Answering Product-related Questions with Heterogeneous Information*

Wenxuan Zhang, Qian Yu, Wai Lam
The Chinese University of Hong Kong
{wxzhang,yuqian,wlam}@se.cuhk.edu.hk

Abstract

Providing instant response for product-related questions in E-commerce question answering platforms can greatly improve users' online shopping experience. However, existing product question answering (PQA) methods only consider a single information source such as user reviews and/or require large amounts of labeled data. In this paper, we propose a novel framework to tackle the PQA task via exploiting heterogeneous information including natural language text and attribute-value pairs from two information sources of the concerned product, namely product details and user reviews. A heterogeneous information encoding component is then designed for obtaining unified representations of information with different formats. The sources of the candidate snippets are also incorporated when measuring the question-snippet relevance. Moreover, the framework is trained with a specifically designed weak supervision paradigm making use of available answers in the training phase. Experiments on a real-world dataset show that our proposed framework achieves superior performance over state-of-the-art models.

1 Introduction

To help potential consumers address their concerns during online shopping, many E-commerce sites now provide a community question answering (CQA) platform, where users can post questions for a specific product, and others can voluntarily answer them. Very often, it takes a long time for an asker to wait for an answer on such platforms. Therefore, automatically providing a proper response to a product-related question can greatly improve user online shopping experience and stimulate purchase decisions.

Several efforts have been made to tackle such product-related question answering (PQA) task (McAuley and Yang, 2016; Yu et al., 2018a; Gao et al., 2019; Chen et al., 2019b; Deng et al., 2020b). The existing methods can be generally categorized regarding the involved information source, i.e., from where the responses are obtained. A pioneer work by McAuley and Yang (McAuley and Yang, 2016) investigates answer selection via detecting clues from user reviews. From then on, the review set becomes a commonly used auxiliary information for predicting the answer types or distinguishing true answers from randomly sampled ones (Wan and McAuley, 2016; Yu and Lam, 2018). However, these methods are not feasible for newly-posted questions without candidate answers. A recent approach for PQA task is to directly extract review sentences as the response for a given question (Chen et al., 2019a). But it requires a large number of labeled question-review pairs, whose annotation is a time-consuming and laborious work. Other information sources, such as existing QA collections, are also exploited (Yu et al., 2018b), but relevant QA pairs are assumed to be always available for a new question in their setting, which is uncommon in practice.

Besides user reviews, another kind of information, namely product details provided by the manufacturer are always available and can be an important information source for addressing product-related questions. For example, considering the question "*How large is the keyboard*" for the product shown in Figure 1, the attribute-value pair "*Item Dimensions: 10.9×4.8×0.6 in*" from the specification table can be a good response. Such information can be essential for questions looking for factual type information due to their reliability and preciseness, but they are often underutilized in previous works. The above scenario motivates our task of answering product-related questions via exploit-

* 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 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 696–705
December 4 - 7, 2020.

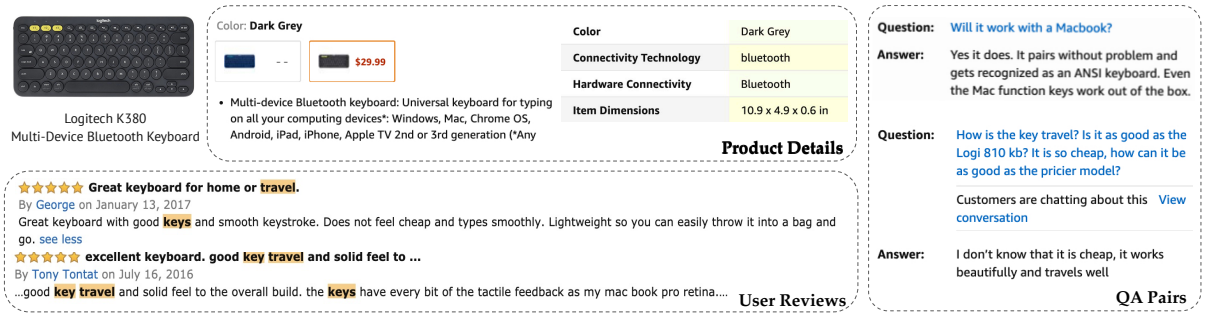

Figure 1: A sample E-commerce product associated with its product details, user reviews, and QA pairs

ing the information from both **product details** and **user reviews** to obtain relevant snippets serving as responses for improving user satisfaction.

