Answer Ranking for Product-Related Questions via Multiple Semantic Relations Modeling

Wenxuan Zhang, Yang Deng, Wai Lam
The Chinese University of Hong Kong
{wxzhang,ydeng,wlam}@se.cuhk.edu.hk

ABSTRACT

Many E-commerce sites now offer product-specific question answering platforms for users to communicate with each other by posting and answering questions during online shopping. However, the multiple answers provided by ordinary users usually vary diversely in their qualities and thus need to be appropriately ranked for each question to improve user satisfaction. It can be observed that product reviews usually provide useful information for a given question, and thus can assist the ranking process. In this paper, we investigate the answer ranking problem for product-related questions, with the relevant reviews treated as auxiliary information that can be exploited for facilitating the ranking. We propose an answer ranking model named MUSE which carefully models multiple semantic relations among the question, answers, and relevant reviews. Specifically, MUSE constructs a multi-semantic relation graph with the question, each answer, and each review snippet as nodes. Then a customized graph convolutional neural network is designed for explicitly modeling the semantic relevance between the question and answers, the content consistency among answers, and the textual entailment between answers and reviews. Extensive experiments on real-world E-commerce datasets across three product categories show that our proposed model achieves superior performance on the concerned answer ranking task.

ACM Reference Format:
Wenxuan Zhang, Yang Deng, Wai Lam. 2020. Answer Ranking for Product-Related Questions via Multiple Semantic Relations Modeling. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401166

1 INTRODUCTION

Appropriately addressing users' concerns during online shopping can greatly improve their shopping experience and stimulate the purchase decisions. To this end, many E-commerce sites such as Amazon and eBay now offer product-specific community question answering platforms for users to post their questions and answer existing questions. Thanks to the convenience of such platforms, a question can typically get multiple user-provided answers. However, these answers, similar with other user-generated content, vary a lot in their qualities [19, 39] and suffer from typical flaws such as spams or even malicious content from the competitors.

Table 1: An example question of a headphone product from *Amazon* accompanied with its multiple answers. There are also some relevant review snippets of the question.

Question: Will these work with Android?
Relevant Review Snippets: - I have a Samsung Galaxy Note 4, I absolutely love these headphones, totally worth what I paid for them. - Easy to sync with an iPhone/Android phones. - Waste of money if using with an android device - I use mine with an android device and still works great. - ......
Answers: A1: The will and despite what some had said, they will sound exactly as they do on an Apple device. A2: Yes they will work with any Bluetooth capable device. A3: They will work but they won't sound as proficient as it would with an apple product. A4: I have Samsung Note 5, the headphones cannot connect via Bluetooth. A5: Its really not worth the money when using with an android.

Table 1 presents an example question, as well as its user-provided answers ranked by the community votes. It can be observed that even for this relatively objective question, the answers can vary diversely. Such variation in the answer contents and qualities motivates the task of automatically ranking these answers to improve user satisfaction. As shown in the example, there usually exists some relevant product reviews for the concerned question, which can provide useful information when ranking the answers if they are effectively utilized. Thus in this paper, we aim to tackle the task of answer ranking for product-related questions, with the associated product reviews treated as auxiliary information which can be exploited for assisting the ranking.

Answer selection methods have been extensively studied for tackling the answer ranking problem in retrieval-based question answering (QA) systems [11, 26, 40, 45]. Most of the existing works focus on measuring the semantic relevance between the question and a candidate answer, where the negative answers for training are usually randomly sampled from the whole answer pool [9, 29, 41] or chosen from irrelevant documents [50]. However, as can be observed from the above example, merely measuring the semantic relevance between the question and answer texts is no longer sufficient for the concerned answer ranking task in E-commerce settings

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14204418) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055093).

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401166

since all of the answers are written specifically for the question and thus most of them are supposed to be topically relevant to the question. Moreover, existing general answer selection models lack the capability of making use of the relevant product reviews during the ranking process. Recently, Zhang et al. [54] attempt to utilize reviews for identifying helpful answers in E-commerce. However, they ignore the relations among answers and reviews, which is essential for the concerned answer ranking problem. On the other hand, some product-related question answering (PQA) methods explore the utilization of product reviews to provide responses for a given question [5, 7, 13, 53], where some selected reviews can be served directly as the response [5, 52] or used to generate a response sentence based on a sequence-to-sequence neural model [7, 13]. These PQA methods assume the lack of user-written answers and thus turn to product reviews for help when addressing the given question. Such assumption neglects the large number of available answers provided by former buyers [3, 51], which can better answer the given question than the responses produced from the reviews.

In E-commerce settings, a key issue for the concerned ranking task is what makes an answer be a good one and how we can characterize such good answers via exploiting the associated reviews. Specifically, appropriately ranking the user-provided answers requires to model the complex semantic relations among these existing information sources, i.e., the question, answers, and product reviews. We argue that three kinds of semantic relations attach great importance: (i) Firstly, similar to general answer ranking problem [31], the *semantic relevance* of the answer content to the question is still essential for determining the ranking. At the same time, it is also necessary to consider the relevance between the question and reviews, which can alleviate the noise from irrelevant review information. (ii) Secondly, the *textual similarity* between each pair of answers indicates their content consistency, which can be regarded as a notion of peer reviews among the entire answer set for verifying the reliability of each answer [43]. Returning to the above example, since more answers agree that "the headphone is compatible with Android devices", the answer with the opposite opinion, such as A4, is thus less reliable than others. Similarly, measuring the similarity between reviews also helps cross verify the content consistency among them and hence captures the crowd's common opinions reflected in the review set for the given question. (iii) Thirdly, one may notice that such common opinions from the reviews often reveal authentic and general judgement from the whole community. Thus, the relationship between an answer and reviews can be modeled by the *textual entailment* relation [2, 6], which examines whether the opinion holding by an answer is coherent with common opinions reflected in the reviews. As shown in Table 1, the review snippets reveal that "the concerned headphone can work with Android", indicating that the first three answers, i.e., A1, A2 and A3, are more consistent with the opinions from the whole community, which leads to a higher rank. In summary, how to model and utilize the aforementioned multiple semantic relations among the question, answers, and relevant reviews for ranking the answers poses a main challenge.

