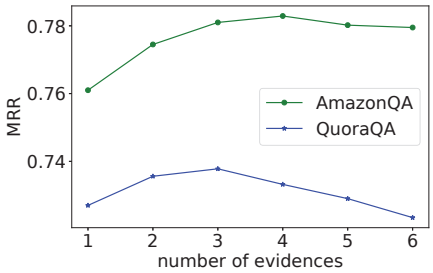

Figure 4: MMR results with respect to different numbers of supporting evidences.

matching or verification component, denoted as $MV_{-matching}$ and $MV_{-verification}$ respectively. Due to space limitation, we only show the experimental results on two datasets, i.e., AmazonQA and QuoraQA, and we get similar results on the remaining one.

As shown in Table 3, we can see that: (1) The performance of both MV_{-pivot} and $MV_{-confidence}$ model have a drop as compared with *MV*. The results verify our assumption in Section 3.1, i.e., there exist noises in the supporting evidences and there might be no consensus existing in supporting evidences. (2) By removing the whole matching component, the performance of $MV_{-matching}$ has a significant drop as compared with *MV*. The results indicate that the traditional matching model important for assessing the goodness of an answer by capturing the relevance signals. (3) By removing the whole verification component, $MV_{-verification}$ degrade to previous BERT for matching the questions and the answer, which demonstrates that the verification component is necessary for measure the goodness of answers by considering the trustworthiness beyond relevance.

4.7 Analysis of MV Framework

In this section, we analyze the effect of the number of evidences, and show an example to give some intuition on how MV works.

4.7.1 Number of Supporting Evidences. We analyze the effect of the number of the evidences used in MV. We show performance on AmazonQA and QuoraQA with respect to numbers of evidences in Figure 4. We observe that the performance gets boosted when more evidences are incorporated into the verification component in the early stage. The reason may be that more supporting evidences can help us distill better consensus and provide more coverage over the answer. However, the performance gradually decreases when the number of evidence exceeds some threshold. Too much evidence increase the risk of introducing more irrelevant evidences which are harmful for consensus extraction. In practice, this hyper-parameter depends on the quality of the information repository and the ability of the search system.

4.7.2 Case Study. To better understand how different models perform, we present an example question from YahooQA with multiple answers and predictions as shown in Figure 5. We take one question "How do i connect my PS2 to a old tv?" accompanied with two human-posed answers. The answer "You need to buy a modulator ... walmart" is selected as the good answer by the user and the answer "1. Plug the VCR … Enjoy!" is evaluated on relevance to the question. Due to the limited space, we show two pieces of supporting evidences. We can see that both answers are quite relevant to the question. Without extending the quality measurement of an answer from topical relevance to trustworthiness, BERT predicts the latter answer as the good one. Our model predicts the previous one as the good one, which is consistent with the decision of the questioner . The results again demonstrate the effectiveness of MV by considering a good answer as both relevant and trustworthy.

Question: How do i connect my PS2 to a old tv?
Good Answer: You need to buy a modulator and connect that to the old tv through a cable cord, you can buy one for like 15 dollars at any walmart.

Relevant Answer: 1. Plug the VCR to the TV 2. Plug PS2 to VCR 3. Enjoy!

Evidence:Anyway, i have an old tv with only an antenna jack in the back and wish to connect my PS2 to it. I have tried already using a tv adapter used for the atari and a rca splitter.

Evidence: hey, I have an old ps2 slim however I am missing the cable needed to connect it to a tv. All I have is the power cord. I don't know what cable it is I need, how to connect it, or if I even can on my tv.

Our Model: Good Answer(0.985(s_m = 0.972,s_v = 0.993)) Relevant Answer(0.889(s_m = 0.992,s_v = 0.812))

Original BERT: Good Answer(0.873) Relevant Answer(0.993)

Figure 5: An example from YahooQA dataset.

5 CONCLUSION

In this paper, we proposed a novel matching-verification (MV) framework for answer ranking based on the consensus theory, which attempts to help us climb from relevance to trustworthiness. Specifically, our MV framework includes two major components. A matching component assesses the relevance of a candidate answer to a given question, and a verification component distills reliable consensus from noisy supporting evidences to verify the trustworthiness. Experimental results demonstrated that our model can well capture the trustworthiness of an answer to a given question, and outperform all the state-of-the-art baselines on three CQA answer ranking datasets. In the future work, we would like to explore other types of external knowledge for verification and design new structure for the verification procedure to capture more interaction information.

6 ACKNOWLEDGEMENT

This work was supported by Beijing Academy of Artificial Intelligence (BAAI) under Grants No. BAAI2019ZD0306, and funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61722211, 61773362, 61872338, 62006218, and 61902381, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, the National Key RD Program of China under Grants No. 2016QY02D0405, the Lenovo-CAS Joint Lab Youth Scientist Project, the K.C.Wong Education Foundation, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.

[2] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038* (2016).

[3] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *arXiv preprint arXiv:1906.04341* (2019).