This task presents some new research challenges: (i) The heterogeneity of candidate information needs to be appropriately handled. From the above example, we can see that there exists both attribute-value pairs and natural language texts as candidate responses, which implies that typical answer selection approaches (Tan et al., 2016; Wang et al., 2017; Rao et al., 2019) are incapable of handling the concerned task. (ii) Product details and user reviews contain different types of information, which are suitable for answering questions with different information needs. Returning to the example in Figure 1, considering a more subjective question asking about user experience *"How is the key travel"*, snippets from reviews such as *"...good key travel and solid feel.."* can provide more appropriate responses. Thus, we can observe that questions with different intents can be better answered by snippets from different sources, which should be exploited when measuring the question-snippet relevance. (iii) Training a model to capture the relevance between a question and a candidate snippet with typical supervised paradigms requires a large volume of labeled data. However, it is very time-consuming to manually label the question-snippet pairs in the PQA task due to the product-specific nature of questions and candidate snippets (Chen et al., 2019a), which demands a better solution for training such models.

To tackle these challenges, we propose a novel framework for the PQA task using Heterogenous Information via a Weak Supervision paradigm (HIWS). Given a product-related question, HIWS exploits the corresponding product details and user reviews to return a ranked snippet list serving as the response. Specifically, a heterogeneous information encoding component is first developed to encode different information formats into a unified representation composed of a free text sentence and a set of focused aspects. Then for measuring the question-snippet relevance, a gated fusion approach is designed to get aspect-enhanced representations. Also, a question intent analysis module is designed to better determine which information source is more suitable for providing responses. To handle the shortage of labeled data for model training, we develop a weak supervision paradigm making use of the original user-posted answers during training. Some external resources including pre-trained language models such as BERT (Devlin et al., 2019) are utilized to obtain weak supervision signals to facilitate the training process.

Our main contributions are as follows:

- We explore to utilize heterogeneous information including attribute-value pairs and natural language sentences from both product details and user reviews to tackle the PQA task.
- To handle the lack of labeled data, we design an effective weak supervision paradigm making use of available answers in training phase.
- Experiments on real-world E-commerce dataset show that our proposed model achieves superior performance over state-of-the-art models.

2 The Proposed Framework

For a product p, its associated information can be represented as a tuple $\mathcal{C}_p = (\mathcal{A}, \mathcal{D}, \mathcal{R})$, where $\mathcal{A} = \{(a_i, v_i)\}$ is a set of attribute-value pairs extracted from the corresponding specification table. $\mathcal{D} = \{d_i\}$ denotes the textual product description snippets represented by d_i, $\mathcal{R} = \{r_i\}$ denotes the review set composed of review snippets represented by r_i. Now given a question q regarding the product p, our task is to automatically rank the candidate

snippets in $\mathcal{C}_p$, which can either be a textual sentence from $\mathcal{D}$ or $\mathcal{R}$, or an attribute-value pair from $\mathcal{A}$ for providing responses to the question q.

As shown in Figure 2, HIWS mainly consists of three components: heterogeneous information encoding, question-snippet relevance matching, and automatic label construction. Concretely, the candidate snippets are first transformed into unified representations. Then we measure the question-snippet relevance both from their aspect-enhanced representations and the intent matching. The overall model is then trained using the automatically-constructed labels via making use of the original answer to the given question.

2.1 Heterogeneous Information Encoding

Heterogeneous Information Unification Given the heterogeneous candidate snippets including natural language sentences and attribute-value pairs, we transform them into unified representations. It can be observed that these two types of information are actually complementary to each other where the attribute term in an attribute-value pair can well indicate the major focus of such snippet, while a textual sentence can usually provide more detailed semantic information.

To highlight the focus of a natural language sentence $\bar{c} \in \mathcal{D} \cup \mathcal{R}$, we can extract m aspect terms:

$$c^a = \{c_1^a, c_2^a, ..., c_m^a\} = \mathrm{AE}(\bar{c}) \quad (1)$$

where $\mathrm{AE}(\cdot)$ refers to a reasonable aspect extraction algorithm such as (He et al., 2017) used in our experiments. c^a are the extracted m aspects. These extracted aspects are typically not exactly the same as the terms in the attribute set, but they play a similar role as characterizing the focus of the candidate snippet.