In this paper, we propose an answer ranking model named MUSE for product-related questions, which comprehensively models **mu**ltiple **se**mantic relations among the question, answers, and relevant reviews. Concretely, we first conduct a word-to-word attention mechanism from the question to each individual answer during the answer encoding phase. Then the important information in the answer text and the relevance information with the question can be highlighted to obtain the textual features for each answer. Next, we construct a multi-semantic relation graph with the question, each answer and each review snippet as nodes. Then a customized graph convolutional neural network is designed for explicitly modeling the interrelationship between the nodes under multiple semantic relations. Precisely, for a specific answer, the textual features obtained in the earlier step are further refined by considering the semantic relevance with the question, the textual similarity with other answers and the textual entailment relation with relevant reviews. By modeling the relations between a given answer with other answers and relevant reviews, the coherence information between the concerned answer with the common opinion is accumulated to assist the ranking. Finally, we adopt a joint loss function, combining both pointwise and listwise learning approaches, to consider a specific answer both locally and globally in the entire answer set. To summarize, the contributions of this paper are as follows:

- We investigate the problem of ranking user-provided answers in E-commerce and propose a framework to jointly model multiple semantic relations including the semantic relevance between the question and answers, the textual similarity among answers, and the textual entailment between answers and reviews.
- We model both textual and interaction features of each answer to facilitate the ranking task. Importantly, a novel graph convolutional operation is designed to integrate the coherence information under different semantic relations.
- Experimental results on real-world E-commerce data across three product domains show that our proposed MUSE model achieves superior performance on the concerned task.

2 RELATED WORK

Answer Ranking. Answer selection has been extensively studied for solving the answer ranking problem in retrieval-based question answering systems as exemplified in the community question answering (CQA) [26] and factoid question answering [45]. The research on answer ranking has evolved from early information retrieval (IR) research, which primarily focused on feature engineering with syntactic or lexical approaches [34, 45]. In recent years, deep learning based answer selection methods make several breakthroughs and become the mainstream approach to tackle the answer ranking task. Most of the existing neural models adopt Siamese architecture [30, 35], attentive architecture [40] or compare-aggregate architecture [46, 47, 49] for modeling the semantic relevance between the question and answer without heavy feature engineering. Additionally, some latest studies also learn to rank question-answer pairs from different perspectives such as utilizing external knowledge [10], extracting length-adaptive features [37], modeling user expertise [24], and measuring answer novelties [17, 28].

There are also some works focusing on measuring the quality of answers or similar text content, then the predicted qualities can be used to rank them. Shah and Pomerantz [36] evaluate and predict answer qualities in the CQA platforms. Halder et al. [15] propose a neural model to predict the quality of a response post to the original post, with the awareness of several previous posts in the discussion forum. In the E-commerce scenario, some studies [4, 12] utilize product information such as product titles to predict the quality of a customer review. Recently, Zhang et al. [54] utilize user reviews for identifying helpful answers in the product question answering forums while they neglect the interrelationships among answers. In this work, we focus on the answer ranking problem in E-Commerce settings, where we wish to rank multiple user-provided answers for a product-related question with the help of relevant reviews.

Product-related Question Answering. Recent years have witnessed several successful applications in product-related question answering (PQA) problem. Most of the existing studies exploit customer reviews as major [5, 52, 53] or auxiliary resources [7, 13, 25, 44] for providing responses to the given question. McAuley and Yang [25] divide the question type into yes/no questions and open-ended questions and then tackle the yes/no type question as a classification task, aiming to predict the answer as "yes", "no" or "unsure" with the help of reviews. Following this direction, Yu and Lam [53] further consider the latent aspect information to improve the answer prediction performance. Some other works [5, 52] adopt the retrieval-based methods to retrieve certain review snippets serving as the response. For example, Chen et al. [5] propose a multi-task learning framework to identify reviews for a given question. Recently, some studies utilize the reviews to generate a sentence as the response based on the sequence-to-sequence architecture [7, 13]. Although most of these models utilize the review information, it is often assumed that the user-written answers are unavailable, which is different from our concerned task to directly rank these answers for the given question.

Graph Neural Networks. Graph Neural Networks (GNNs) [21] have been widely adopted to model graph structure data. Some latest studies exploit GNN in the IR-related tasks, which constructs text-based graphs to model the structural relation beyond the context itself. Li et al. [22] propose a large-scale anti-spam method based on GCN for detecting the spam advertisements. Sun et al. [38] propose a GCN encoder for keyphrase extraction that can effectively capture document-level word salience. Chen et al. [8] develop heterogeneous graph attention networks (HGAT) for user profiling. In this work, we explore the utilization of relational GNN [33] to model the interactions between information from different sources under different semantic relations in E-commerce settings.

3 MODEL

In typical E-commerce settings, given a product-related question q of a particular product, its answer set $\mathcal{A} = \{a_1, a_2, \dots\}$ contains $|\mathcal{A}|$ human-written answers. We can also obtain $|C|$ relevant review snippets $C = \{c_1, c_2, \dots\}$ to the question q. Our goal is to rank those answers in the answer set $\mathcal{A}$ with the review snippets C treated as auxiliary information that can be exploited to assist the ranking.

3.1 Model Overview

In this section, we introduce our proposed answer ranking model, MUSE, with modeling of multiple semantic relations among the question, answers, and relevant reviews. As shown in Figure 1, MUSE consists of three main components: textual feature modeling, multiple semantic relations modeling, and answer ranking.

We first employ a word-level attention to attend important and relevant information in the answers from the question during the textual feature modeling. Then a multi-semantic relation graph is constructed to model the multiple semantic relations among those texts from diverse information sources. Correspondingly, a customized graph convolutional network is developed to obtain the interaction-based features for each answer by aggregating the semantic relevance information between the question and answers, the textual similarity information among different answers, and the textual entailment information between the answer with each review snippet. Finally, after obtaining the textual features and interaction features, we design a joint loss function combining pointwise and listwise learning approaches to rank multiple answers.

3.2 Textual Feature Modeling

3.2.1 ***Context Modeling.*** Given a text sequence, which can be either the question sentence q, an answer sentence a_i, or a review snippet c_i, we first map each word in the sequence to a d_e-dimensional dense vector which can be initialized with pre-trained word vectors. We denote the embeddings for the word w_i as $e_i \in \mathbb{R}^{d_e}$. To model the context interactions among words in the sequence, a bi-directional LSTM encoder is then employed to transform each word into a context-aware vector representation:

$$v_i = \text{Bi-LSTM}(e_i, v_{i-1}), \tag{1}$$

where v_i is the hidden state of the Bi-LSTM encoder at the i-th time step. We thus denote the representation of the question, the i-th answer a_i and the i-th review snippet c_i after such context-aware encoding as V_q, V_{a_i} and V_{c_i} respectively:

$$V_* = [v_1^*, v_2^*, ..., v_{|*|}^*]; \quad * \in \{q, a_i, c_i\}, \tag{2}$$

$$V_q \in \mathbb{R}^{|q| \times d_h}, V_{a_i} \in \mathbb{R}^{|a_i| \times d_h}, V_{c_i} \in \mathbb{R}^{|c_i| \times d_h}, \tag{3}$$

where $|q|$, $|a_i|$ and $|c_i|$ denote the sequence lengths of the corresponding text sequences, d_h is the dimension of the hidden state of the Bi-LSTM encoder. To avoid notational clutter, we will omit the index of the answer and review snippet in this section since the same operations are conducted for each answer and review respectively. For example, we will use V_a to represent the context-aware representation for one particular answer instead of V_{a_i}.