[4] Charles LA Clarke, Gordon V Cormack, and Thomas R Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Vergilius Ferm. 1962. Consensus Gentium. In *Dictionary of Philosophy, a cura di Dagobert D. Runes. Totowa: Littlefield, Adams.*

[7] Harry G Frankfurt. 2010. *On truth.* Random House.

[8] Atsushi Fujii and Tetsuya Ishikawa. 2001. Organizing encyclopedic knowledge based on the Web and its application to question answering. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 196–203.

[9] Liang Ge, Jing Gao, Xiaoyi Li, and Aidong Zhang. 2013. Multi-Source Deep Learning for Information Trustworthiness Estimation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. Association for Computing Machinery, New York, NY, USA, 766–774. https://doi.org/10.1145/2487575.2487612

[10] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2019. A deep look into neural ranking models for information retrieval. *arXiv preprint arXiv:1903.06902* (2019).

[11] Kishaloy Halder, Min-Yen Kan, and Kazunari Sugiyama. 2019. Predicting Helpful Posts in Open-Ended Discussion Forums: A Neural Architecture. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 3148–3157. https://doi.org/10.18653/v1/N19-1318

[12] Richard L. Kirkham. 1992. Theories of Truth: A Critical Introduction.

[13] Jeongwoo Ko, Teruko Mitamura, and Eric Nyberg. 2007. Language-independent probabilistic answer ranking for question answering. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 784–791.

[14] Shanshan Lyu, Wentao Ouyang, Yongqing Wang, Huawei Shen, and Xueqi Cheng. 2019. What We Vote for? Answer Selection from User Expertise View in Community Question Answering. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1198–1209.

[15] Shanshan Lyu, Wentao Ouyang, Yongqing Wang, Huawei Shen, and Xueqi Cheng. 2019. What We Vote for? Answer Selection from User Expertise View in Community Question Answering. In *WWW '19*.

[16] Shanshan Lyu, Wentao Ouyang, Yongqing Wang, Huawei Shen, and Xueqi Cheng. 2019. What We Vote for? Answer Selection from User Expertise View in Community Question Answering. In *The World Wide Web Conference*. ACM, 1198–1209.

[17] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *AAAI*.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035.

[19] Jarutas Pattanaphanchai, Kieron O'Hara, and Wendy Hall. 2013. Trustworthiness Criteria for Supporting Users to Assess the Credibility of Web Information. In *Proceedings of the 22nd International Conference on WWW (WWW '13 Companion)*. Association for Computing Machinery, New York, NY, USA, 1123–1130.

[20] Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 464–471.

[21] Tetsuya Sakai, Daisuke Ishikawa, Noriko Kando, Yohei Seki, Kazuko Kuriyama, and Chin-Yew Lin. 2011. Using Graded-Relevance Metrics for Evaluating Community QA Answer Selection. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*. Association for Computing Machinery, New York, NY, USA, 187–196. https://doi.org/10.1145/1935826.1935864

[22] Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 458–467.

[23] Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge-aware attentive neural network for ranking question answer pairs. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 901–904.

[24] Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge-aware Attentive Neural Network for Ranking Question Answer Pairs. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 901–904.

[25] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 373–374.

[26] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of ACL-08: HLT*.

[27] Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 464–473.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 489–498.

[30] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378* (2016).

[31] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[32] Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, and Lin Sun. 2010. Modeling Semantic Relevance for Question-Answer Pairs in Web Social Communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 1230–1238. https://www.aclweb.org/anthology/P10-1125

[33] Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

[34] Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. 2013. Wisdom in the social crowd: an analysis of quora. In *WWW '13*.

[35] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. 22–32.

[36] Yu Wu, Wei Wu, Can Xu, and Zhoujun Li. 2018. Knowledge enhanced hybrid neural network for text matching. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[37] Yufei Xie, Shuchun Liu, Tangren Yao, Yao Peng, and Zhao Lu. 2019. Focusing Attention Network for Answer Ranking. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3384–3390.

[38] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. *arXiv preprint arXiv:1908.00300* (2019).

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

[40] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. 5754–5764.

[41] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1744–1753.

[42] Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. 2020. guided Helpful Answer Identification in E-commerce. In *Proceedings of The Web Conference 2020*. 2620–2626.

[43] Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. 2020. Review-Guided Helpful Answer Identification in E-Commerce. In *Proceedings of The Web Conference 2020 (WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2620–2626. https://doi.org/10.1145/3366423.3380015

[44] Wenxuan Zhang, Wai Lam, Yang Deng, and J. Ma. 2020. Review-guided Helpful Answer Identification in E-commerce. *Proceedings of The Web Conference 2020* (2020).

[45] Zhou Zhao, Hanqing Lu, Vincent Wenchen Zheng, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Community-Based Question Answering via Asymmetric Multi-Faceted Ranking Network Learning. In *AAAI*.

[46] Zhi-Min Zhou, Man Lan, Zheng-Yu Niu, and Yue Lu. 2012. Exploiting User Profile Information for Answer Ranking in CQA. In *Proceedings of the 21st International Conference on World Wide Web*. Association for Computing Machinery, 767–774.