For an attribute-value pair $(a_i, v_i) \in \mathcal{A}$, since the main focus of such a snippet is already highlighted by the attribute term a_i, we directly treat a_i as the aspect c^a and construct a pesudo-sentence c^t by concatenating the attribute and value terms. To this end, any raw snippet $\hat{c} \in \mathcal{C}_p$, regardless of its original information type (i.e., whether it is an attribute-value pair or a natural language sentence), is mapped to a unified representation, denoted as c, as follows:

$$c = (c^t, c^a), \text{where } c^a = \{c_1^a, c_2^a, ..., c_m^a\} \quad (2)$$

where c^t is the textual sentence of $\hat{c}$. Such a unified representation facilitates effective processing of different input formats and also enriches the input representation for later process.

Snippet Encoding We next encode the unified candidate snippet representation c and the question q to vector representations. We first employ an embedding layer to transform each word into their corresponding word vector. The embedding of the word w is denoted as $e_w = [e_w^c; e_w^g]$, which is a concatenation of character-level embedding e_w^c and word-level embedding e_w^g. A bidirectional long short-term memory (Bi-LSTM) network is then employed to encode the local context information for each word in the question and the textual sentence c^t of the candidate snippet, which generates the context-aware question and snippet representations as follows:

$$h_i^q = \text{Bi-LSTM}(e_i^q, h_{i-1}^q), i \in [1, l_q] \quad (3)$$

$$h_i^c = \text{Bi-LSTM}(e_i^c, h_{i-1}^c), i \in [1, l_c] \quad (4)$$

where h_i^* is the hidden state of the encoder at the i-th time step. l_q and l_c are the length of the corresponding sequence. We denote the context-aware question and snippet representation as $H^q \in \mathbb{R}^{l_q \times d_h}$ and $H^c \in \mathbb{R}^{l_c \times d_h}$ respectively, where d_h is the number of hidden units of the LSTM network.

Besides the free text part, there are also m aspects for each candidate snippet c. They are useful when measuring the relevance between q and c since they can be regarded as the most salient part of the candidate snippet. Unlike a textual sentence, aspect terms are often quite short, so we directly employ the character-level embedding to transform each aspect term c_i^a to a vector representation denoted as h_i^a:

$$h_i^a = e_{c_i^a}^c = \mathrm{MaxPool}(\mathrm{Conv}(c_i^a)) \quad (5)$$

where $\mathrm{MaxPool}(\cdot)$ and $\mathrm{Conv}(\cdot)$ denote the max-pooling and convolutional operations (Kim, 2014).

2.2 Question-Snippet Relevance Matching

Aspect-enhanced Representations To utilize the aspect information, we design a gated attention mechanism to highlight the relevant information in the question q. Specifically, for the *k*-th word in the context-aware question representation, denoted as H_k^q, we measure the relative importance $\alpha_{k[i]}$ of this word given the *i*-th aspect term:

$$\alpha_{k[i]} = \frac{\exp\left((H_k^q)^T h_i^a\right)}{\sum_{j=1}^{l_q} \exp\left((H_j^q)^T h_i^a\right)} \quad (6)$$

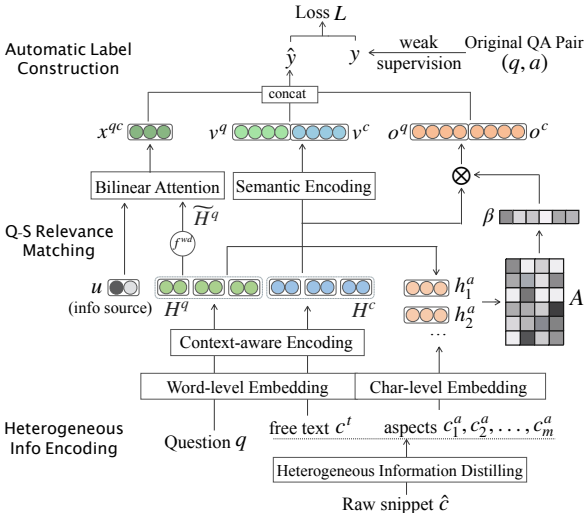

Figure 2: The architecture of proposed HIWS model

Since there are in total m aspects for a given candidate snippet c, we can similarly obtain $\alpha_{k[1]}, \alpha_{k[2]}, ..., \alpha_{k[m]}$ attention scores for the k-th question word. These attention scores reflect different relative associations of the concerned word with different aspects. Then for every word in the question q, we can obtain these attention scores, giving us an attention matrix $A \in \mathbb{R}^{l_q \times m}$. To get one compositive attention weight for each word in the question, we apply a gated fusion approach to combine these aspects. Specifically, a linear transformation is employed as a gate to learn an appropriate combination between these different attention weights as follows:

$$\beta = \tanh(W_a A^T + b_a) \tag{7}$$

where $\beta \in \mathbb{R}^{l_q}$ denotes the relative importance of each word in the question q, W_a and b_a are trainable parameters. Then we can utilize the combined attention weight to obtain an aspect-enhanced question representation o^q:

$$o^q = \sum_{k=1}^{l_q} H_k^q \cdot \beta_k \tag{8}$$

Here o^q represents the question representation with an enhancement from multiple aspects of the candidate snippet, which captures the relevance information between q and c from the view of aspect terms. Based on the intuition that explicitly highlighting these aspects in c^t is also helpful to capture its major information, we apply similar operations to H^c, giving an aspect-aware snippet representation o^c.

Question Intent Analysis for Multi-source Candidate Information The question intent helps identify what type of information the user is looking for and how to respond them. For example, it can be much more helpful to respond a question asking about personal experience with snippets from reviews. In contrast, the product details will be more suitable and convincing for a question looking for concrete product specifications. Thus, a question intent matching module is designed to detect such matching signals.

It can be observed that the beginning words of a question often have stronger ability for indicating the question intent. Thus, given the question representation H^q, a weight decay function $f^{wd}()$ is applied on it to emphasize the importance of the beginning words. Precisely, for the i-th word in the question, we multiply H_i^q by n^i, where $n \in (0, 1)$ can be set in advance such as $n = 0.9$ used in our experiments or learned with the model. Then we can obtain the encoded question representation r^q as follows:

$$\widetilde{H}_i^q = f_i^{wd}(H_i^q) = n^i \otimes H_i^q \tag{9}$$

$$r^q = \sum_{i=1}^{l_q} \widetilde{H}_i^q \tag{10}$$

where $\otimes$ refers to the element-wise multiplication. We denote the question representation after such transformation as r^q. Then given a one-hot feature vector $u \in \mathbb{R}^2$ of the candidate snippet c indicating its information source i.e., from product details or user reviews. A bilinear attention layer is employed to achieve the question intent matching analysis:

$$x^{qc} = \tanh\left(r^q W_m u + b_m\right) \tag{11}$$

where W_m and b_m are trainable parameters, x^{qc} denotes a low-dimensional vector reflecting the intent matching between the question and the candidate snippet.

Matching Signal Aggregation and Prediction After obtaining the aspect-enhanced representations and the question intent matching signals, we also employ a Siamese architecture to encode H^q and H^c with another Bi-LSTM encoder for capturing their main semantic information:

$$v^q = \text{Bi-LSTM}_{l_q}(H^q) \tag{12}$$

$$v^c = \text{Bi-LSTM}_{l_c}(H^c) \tag{13}$$

We use l_* as the subscripts in the above equations to differentiate it from Equation (3) indicating that only the last hidden state is taken as the encoded

representation. By utilizing the same sentence encoder, it helps map them into the same semantic space for determining their semantic relevance.

Then these different matching signals can be aggregated and fed to a MLP layer to make the final prediction $\hat{y}$:

$$\hat{y} = \text{MLP}([v^q; v^c; o^q; o^c; x^{qc}]) \quad (14)$$

where the aggregated vector contains matching features from different perspectives including the core semantic information v^q and v^c, the aspect-enhanced representations o^q and o^c which highlight the major focuses discussed in each sequence, as well as the question intent matching signals x^{qc} containing information about which information source is better for answering the concerned question regarding its intent.

The overall model is then trained to minimize the cross entropy loss between the predicted relevance score $\hat{y}$ and the automatically-constructed label y which will be introduced in the next section:

$$L = -\frac{1}{N}\sum_{n=1}^{N}[\hat{y}_n \log y_n + (1-\hat{y}_n)\log(1-y_n)] \quad (15)$$

where $\hat{y}_n$ and y_n denote the prediction and label of the n-th training instance, N is the total number of training instances.

2.3 Automatic Label Construction

In order to learn a matching function between the question and candidates, the most typical approach is to utilize a large number of annotated sentence pairs (Chen et al., 2019a) to conduct the training. However, this manual solution is not effective in PQA settings due to the large volume of candidate snippets and the product-specific nature of questions and candidates. Fortunately, we can take advantage of the original user-posted answers to their corresponding questions via a weak supervision paradigm during the training phase which has been successfully applied to provide imperfect labels but with far more less human efforts in many NLP tasks such as knowledge-base completion (Hoffmann et al., 2011) and sentiment analysis (Severyn and Moschitti, 2015b) etc.