3.2.2 ***Question-attended Answer Encoding.*** To explicitly highlight the core semantic units in the answer sentence and their relevance with the question, we employ a word-to-word attention mechanism to attend the important information in the question to each word of the answer. Specifically, for the i-th word in V_a, we consider its similarity with every single word in the question:

$$\alpha_i = \tanh(V_q \cdot v_i^a + b_a) \in \mathbb{R}^{|q|}, \tag{4}$$

$$\alpha'_{i,j} = \frac{\exp(\alpha_{i,j})}{\sum_k \exp(\alpha_{i,k})}, \tag{5}$$

$$o_i^a = \sum\nolimits_{j=1}^{|q|} \alpha'_{i,j} \cdot v_j^q \in \mathbb{R}^{d_h}, \tag{6}$$

where α_i denotes the similarity scores of the i-th word in the answer with every word in the question, $\alpha'_{i,j}$ is the normalized importance weight between the i-th word in the answer and the j-th word in the

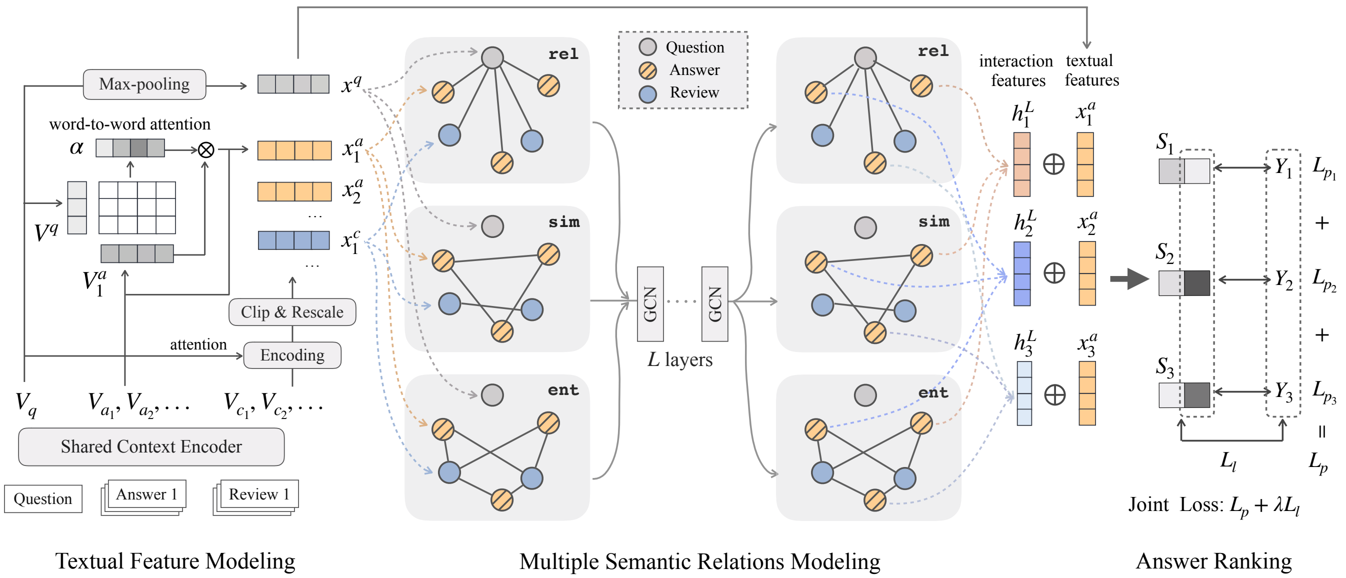


Figure 1: The architecture of the MUSE model composed of three main components, namely textual feature modeling, multiple semantic relations modeling, and answer ranking. Three types of semantic relations are specifically considered, including the semantic relevance (rel) between the question with answers and with reviews, the textual similarity (sim) among answers and among reviews, the textual entailment (ent) between answers and reviews.

question. Then for each word in the answer, we obtain a question-attended representation o_i^a as a weighted sum of the embeddings of question words. Next we concatenate the context-aware answer representation v_i^a and the question-attended answer representation o_i^a for the i-th word to obtain an enriched answer representation:

$$\hat{v_i^a} = \tanh(W_a \cdot [v_i^a; o_i^a] + b_{aa}), \tag{7}$$

where W_a and b_{aa} are trainable parameters, [;] denotes the concatenation operation. A max-pooling operation is then employed to obtain the encoded vector representations x_a and x_q for the answer a and the question q respectively:

$$x_a = \text{Max-Pool}([\hat{v_1^a}, \hat{v_2^a}, \ldots, \hat{v_{|a|}^a}]) \in \mathbb{R}^{d_h}, \tag{8}$$

$$x_q = \text{Max-Pool}([v_1^q, v_2^q, \ldots, v_{|q|}^q]) \in \mathbb{R}^{d_h}. \tag{9}$$

We denote the textual features obtained from the above operations for the i-th answer as x_i^a.

3.2.3 ***Clip-Rescale Attention for Review Encoding.*** For the relevant reviews, we also obtain encoded review representations as auxiliary information for assisting the ranking of answers. Although the review snippets in C are typically obtained with an initial retrieval process, there is still much noise contained in them, since these product reviews are originally written without explicitly responding to any question. To prevent the irrelevant information distracting the encoding, we employ a more aggresive clip-and-rescale attention mechanism inspired by [1] to obtain the question-attentive representation for each review snippet:

$$\beta = \text{softmax}(V_c W_c x_q^T + b_c) \in \mathbb{R}^{|c|}, \tag{10}$$

$$\beta' = \text{Rescale}(\beta \odot m) \in \mathbb{R}^{|c|}, \tag{11}$$

where W_c and b_c are trainable parameters, V_c denotes the context-aware representation for a review snippet c, β contains the original attention weights for each word in the review, $m = \{0, 1\} \in \mathbb{R}^{|c|}$ denotes a mask vector for β where only the index whose corresponding weight score are among the top k in β will be 1, and 0 otherwise, $\odot$ denotes the element-wise vector multiplication. Rescale() refers to the vector rescale operation: $\text{Rescale}(\text{v}) = v/\Sigma_i |v_i|$. Hence $\beta_j' \in \mathbb{R}$ refers to the importance score of the j-th word in the review snippet and is forced to be 0 for those unimportant words. Then we compute the review representation as the weighted sum of its context-aware representation:

$$x_c = \sum\nolimits_{i=1}^{|c|} \beta_i' v_i^c. \tag{12}$$

The same operations introduced above are conducted for every review snippet. We thus denote the vector representation after such textual feature modeling for the i-th review as x_i^c. Following similar notation convention, we denote the question representation as x^q.

3.3 Multiple Semantic Relations Modeling

3.3.1 ***Multiple Semantic Relations.*** Effectively ranking the answers requires to exploit the rich semantic relations among the question, answers, and reviews. To this end, we identify three types of semantic relations among these diverse information sources which are useful for ranking the user-provided answers:

(1) Semantic Relevance between the question and answer text is typically exploited in the general answer ranking task [31]. After bringing in review information, measuring the semantic relevance between the question and review snippets is also useful for alleviating the noise from irrelevant review information.

(2) Textual Similarity between each pair of answers can effectively measure their content consistency [43] and hence help identify the core opinions in the entire answer set for the given question. Similarly, considering such relation among review snippets in the review set can reveal the common opinions reflected in the reviews. **(3) Textual Entailment** relation between an answer and a review snippet indicates whether the answer is supported by that specific review, which is inspired by some attempts of utilizing textual entailment relation for general question answering problem [16, 42]. Concretely, we treat the review as external evidence to examine the opinion coherence of a given answer with the common opinions of the community.

To model these different semantic relations, especially capturing the coherence information of an answer with other answers and user reviews, it requires to aggregate the complex interactions among the existing information sources. Importantly, we can observe that these relations are closely connected and supposed to be modeled concurrently when ranking the answers. For example, each answer needs to be considered with different purposes when measuring its relation with the question, other answers, and review snippets. The coherence information from one relation can also affect its interaction with another information sources under different relations. Therefore, we propose a multi-semantic relation graph and utilize the graph convolutional networks (GNN) [21], which is shown to excel at aggregating the structural information from the neighborhoods, to capture the coherence information under different semantic relations.

3.3.2 ***Graph Construction.*** Formally, we denote an undirected graph with multiple semantic relations as $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{R})$, with nodes $n_i \in \mathcal{N}$, labeled edges (i.e., semantic relations) between node n_i and n_j as $(n_i, r, n_j) \in \mathcal{E}$, where $r \in \mathcal{R}$ is the relation type between two nodes. Then to construct the graph, we treat the question q, each answer sentence $a_i \in \mathcal{A}$ and each review snippet $c_i \in C$ as a node in $\mathcal{G}$. The total number of nodes is thus $1 + |\mathcal{A}| + |C|$. We initialize each node with their corresponding textual features x^* obtained from the textual feature modeling, which are encoded with their core semantic information.

To represent the multiple semantic relations, we make use of different adjacency matrices for the graph $\mathcal{G}$. Specifically, the relation type between two nodes $r \in \mathcal{R} = \{\texttt{rel}, \texttt{sim}, \texttt{ent}\}$, which represents the semantic relevance, textual similarity, and textual entailment relations respectively. Three adjacency matrices can thus be constructed for $\mathcal{G}$:

$$A^{\texttt{rel}}_{i,j} = \begin{cases} 1 & \text{if } n_i = q, n_j \in \{\mathcal{A}, C\} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$A^{\texttt{sim}}_{i,j} = \begin{cases} 1 & \text{if } n_i, n_j \in \mathcal{A} \text{ or } n_i, n_j \in C \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$A^{\texttt{ent}}_{i,j} = \begin{cases} 1 & \text{if } n_i \in \mathcal{A}, n_j \in C \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

3.3.3 ***Coherence Information Aggregation.*** Motivated by the Relational GCN [33], which shows good performance when considering the multiple relations between entities in a knowledge graph for the link prediction task, we develop a novel architecture for modeling the multiple semantic relations among the question, answers, and reviews for the concerned task. For a node n_i, the opinion coherence information is aggregated from its neighboring nodes:

$$h_i^{(l+1)} = \mathrm{ReLU}\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \Lambda^r_{i,j} W_r^{(l)} h_j^{(l)} + W_s^{(l)} h_i^{(l)} \right), \tag{16}$$

where $h_i^{(l)}$ is the hidden state of the node n_i at the l-th layer of the network, $\mathcal{N}_i^r$ denotes the neighboring indices of the node n_i under the relation r, $W_r^{(l)}$ and $W_s^{(l)}$ are trainable parameters representing the transformation from neighboring nodes and from the node n_i itself. $\Lambda^r_{i,j}$ is a normalization constant such as $\Lambda^r_{i,j} = 1/|\mathcal{N}_i^r|$ in [33]. To avoid the scale changing of the feature representation as commonly observed to be harmful for the performance, we apply a symmetric normalization transformation:

$$\Lambda^r = D_r^{-1/2} A^r D_r^{-1/2}, \; r \in \{\texttt{rel}, \texttt{sim}, \texttt{ent}\}, \tag{17}$$

where A^r is the adjacency matrix under the relation $r \in \mathcal{R}$, D_r is the corresponding degree matrix of A^r.

Unlike the basic GCNs using one convolutional filter matrix to model the feature transformation, the aggregation operation in Equation (16) employs different weight matrices $W_r^{(l)}$ for different semantic relations in each layer, which can capture the coherence information explicitly under different relations. Besides, a self-connection weight matrix $W_s^{(l)}$ is utilized to control how much information in the node n_i itself at each update should be kept.

At each step, the representations of the answer nodes are enriched by their neighbourhoods. Specifically, the first layer takes the textual feature vectors obtained from Section 3.2 as the input for each node, i.e. for the question node: $h^{(0)} = x^q$, for the node of the i-th answer: $h^{(0)} = x_i^a$ and for the node of the i-th review snippet: $h^{(0)} = x_i^c$. Then the transformation in Equation (16) can be stacked up to L layers to include the dependencies across multiple relational steps. We take the output of each answer node at the last layer as their interaction features and denote h_i^L as the feature representation for the i-th answer.

3.4 Answer Ranking

For each answer a_i, after obtaining the textual feature x_i^a and the interaction features h_i^L, they are then concatenated and fed to a MLP with one hidden layer to get the final prediction scores:

$$S_i = \mathrm{MLP}([x_i^a; h_i^L]) \in \mathbb{R}^2, \tag{18}$$

where S_i is the prediction vector for the i-th answer. We denote the concatenation of the prediction vectors of the entire answer set as $S \in \mathbb{R}^{|\mathcal{A}| \times 2}$. Finally, we employ a joint loss function, combining the pointwise and listwise learning approaches, to conduct training for learning to rank the answers.

3.4.1 ***Pointwise Loss Function.*** One of the most commonly used training strategies for answer ranking problem is the pointwise learning approach. Specifically, for each answer a_i, a softmax function is applied to its prediction vector S_i to obtain the predicted distribution $\hat{S_i}$ = Softmax(S_i) $\in \mathbb{R}^2$. Then each answer is considered separately and the cross-entropy loss is computed:

$$\mathcal{L}_p = \frac{1}{|\mathcal{A}|}\sum\nolimits_{i=1}^{|\mathcal{A}|}(-Y_i \; log \; \hat{S}_i), \tag{19}$$

where $Y_i \in \mathbb{R}^2$ denotes the one-hot encoding of the label for the i-th answer. Then the total loss $\mathcal{L}_p$ for each question is the average of the cross-entropy loss of all its answers.