Given a question q, we have its answer a during the training phase as auxiliary information to obtain the label y for the candidate snippet c. To make use of the information of the whole QA pair, the entire QA pair (q, a) is first fused to an integrated textual snippet p^{qa} with some heuristic rules (details are given in Sec 3.3). Then the problem of obtaining the relevance label between c and q are cast as measuring the relation between c^t with p^{qa}. We measure such relation from two perspectives, namely, syntactic relevance and semantic relevance.

Syntactic Relevance. Word overlapping between two text items can be a strong signal indicating their relevance. Here we adopt the idea of ROUGE (Lin, 2004) which is initially proposed for computing a recall-based word overlapping score to compute the syntactic-level relevance score s_1:

$$s_1 = \text{ROUGE-1}(c^t, p^{qa}) + \text{ROUGE-2}(c^t, p^{qa}) \quad (16)$$

where ROUGE-N refers to the overlap of N-grams between c^t and p^{qa}.

Semantic Relevance. To address the issue of the semantic gap between two text items, many word and sentence embedding models have been proposed and successfully applied to many NLP tasks recently. Here, we utilize some pre-trained text embedding models to compute the semantic relevance between the integrated QA snippet p^{qa} and c^t:

$$s_i = \cos(\text{Pre-TE}_i(p^{qa}), \text{Pre-TE}_i(c^t)) \quad (17)$$

where Pre-TE refers to a pre-trained text encoder. We adopt GloVe (Pennington et al., 2014), Elmo (Peters et al., 2018) and BERT (Devlin et al., 2019) in our experiments. $\cos(\cdot)$ denotes the cosine similarity score between the two encoded sentence representations. We denote the computed relevance scores with the aforementioned pre-trained models as s_2, s_3, s_4 respectively.

After obtaining these relevance signals, a small amount of human-annotated question-snippet pairs are used to train a simple classifier for learning to combine these signals into the single label y[1]. Note that it seems to be unnecessary to design any framework if a simple classifier with a few amount of labeled data and some pre-trained models can achieve a high accuracy. This is because we use the information of the entire QA pair to obtain the label y denoting the question-snippet relevance, which is different when we only have the question q and needs to retrieve relevant snippets during the testing phase. Thus a simple classifier with a few amount of labeled data can learn to integrate these relevance scores for the construction of "gold" labels with the help of original answers.

[1] 40 questions with their candidate snippets are annotated for this purpose, a SVM classifier is used in our experiment.

3 Experiments

3.1 Dataset

We perform experiments on real-world data to validate the model effectiveness. The question-answer pairs and reviews are drawn from the Amazon QA dataset (McAuley and Yang, 2016) and Amazon review dataset (Ni et al., 2019). Product details are crawled from the corresponding products' pages and incorporated into our dataset. In this way, we construct a heterogeneous dataset, which includes in total 5,395 QA pairs of 3,840 products spanning three product categories, namely, *"Cell Phones and Accessories"*, *"Sports and Outdoors"* and *"Tools and Home Improvement"*.

For each question, we first utilize the BM25 algorithm to conduct an initial filtering and collect the 50 top-ranked snippets from the corresponding product information as candidate snippets. After discarding empty or meaningless strings, we obtain 219,563 question-candidate snippet pairs in total. The dataset is split for training/validation/testing as 4,023 / 779 / 593 questions respectively, which results in 163,063 / 32,178 / 24,322 question-snippet pairs in each set. To obtain training and validation set, we utilize the weak supervision paradigm described in Sec 2.3 to automatically construct labels. For the testing set, in order to evaluate the effectiveness of the whole framework, the relevance labels between the questions and candidate snippets are annotated manually by two trained human annotators, the disagreements of the annotations are resolved by another experienced annotator

3.2 Baselines and Evaluation Metrics

To compare with our proposed framework, we adopt several strong baseline and state-of-the-art question answering models, including **CNN** (Severyn and Moschitti, 2015a), **QA-LSTM** (Tan et al., 2016), **MatchPyramid** (Pang et al., 2016), **BiMPM** (Wang et al., 2017), **Conv-KNRM** (Dai et al., 2018), **HCAN** (Rao et al., 2019) for comparisons. These models take the question and natural language sentence part of the candidate snippet as input, and are trained using the same automatically-constructed labels derived from original QA pairs as our proposed HIWS framework.