3.4.2 ***Listwise Loss Function.*** Another choice of learning approach is the listwise method considering all candidate answers to the given question at the same time. Given the prediction matrix S, we first normalize the prediction scores among all answers:

$$\hat{y} = \frac{[S_{1,1}, S_{2,1}, ..., S_{|\mathcal{A}|,1}]}{||[S_{1,1}, S_{2,1}, ..., S_{|\mathcal{A}|,1}]||_p} \in \mathbb{R}^{n_a}, \tag{20}$$

where $S_{i,1}$ is the unnormalized prediction score of answer a_i being a positive answer, p is set to 1 in the experiments to compute the vector norm. Similarly, we also normalize the whole label list $y \in \mathbb{R}^{|\mathcal{A}|}$ of all answers with $y' = y/||y||_p$ where $y_i = \{0, 1\}$ is the raw label for the i-th answer. Then we can compute the listwise loss for a given question with Kullback-Leibler divergence:

$$\mathcal{L}_l = \frac{1}{|\mathcal{A}|} KL\_Div(\hat{y}||y') = \frac{1}{|\mathcal{A}|}\sum\nolimits_{i=1}^{|\mathcal{A}|} \hat{y}_i \; log(\frac{\hat{y}_i}{y'_i}). \tag{21}$$

3.4.3 ***Joint Loss Function.*** The pointwise and listwise learning approaches consider a specific answer locally and globally in the entire answer set respectively. In the pointwise learning approach, we focus on each individual answer locally, and the goal is to accurately predict their corresponding labels. In the listwise method, we examine the entire answer list globally, attempting to differentiate the good and bad answers, and making the former rank higher. In order to combine the strengths of both of them, we propose to employ a joint loss function in this work.

Specially, we combine these two types of loss functions to a joint loss $\mathcal{L}$ to train our proposed MUSE model. The above introduced two loss functions in Equation (19) and Equation (21) are for one single question (i.e. one data instance) in the dataset. Then for the whole dataset with in total $|Q|$ questions, the joint loss function is:

$$\mathcal{L} = \frac{1}{|Q|}\sum\nolimits_{i=1}^{|Q|}(\mathcal{L}_{p_i} + \lambda\mathcal{L}_{l_i}) + \eta \; ||\Theta||_2, \tag{22}$$

where $\mathcal{L}_{p_i}$ and $\mathcal{L}_{l_i}$ are the pointwise and listwise loss functions for the i-th question respectively. λ is a hyper-parameter for balancing between these two loss functions, η is the L2 regularizer weight, Θ is the set of all trainable parameters in the model.

4 EXPERIMENT SETUP

4.1 Datasets

We evaluate our proposed model on Amazon QA dataset [44], and utilize three product categories with the largest number of question-answer pairs for evaluation, including *Electronics, Home and Kitchen, Sports and Outdoors.* The dataset contains questions accompanied with their multiple user-written answers. The product ID of each question is then utilized to align with the Amazon review dataset [18, 27] for obtaining the corresponding reviews for the question. Since an entire raw review can be lengthy and talks about multiple aspects of the concerned product, each review text is chunked into snippets at the sentence level. Then for each question, we adopt BM25 to rank all the review snippets and collect the top 5 relevant snippets for each question in our experiments.

Table 2: Statistics of data splits of three product categories.

Category		# Product	# Q	# A	# Pos A
Electronics	Train+Val	11,172	15,547	80,115	28,919
	Test	1,657	1,727	8,823	3,184
Home	Train+Val	8,590	12,731	66,956	24,838
	Test	1,349	1,414	7,461	2,801
Sports	Train+Val	4,949	6,952	35,858	13,230
	Test	746	772	4,065	1,511

Similar to previous works [15, 54], we treat the user votes from the community as a proxy of the gold label for the quality of an answer. Thus, the answer whose number of positive votes is greater than the number of negative votes is treated as a high-quality (positive) answer, otherwise it is treated as a negative one. We split 10% of the dataset for each product category for testing and the rest is used for training and validation. The statistics are summarized in Table 2, including the number of products (# Product), questions (# Q), answers (# A), and positive answers (# Pos A).

4.2 Baselines and Evaluation Metrics

We compare our proposed MUSE model with some traditional and state-of-the-art methods. To conduct a more comprehensive comparison, we slightly modify some models to take the advantage of utilizing relevant reviews as one of their inputs.

- **BM25** [32]: It is a widely-used retrieval model for ranking candidate answers given a question.
- **CNN** [35]: It employs a CNN-based Siamese network to encode QA pairs for ranking the answers.
- **Attentive-BiLSTM** [40]: It utilizes a bidirectional LSTM as well as an attention mechanism to measure the relevance between the question and answer text.
- **aNMM** [48]: It is an **a**ttention based **N**eural **M**atching **M**odel, which employs a value-shared weights scheme and a gated attention network to improve the ranking performance.
- **BiMPM** [47]: **Bi**lateral **M**ulti-**P**erspective **M**atching is one of the state-of-the-art models in many retrieval based QA tasks. It matches QA sentence pair from multiple perspectives.
- **HCAN** [31]: **H**ybrid **C**o-**A**ttention **N**etwork is one recent model for modeling short text relations. It combines the semantic matching and relevance matching components to complement each other for better performance.
- **PRHNet** [12]: It is one of the state-of-the-art models for predicting the quality of product reviews. We concatenate the QA pair and treat it as a single review and utilize relevant reviews as the "product information" in the original model.
- **PHP** [15]: **P**ost **H**elpfulness **P**rediction is one recent model for predicting the quality of a replying post to an initial post in the online discussion forum. We treat the question and answer as the original and replying post respectively, and treat reviews as the previous posts used in the model.

For our proposed MUSE model, we use different suffixes to denote the variants of it trained with different learning approaches, where

Table 3: Answer ranking results of MUSE and baseline models. † denotes that MUSE-Joint-Loss model achieves better performance than the strong baseline PHP with statistical significance test for $p < 0.05$.

	Electronics				Home & Kitchen				Sports & Outdoors			
	MAP	MRR	P@1	P@3	MAP	MRR	P@1	P@3	MAP	MRR	P@1	P@3
BM25 [32]	0.571	0.576	0.380	0.383	0.585	0.592	0.406	0.397	0.586	0.589	0.384	0.398
CNN [35]	0.633	0.668	0.460	0.411	0.638	0.675	0.474	0.407	0.628	0.664	0.452	0.407
aNMM [48]	0.619	0.651	0.445	0.386	0.633	0.670	0.465	0.413	0.624	0.659	0.448	0.403
Att-BiLSTM [40]	0.642	0.671	0.464	0.408	0.639	0.673	0.471	0.416	0.633	0.665	0.464	0.408
BiMPM [47]	0.647	0.678	0.480	0.405	0.656	0.688	0.491	0.425	0.636	0.680	0.482	0.409
HCAN [31]	0.643	0.676	0.472	0.412	0.659	0.686	0.492	0.429	0.632	0.666	0.459	0.404
PRHNet [12]	0.646	0.677	0.478	0.406	0.649	0.683	0.483	0.421	0.634	0.669	0.469	0.405
PHP [15]	0.652	0.679	0.475	0.414	0.648	0.681	0.484	0.421	0.638	0.667	0.463	0.409
MUSE-Pointwise	0.663	0.693	0.504	0.425	0.679	0.710	0.521	0.450	0.649	0.684	0.482	0.431
MUSE-Listwise	0.678	**0.715**	**0.539**	0.417	0.675	0.712	**0.527**	0.443	0.657	0.688	0.491	0.435
MUSE-Joint-Loss	**0.695**†	0.711†	0.511†	**0.450**†	**0.693**†	**0.714**†	0.518†	**0.466**†	**0.661**†	**0.694**†	**0.498**†	**0.437**†

"-Pointwise", "-Listwise" and "-Joint-Loss" refer to training with the pointwise, listwise, and joint loss function, respectively.

For evaluation metrics, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), as well as Precision@N (P@N) are adopted to measure the model performance on ranking user answers. We take $N = 1$ and $N = 3$ in the experiments for P@N metric, which correspond to two common use cases in E-commerce: P@1 measures the performance when only a single answer is paired with each question in the product main page[1] and P@3 measures how well the top three answers for each question in the detailed question page containing all user-provided answers[2].

4.3 Experiment Configurations

For the network configurations, we tuned hyper-parameters with the validation set. Specifically, the hidden dimension of the Bi-LSTM context encoder is set to 100, the hidden size of the weight matrix W_a in Equation (7) is 200, k is set to 8 for alleviating the noise in reviews. In the semantic relation modeling, we stack two GCN layers to obtain the interaction features for the answers, i.e. $L = 2$, where the hidden dimension of each layer is set to 150 and 100 respectively. The hyper-parameter λ used to balance between two loss functions in Equation (22) is set to 2 and η is set to 0.001.

For the training process, we initialize the word embedding layers of all neural models with the pre-trained 300D GloVE word embeddings[3]. We adopt the Adam optimizer [20] to train all learnable parameters and the batch size is set to 50. All the network weights W_* are initialized randomly from Xavier uniform distribution [14].

5 RESULTS AND ANALYSIS

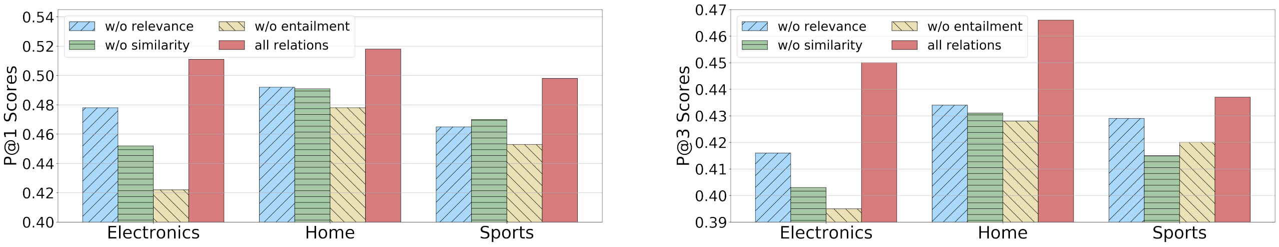
5.1 Answer Ranking Performance

The answer ranking results among three product categories in terms of MAP, MRR, P@1, and P@3 scores are summarized in Table 3. It shows that MUSE outperforms all baseline models on each dataset. Also, we conduct a statistical significance test comparing MUSE with PHP. The results indicates that MUSE achieves better performance than PHP with statistical significance test at $p < 0.05$.

There are several notable observations from the results: (1) Compared to the basic BM25 model, we can see that deep learning models generally provide strong baselines for the concerned ranking task. In particular, BiMPM and HCAN models outperform other answer selection models by taking into account deeper and boarder semantic relevance information. (2) Models with consideration of relevant review information, e.g. PHP model and our proposed MUSE model, can generally achieve better performance. Such results demonstrate that merely considering the semantic relevance between the question and answer text is not sufficient for ranking the user-provided answers in E-commerce settings. (3) Our proposed MUSE model consistently and substantially outperforms all the baselines across three categories. This result shows that carefully modeling the rich semantic relations among the available information sources, i.e. the question, multiple answers, and relevant reviews, is necessary for effectively ranking the answers. Importantly, MUSE utilizes the multi-semantic relation graph to model the coherence information between each specific answer with the common opinions reflected in the entire answer set and reviews, which leads to its superior performance when ranking user answers in E-commerce scenario.

Comparing the performance between different variants of MUSE model, it can be observed that training with the joint loss function (i.e. MUSE-Joint-Loss) generally achieves better performance than learning with the pointwise approach (i.e. MUSE-Pointwise) or the listwise approach (i.e. MUSE-Listwise). MUSE-Joint-Loss model combines the advantages of two learning approaches and considers an answer both from the perspective of its own label and from the perspective of the labels of the entire answer list, thus it achieves better results among the majority of cases. Notably, MUSE-Listwise largely outperforms existing models regarding the MRR and P@1 scores. For example, it obtains about 6% absolute improvement of P@1 score on the *Electronics* category compared with the best performance given by the baseline model. The reason is that by normalizing the prediction list of all answers and minimizing its difference with the label list, we push the positive answers to have

[1] e.g., https://www.amazon.com/dp/B07DLPWYB7?th=1#Ask
[2] e.g., https://www.amazon.com/ask/questions/Tx12DHUXVP6P535/ref=ask_ql_ql_al_hza
[3] http://nlp.stanford.edu/data/glove.6B.zip



(a) P@1 scores for three datasets

(b) P@3 scores for three datasets

Figure 2: Effect of different semantic relations in terms of P@1 and P@3 scores among three product categories

Table 4: Ablation study on the *Electronics* category

Model Variants	MAP	MRR	P@1	P@3
MUSE-Joint-Loss	**0.695**	**0.711**	**0.511**	**0.450**
- w/o textual feature	0.649	0.681	0.478	0.409
- w/o interactive feature	0.646	0.673	0.468	0.411
- w/o Q-to-A attention	0.656	0.688	0.490	0.414
- w/o Q-to-C attention	0.663	0.692	0.497	0.419

relatively higher scores. Thus the model can find one positive answer more easily and rank it as the top answer, which leads to better MRR and P@1 metrics.

5.2 Ablation Study

5.2.1 ***Impact of Main Components of MUSE.*** We perform ablation studies by leaving out some important components in the proposed MUSE model to investigate their effectiveness. The results on the largest *Electronics* dataset are presented in Table 4. We first create two variant models by discarding the textual feature x_i (denoted as w/o textual feature) and the interaction feature h_i (denoted as w/o interaction feature) for a specific answer when obtaining the prediction score S_i in Equation (18) respectively. From the results, we can find that both of them play an important role in contributing useful answer representations for the ranking. Specifically, the model without interaction feature suffers a slightly larger performance decrease, which is likely due to the fact that the textual features x_i is used as the initialization to compute h_i in the multiple semantic relation modeling so that some textual information encoded in x_i can be preserved in the ranking process.