Two retrieval-based unsupervised models are also adopted: (1) **BM25**: It is a widely-used bag-of-words retrieval model. (2) **QCEM**: Question Candidate Embedding Matching is an unsupervised method that sums the word vectors of each sentence as the sentence embedding, and cosine similarity is utilized for predicting sentence relevance.

Table 1: Response Selection Performance

	MAP	MRR	P@5	P@10
BM25	0.417	0.549	0.296	0.234
QCEM	0.479	0.623	0.385	0.278
CNN	0.576	0.665	0.430	0.329
QA-LSTM	0.561	0.656	0.419	0.327
MatchPyramid	0.630	0.700	0.466	0.353
BiMPM	0.613	0.683	0.458	0.336
Conv-KNRM	0.615	0.696	0.457	0.337
HCAN	0.632	0.710	0.459	0.339
HIWS	**0.674**	**0.749**	**0.498**	**0.363**

For evaluation metrics, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision at N (P@N) are used to measure the performance. Precision at N (P@N) is the precision of the N retrieved snippets. We set N=5 and N=10 which correspond to P@5 and P@10 respectively in our experiments.

3.3 Implementation Details

For the automatic label construction, we first utilize the user-posted answer to paraphrase the question for obtaining the integrated snippet p^{qa} according to the part-of-speech tags and syntactic structure of the question with heuristic rules. For example, for a question *"does it have a front-facing camera?"* with the answer *"No."*, it will be combined to *"It does not have a front-facing camera"*.

For the network architecture, we initialize the word embedding layer with the pre-trained 300D GloVe word vectors (Pennington et al., 2014). The sizes of the CNN filters in the character-level embedding are set to $[2, 3, 4, 5]$, each with 75 filters, resulting in 300D character-level embedding for each word. The hidden dimension of the context-aware Bi-LSTM encoder is set to 150, with the dropout rate being 0.3. The hidden dimension of the sentence encoder in Eq. (12) is set to 64, with the dropout rate also being 0.3. The hidden dimensions of the MLP layer in the final prediction layer are set to 300 and 100 respectively, with ReLU as the activation function. All models are trained with the batch size of 100. The number of aspects m for each candidate snippet is set to be 3 which is a moderate number for a single sentence.

Table 2: Effectiveness of Weak Supervision Paradigm

	BiMPM		HCAN		HIWS	
	MAP	MRR	MAP	MRR	MAP	MRR
with QA	0.338	0.409	0.329	0.402	0.310	0.393
with SQS	0.443	0.492	0.432	0.495	0.479	0.556
with WS	**0.613**	**0.683**	**0.632**	**0.710**	**0.674**	**0.749**

Table 3: Ablation study for components in HIWS

Ablation of HIWS	MAP	MRR
w/o syntactic relevance score	0.273	0.434
w/o semantic relevance score	0.543	0.626
w/o question intent matching	0.667	0.737
w/o aspect-enhanced representations	0.631	0.704
HIWS	**0.674**	**0.749**

3.4 Quantitative Evaluation Results

Response Selection Performance The evaluation results are presented in Table 1, which demonstrates that our proposed HIWS achieves the best performance among all evaluation metrics compared with both retrieval-based solutions and supervised QA matching methods. We can observe that some simple QA models such as QA-LSTM and unsupervised models such as QCEM can still achieve reasonable performance. For those state-of-the-art models such as BiMPM and HCAN, although equipped with complicated network architecture, they do not perform as promising as expected. Such a result is due to the fact that these QA models merely focus on the matching between text items and ignore some important characteristics in the E-commerce scenario such as the heterogeneous information formats and multiple information sources of the candidate snippets. HIWS exploits such characteristics and utilize the extracted aspects to obtain enriched representations, leading to its superior performance.