In addition, to testify the usefulness of the question attention operation during the textual feature modeling of the answers and reviews, we create two variant models by directly conducting a max-pooling operation on V_a and V_c in Equation (2) to get x_a and x_c respectively, instead of employing attention mechanism (denoted as "w/o Q-to-A attention" and "w/o Q-to-C attention" in Table 4 respectively). It can be observed that both lead to a performance decrease, indicating that utilizing the question to attend the important information during the encoding phase of the answer and review sentences is useful for capturing relevant and important information for the subsequent learning process.

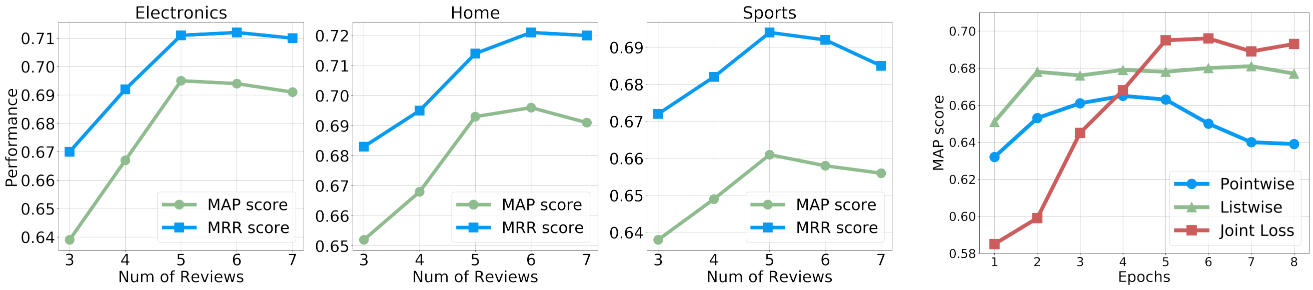
Especially, we can notice that leaving out the question attention of the answers results in a larger performance decrease. This result shows that the core semantic information in answers is still essential for modeling the textual representations for answer ranking.

5.2.2 ***Impact of Different Semantic Relations.*** To better rank the multiple answers for a given question, we model the multiple semantic relations among the question, answers, and relevant reviews in this paper. Thus, we examine the effect of each semantic relation during the graph construction phase in this section by removing one relation at each time. We present P@1 and P@3 scores among three product categories in Figure 3, where "w/o relevance", "w/o similarity" and "w/o entailment" denote the MUSE-Joint-Loss model without the semantic relevance, textual similarity, and textual entailment relation when constructing the multi-semantic graph in Section 3.3.2 respectively. Also "all relation" refers to the performance of the model with all three relations. We can see that each of the semantic relations contributes to the final ranking performance and discarding any of them leads to performance degradation. This result illustrates the importance of explicitly modeling the complex relations among the multiple information sources in E-commerce scenario. In addition, it can be noticed that the entailment relation attaches more importance than the other two relations, which validates the necessity to utilize relevant reviews as external sources for modeling the opinion coherence between a concerned answer with the common opinion. Moreover, discarding the semantic relevance relation leads to the least performance decrease since the reviews obtained by an initial retrieval process are somehow already related to the question.

5.3 Analysis of MUSE model

5.3.1 ***Number of Reviews.*** The relevant reviews are utilized as important external information in our concerned task. The proposed MUSE model aggregates common opinions with the review-review similarity relation and models the opinion coherence of an answer with the review-answer entailment relation. In this section, we vary the number of review snippets used in the model, i.e. the value of $|C|$, to investigate its effect on the model performance. The MAP and MRR scores with different number of reviews used in the model on three datasets are presented in Figure 3a.

We can see that, as expected, the performance of the model is getting better when more review information is utilized at the beginning. However, both MAP and MRR scores become generally unchanged (e.g. on the *Electronics* and *Home* dataset) or even slightly decrease (e.g. on the *Sports* dataset) when we further increase the number of reviews. On one hand, more reviews can provide more



(a) (b)

Figure 3: Analysis of MUSE model. Figure (a) shows the performance with respect to the number of reviews utilized by MUSE on three datasets. Figure (b) presents the MAP score on the test set at each epoch of MUSE with different learning approaches.

comprehensive common opinions from the community to help rank the candidate answers, leading to the performance improvement at the beginning. On the other hand, increasing the number of reviews used in the model also introduces more parameters and leads to the overfitting issue for those smaller datasets such as *Sports*.

5.3.2 ***Different Learning Approaches.*** We compute the MAP scores at each epoch on the test set of the proposed MUSE model trained with different loss functions. The results on the largest *Electronics* dataset are shown in Figure 3b. It can be observed that MUSE trained with listwise loss function converges in fewer epochs than the other two approaches and is less affected by the overfitting issue, which is consistent with some observations from previous studies [1, 23]. On the contrary, the model trained with pointwise learning approach is more likely to overfit after a few epochs. Such a phenomenon is likely due to the imbalance proportion between the high quality and low quality answers. Thus the model which is trained to only recognize the label of each answer individually will tend to predict an answer as a negative one so as to minimize the overall cross-entropy loss. For the model trained with the joint loss function, it is robust to the overfitting problem but converges at a relatively slow pace compared with the listwise learning approach.

5.4 Case Study

To gain some insights into the proposed MUSE model, we present a sample case in Table 5, which includes a question of an *egg cooker* product, its multiple user-provided answers, as well as the relevant review snippets. The answers $a_1, ..a_4$ are ranked by their original community votes. We also present the ranks given by MUSE and two strong baseline models, namely HCAN and PHP for each answer, where "Rx" denotes that the answer is ranked at the x-th position by the corresponding model. From the results, we can see that HCAN performs poorly on this case since it only considers the semantic relevance between the answer with the question text, while all the associated answers are quite topically relevant to the given question. Besides, the PHP model, which incorporates review information into the modeling, also fails to ranks all answers correctly, indicating that the semantic relations need to be appropriately exploited. The proposed MUSE model utilizes the answer-answer similarity and review-review similarity relations to capture the common opinion that "the concerned egg cooker cannot turn off automatically". The answer-review entailment relation can then help examine the opinion coherence between each specific answer with the common opinion and hence help the ranking process. Therefore, it successfully ranks all answers in this case. This real-world example indicates the importance of taking review information into consideration. More importantly, carefully modeling the complex semantic relations between the question, answers, and reviews is essential for tackling this task in E-commerce settings.

Table 5: A sample case of multiple answers ranked by their original community votes, as well as their predicted ranks by MUSE and two baseline models.