Effectiveness of Proposed Weak Supervision Paradigm We investigate two alternative strategies for tackling the shortage of labeled data and compare them with our proposed weak supervision strategy to examine its effectiveness. The results on the same test set are reported in Table 2. Specifically, we train HIWS and two baselines, namely BiMPM and HCAN with different methods: "with QA" denotes training with the QA pairs instead of question-snippet pairs as in Table 1. We treat questions with their original answers as the positive samples and other randomly selected answers as negative samples for model training; "with SQS" refers to models which are first trained with QA pairs, then the **S**mall number of annotated **Q**uestion-**S**nippet pairs introduced in Sec 2.3 are used to fine-tune the model; "with WS" means the model is trained with the proposed weak supervision approach. Comparing these model variants, we can observe that models trained with the original QA pairs perform quite worse, showing the semantic gap between the original answers and the candidate snippets needs to be handled properly. Models with SQS outperform models with QA via fine-tuning with proper data, but it still failed to achieve satisfactory results due to the limited amount of labeled data. However, performance for all models can be improved with our proposed weak supervision paradigm, demonstrating its effectiveness on utilizing original answer information for bridging the connection between the question and snippets in the E-commerce settings.

Ablation Analysis We conduct ablation analysis to investigate the effectiveness of some important components in HIWS as shown in Table 3. We first create two sets of training labels whose construction step only involves one kind of relevance scores introduced in Sec 2.3, denoted as "w/o syntactic relevance score" and "w/o semantic relevance score" respectively. It can be observed that these two kinds of linguistic considerations, especially the syntactic relevance, are quite essential for automatically obtaining the labels for conducting training and thus directly influence the final performance of our model. Another two important components in HIWS are the aspect-enhanced representations and the question intent matching. As shown in Table 3, these two components contribute to some performance boost, especially the aspect-enhanced module. For constructing the variant model without aspect-enhanced representations, we still feed the embedded aspect h_i^a into the aggregation layer. Thus, even without considering the interaction between aspects and the question as in HIWS, this variant still outperforms some baselines.

Performance with Different Amount of Data We further investigate the robustness of HIWS via examining its performance with different amount of training data. The MAP and MRR scores under each product category are reported in Figure 3, where "w/ n data" refers to HIWS trained with n proportion of the entire training data. It can be observed that even when we use a moderate amount of training data such as 3/4 training data, the per-

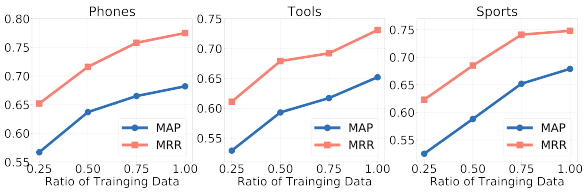

Figure 3: Performance with Different Amount of Data

formance does not drop significantly. Such results show the robustness of our proposed model implying that it can effectively utilize the available QA pairs to automatically construct useful training signals for learning the question-snippet relevance relation.

3.5 Case Study

To gain some insights into HIWS, we present two sample questions with the top-one responses given by HIWS and two strong existing methods in Table 4. The information sources of each snippet are marked, where $\mathcal{A}, \mathcal{D}, \mathcal{R}$ refers to attribute-value pairs, product textual descriptions and reviews respectively. Following each information source symbol, the correctness of the retrieved response is given. From the results, we can observe that HIWS successfully handles candidate snippets from different sources to answer the product questions with different information needs. For example, it precisely retrieves the corresponding attribute of the product for Question-1, which is more reliable and precise than the snippet retrieved from the review set by the existing models. Moreover, HIWS correctly handles the second question while the focused aspect is missing in responses from other methods. This is likely due to the aspect-enhanced representations for highlighting the major focus in the question and snippets. This result shows the necessity of effectively exploring different types of information of the concerned product instead of considering a single information source as in previous works.

4 Related Work

In recent years, many deep learning based methods have been proposed for the answer selection task in community question answering (CQA) platforms. These models can be generally categorized into two types according to their network architecture (Lai et al., 2018), namely Siamese networks (Tan et al., 2016; Mueller and Thyagarajan, 2016) and Compare-Aggregate networks (Wang et al., 2017; Rao et al., 2019; Deng et al., 2020a).

Table 4: Two sample questions with the top-one responses returned by HIWS and two existing models. The information sources and the gold labels of the snippets are also marked out in the parentheses at the end of each snippet respectively.