Question: Is it automatic shut off?			
Relevant Review Snippets C: c_1: A buzzer sounds to let you know the eggs are done c_2: When the alarm sounds you need to turn it off and open it... c_3: I would have liked the cooker to turn off automatically but instead a bell rings until you turn if off. c_4: Also, by the time the timer goes off, the hot pan has a burning smell. c_5: and it turns off itself after the bell rings.			
Answers	**HCAN**	**PHP**	**MUSE**
a_1: No, it beeps until you turn it off.	R3	R1	R1
a_2: No it’s not but it beeps very loud.	R4	R4	R2
a_3: Yes, and it works very well.......we just had poached eggs yesterday. recommend this product :)	R1	R3	R3
a_4: Yes there’s an automatic shut-off when the cooking cycle is finished.	R2	R2	R4

6 CONCLUSIONS

We investigate the answer ranking problem for product-related questions in this paper. To tackle the ranking task in E-commerce settings, we propose a framework named MUSE to jointly model the multiple semantic relations among the question, answers, and relevant reviews. MUSE employs a novel graph convolutional operation customized to integrate the coherence information under different semantic relations to facilitate the ranking task. Extensive experiments on real-world E-commerce datasets show that our proposed model achieves superior performance compared with some strong baseline models.

REFERENCES

[1] Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A Compare-Aggregate Model with Dynamic-Clip Attention for Answer Selection. In *CIKM*. 1987–1990.

[2] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. 632–642.

[3] David Carmel, Liane Lewin-Eytan, and Yoelle Maarek. 2018. Product Question Answering Using Customer Generated Content - Research Challenges. In *SIGIR*. 1349–1350.

[4] Cen Chen, Minghui Qiu, Yinfei Yang, Jun Zhou, Jun Huang, Xiaolong Li, and Forrest Sheng Bao. 2019. Multi-Domain Gated CNN for Review Helpfulness Prediction. In *WWW*. 2630–2636.

[5] Long Chen, Ziyu Guan, Wei Zhao, Wanqing Zhao, Xiaopeng Wang, Zhou Zhao, and Huan Sun. 2019. Answer Identification from Product Reviews for User Questions by Multi-Task Attentive Networks. In *AAAI*. 45–52.

[6] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *ACL*. 1657–1668.

[7] Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. 2019. Review-Driven Answer Generation for Product-Related Questions in E-Commerce. In *WSDM*. 411–419.

[8] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. 2019. Semi-supervised User Profiling with Heterogeneous Graph Attention Networks. In *IJCAI*. 2116–2122.

[9] Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2019. Joint Learning of Answer Selection and Answer Summary Generation in Community Question Answering. *CoRR* abs/1911.09801 (2019).

[10] Yang Deng, Ying Shen, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge as A Bridge: Improving Cross-domain Answer Selection with External Knowledge. In *COLING*. 3295–3305.

[11] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *SIGdial*. 37–49.

[12] Miao Fan, Chao Feng, Lin Guo, Mingming Sun, and Ping Li. 2019. Product-Aware Helpfulness Prediction of Online Reviews. In *WWW*. 2715–2721.

[13] 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*. 429–437.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[15] Kishaloy Halder, Min-Yen Kan, and Kazunari Sugiyama. 2019. Predicting Helpful Posts in Open-Ended Discussion Forums: A Neural Architecture. In *NAACL-HLT*. 3148–3157.

[16] Sanda M. Harabagiu and Andrew Hickl. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *ACL*.

[17] Shahar Harel, Sefi Albo, Eugene Agichtein, and Kira Radinsky. 2019. Learning Novelty-Aware Ranking of Answers to Complex Questions. In *WWW*. 2799–2805.

[18] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.

[19] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *SIGIR*. 228–235.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[22] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. Spam Review Detection with Graph Convolutional Networks. In *CIKM*. 2703–2711.

[23] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[24] 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 2019, San Francisco, CA, USA, May 13-17, 2019.* 1198–1209.

[25] Julian McAuley and Alex Yang. 2016. Addressing Complex and Subjective Product-Related Queries with Customer Reviews. In *WWW*. 625–635.

[26] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *SemEval@ACL*. 27–48.

[27] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*. 188–197.

[28] Adi Omari, David Carmel, Oleg Rokhlenko, and Idan Szpektor. 2016. Novelty based Ranking of Human Answers for Community Questions. In *SIGIR*. 215–224.

[29] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. In *IJCAI*. 1305–1311.

[30] Jinfeng Rao, Hua He, and Jimmy Lin. 2017. Experiments with Convolutional Neural Network Models for Answer Selection. In *SIGIR*. 1217–1220.

[31] 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*. 5373–5384.

[32] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[33] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*. 593–607.

[34] Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic Feature Engineering for Answer Selection and Extraction. In *EMNLP*. 458–467.

[35] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR*. 373–382.

[36] Chirag Shah and Jeffrey Pomerantz. 2010. Evaluating and predicting answer quality in community QA. In *SIGIR*. 411–418.

[37] Taihua Shao, Fei Cai, Honghui Chen, and Maarten de Rijke. 2019. Length-adaptive Neural Network for Answer Selection. In *SIGIR*. 869–872.

[38] Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. DivGraphPointer: A Graph Pointer Network for Extracting Diverse Keyphrases. In *SIGIR*. 755–764.

[39] Maggy Anastasia Suryanto, Ee-Peng Lim, Aixin Sun, and Roger H. L. Chiang. 2009. Quality-aware collaborative question answering: methods and evaluation. In *WSDM*. 142–151.

[40] Ming Tan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. In *ACL*.

[41] Yi Tay, Minh C. Phan, Anh Tuan Luu, and Siu Cheung Hui. 2017. Learning to Rank Question Answer Pairs with Holographic Dual LSTM Architecture. In *SIGIR*. 695–704.

[42] Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing Entailment for Multi-Hop Question Answering Tasks. In *NAACL-HLT*. 2948–2958.

[43] Kateryna Tymoshenko and Alessandro Moschitti. 2018. Cross-Pair Text Representations for Answer Sentence Selection. In *NAACL*. 2162–2173.

[44] Mengting Wan and Julian J. McAuley. 2016. Modeling Ambiguity, Subjectivity, and Diverging Viewpoints in Opinion Question Answering Systems. In *ICDM*. 489–498.

[45] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP-CoNLL*. 22–32.

[46] Shuohang Wang and Jing Jiang. 2017. A Compare-Aggregate Model for Matching Text Sequences. In *ICLR*.

[47] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. In *IJCAI*. 4144–4150.

[48] Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM*. 287–296.

[49] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and Effective Text Matching with Richer Alignment Features. In *ACL*. 4699–4709.

[50] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *EMNLP*. 2013–2018.

[51] Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce. In *WSDM*. 682–690.

[52] Jianxing Yu, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Answering Opinion Questions on Products by Exploiting Hierarchical Organization of Consumer Reviews. In *EMNLP-CoNLL*. 391–401.

[53] Qian Yu and Wai Lam. 2018. Review-Aware Answer Prediction for Product-Related Questions Incorporating Aspects. In *WSDM*. 691–699.

[54] Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. 2020. Review-guided Helpful Answer Identification in E-Commerce. In *WWW*. 2620–2626.