Question-1: What is the overall length of this bulb ?
HIWS: Product Dimensions : 6.5 x 2.5 x 2.5 inches ($\mathcal{A}$) (✓) **MatchPyramid:** I decided to try using these before i went more expensive route, the bulb are indeed quite large the length of a hand perhaps. ($\mathcal{R}$) (×) **HCAN:** I will update this review to render my durability opinion, one last note pay attention to the length of these bulb. ($\mathcal{R}$) (×)
Question-2: Will this work with my unlocked fire phone i have straight talk i want to switch to the amazon fire phone.
HIWS: Sim card will only work with an att compatible or unlocked gsm phone ($\mathcal{D}$) (✓) **MatchPyramid:** Keep your current phone number. Works with SIMs, IM, social networks, email, and web. ($\mathcal{D}$) (×) **HCAN:** I have tmobile and the service is not good in my area so i want to switch to straight talk ($\mathcal{R}$) (×)

Product-related Question Answering (PQA) problem has drawn a lot of attention recently, due to the increasing popularity of online shopping. Most of the existing works utilize reviews as their major information to provide responses for a given question. McAuley and Yang (2016) treat reviews as "experts" to handle the answer selection task. Later, product aspects are considered to further improve the performance (Yu and Lam, 2018). Chen et al. (2019a) propose to tackle PQA task by directly retrieving review sentences as answers. However, it requires a large number of labeled question-review pairs. Yu et al. (2018b) assume that relevant QA pairs are always available for a given question which can be utilized to provide the responses. Some other works formulate the PQA task as a reading comprehension problem (Xu et al., 2019), where the main focus is to extract a text span as the answer given a relevant review, which is unavailable in many cases. Given some successful applications of text generation models such as text summarization (Rush et al., 2015) and response generation (Tao et al., 2018), some models are proposed to generate an answer sentence (Gao et al., 2019; Chen et al., 2019b) given relevant product information, some later works specifically consider the user opinion information during such generation process (Deng et al., 2020b). Since most product-related questions are looking for diverse answers, we argue that information extracted from reliable sources is more effective and explainable

solution for the PQA task. More recently, some studies consider the answer helpfulness prediction task (Zhang et al., 2020b) and answer ranking problem (Zhang et al., 2020a) in the context of PQA, assuming the existence of user-provided answers to a given question. Different from them, we aim to provide instant responses for a newly-posted question in E-commerce.

5 Conclusions

We propose a novel framework for answering product-related questions via exploiting heterogeneous information including attribute-value pairs and free text sentences from both product details and user reviews. To tackle the shortage of labeled data, we design a weak supervision paradigm by making use of the existing QA pairs to automatically construct labels for training. Extensive experiments conducted on a real-word dataset demonstrate the superiority of our proposed framework.

References

Long Chen, Ziyu Guan, Wei Zhao, Wanqing Zhao, Xiaopeng Wang, Zhou Zhao, and Huan Sun. 2019a. Answer identification from product reviews for user questions by multi-task attentive networks. In *AAAI*, pages 45–52.

Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. 2019b. Review-driven answer generation for product-related questions in e-commerce. In *WSDM*, pages 411–419.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*, pages 126–134.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020a. Joint learning of answer selection and answer summary generation in community question answering. In *AAAI*, pages 7651–7658.

Yang Deng, Wenxuan Zhang, and Wai Lam. 2020b. Opinion-aware answer generation for review-driven question answering in e-commerce. In *CIKM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *WSDM*, pages 429–437.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*, pages 388–397.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Tuan Manh Lai, Trung Bui, and Sheng Li. 2018. A review on deep learning techniques applied to answer selection. In *COLING*, pages 2132–2144.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *WWW*, pages 625–635.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*, pages 188–197.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*, pages 2793–2799.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237.

Jinfeng Rao, Linqing Liu, Yi Tay, Wei Yang, Peng Shi, and Jimmy Lin. 2019. Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In *EMNLP-IJCNLP*, pages 5373–5384.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

Aliaksei Severyn and Alessandro Moschitti. 2015a. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, pages 373–382.

Aliaksei Severyn and Alessandro Moschitti. 2015b. Twitter sentiment analysis with deep convolutional neural networks. In *SIGIR*, pages 959–962.

Ming Tan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *ACL*.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*, pages 4418–4424.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *ICDM*, pages 489–498.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, pages 4144–4150.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *NAACL-HLT*, pages 2324–2335.

Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018a. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *WSDM*, pages 682–690. ACM.

Qian Yu and Wai Lam. 2018. Review-aware answer prediction for product-related questions incorporating aspects. In *WSDM*, pages 691–699.

Qian Yu, Wai Lam, and Zihao Wang. 2018b. Responding e-commerce product questions via exploiting qa collections and reviews. In *COLING*, pages 2192–2203.

Wenxuan Zhang, Yang Deng, and Wai Lam. 2020a. Answer ranking for product-related questions via multiple semantic relations modeling. In *ACM SIGIR*, pages 569–578.

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