Multi-Type Textual Reasoning for Product-Aware Answer Generation

Yue Feng, Zhaochun Ren, Weijie Zhao, Mingming Sun, Ping Li
Cognitive Computing Lab
Baidu Research
No. 10 Xibeiwang East Road, Beijing 10193, China
10900 NE 8th St, Bellevue, Washington 98004, USA
{leah.yue.feng, zhc.ren, zhaoweijie12, sunmingming, pingli98}@gmail.com

ABSTRACT

By reading reviews and product attributes, e-commerce question-answering task aims to automatically generate natural-sounding answers for product-related questions. Existing methods, however, typically assume that each review and each product attribute are semantically independent, ignoring the relation among all these multi-type texts. In this paper, we propose a review-attribute heterogeneous graph neural network (abbreviated as RAHGNN) to model the logical relation of all multi-type text. RAHGNN consists of four components: a review-attribute heterogeneous graph constructor, a question-aware input encoder, a heterogeneous graph relation analyzer, and a context-based answer decoder. Specifically, after constructing the heterogeneous graph with reviews and product attributes, we derive the initial representation of each review node and attribute node based on question attention network and key-value memory network respectively. RAHGNN analyzes the relation according to the subgraph structure and subgraph semantic meaning using node-level attention and semantic-level attention. Finally, the answer is generated by the recurrent neural network with the relation representation as context input. Extensive experimental results on a large-scale real-world e-commerce dataset not only show the superior performance of RAHGNN over state-of-the-art baselines, but also demonstrate its potentially good interpretability for multi-type text relation in product-aware answer generation.

CCS CONCEPTS

• **Information systems** → **Question answering**.

KEYWORDS

Question answering, e-commerce, product-aware, reasoning.

ACM Reference Format:
Yue Feng, Zhaochun Ren, Weijie Zhao, Mingming Sun, Ping Li. 2021. Multi-Type Textual Reasoning for Product-Aware Answer Generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404835.3462899

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07...$15.00
https://doi.org/10.1145/3404835.3462899

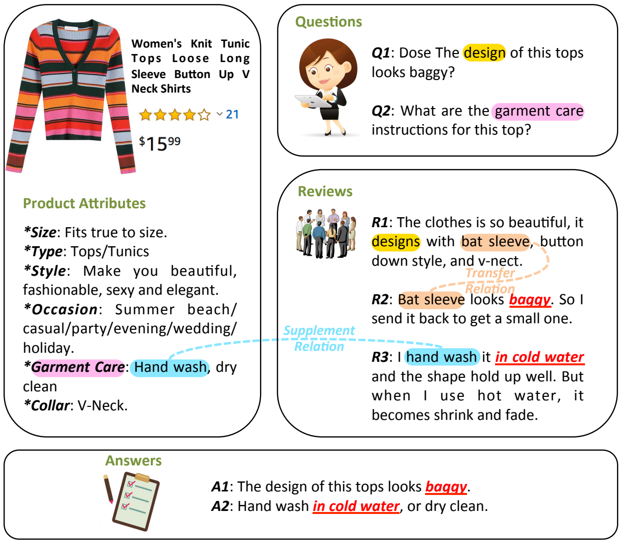

Women's Knit Tunic Tops Loose Long Sleeve Button Up V Neck Shirts
21
$15.99
Product Attributes
*Size: Fits true to size.
*Type: Tops/Tunics
*Style: Make you beautiful, fashionable, sexy and elegant.
*Occasion: Summer beach/casual/party/evening/wedding/holiday.
*Garment Care: Hand wash, dry clean
*Collar: V-Neck.
Questions
Q1: Dose The design of this tops looks baggy?
Q2: What are the garment care instructions for this top?
Reviews
R1: The clothes is so beautiful, it designs with bat sleeve, button down style, and v-nect.
Transfer Relation
R2: Bat sleeve looks baggy. So I send it back to get a small one.
Supplement Relation
R3: I hand wash it in cold water and the shape hold up well. But when I use hot water, it becomes shrink and fade.
Answers
A1: The design of this tops looks baggy.
A2: Hand wash in cold water, or dry clean.

Figure 1: Examples of the multi-type text relation for product-aware question answering.

1 INTRODUCTION

To increase the number of sales, most e-commerce portals provide a question-answering (QA) service to facilitate the customers' shopping procedure by answering their questions about products [13, 39]. With the QA system, the potential customers can consult the consumers who purchased the same product before [5, 20, 27] for additional information (e.g., the accuracy of clothes' size, using experience, quality) about the product. One major limitation of such a QA system is that most questions cannot be answered in time. Due to the lack of timely answers, potential customers have to read the product's reviews by themselves instead of finding the desired information. However, the information provided by reviews is usually overloaded and contradictory. To obtain their desired information, customers have to extract and reason the semantic units in reviews for proper answers [14, 15]. Such a complex information-acquiring procedure significantly increases the difficulty for users to obtain useful product-related information. A more complicated information-acquiring procedure can directly lead to a lower buying inclination of customers. Therefore, designing a more effective and efficient product-aware question-answering system becomes more and more important in the e-commerce area.

To achieve automatic question answering, most existing approaches concentrate on analyzing the product's reviews to produce proper answers [5, 7, 8, 11, 26, 50]. Among existing methods, some of them [23, 46–49] are based on information extraction from reviews, which directly use a relevant review or collect a review span to produce the answer to a specific question. However, since most questions cannot be answered only by using review spans, the generation-based models are recently proposed to obtain a more natural and informative answer. Based on the product's attributes and reviews, such generation-based algorithms [15, 33] usually directly produce proper answers from scratch.

One major limitation of existing approaches is that they usually analyze each review and the corresponding attribute of the product individually, i.e., they neglect the relationship between different reviews/attributes of the product. To answer specific questions in the wild, it is necessary to take the text information from different reviews and attributes into consideration. As shown in Figure 1, *Q1* asks *"Does The design of this tops looks baggy?"*. *R1* and *R2* do not answer this question directly. But they provide a common entity *"bat-like sleeve"*. If we transfer the information provided by *R1* and *R2* to answer *Q1* indirectly, it is easy to generate the answer that *"The design of this tops looks baggy"*. We demonstrate another example in *Q2*, which asks *"What are the garment care instructions for this tops?"*. It is hard to produce the answer to this question from scratch. However, if we refer to the product attribute Garment Care and *R3*, which says *"Hand wash, dry clean"* and *"hand wash it in cold water"*, respectively. We can easily obtain an accurate answer to *Q2* by combining the context of the review and the attribute together, which provides more details against the answer generated by only using the product attribute. The above examples show that we may generate more accurate and pleasing answers to complex questions by integrating, understanding, and reasoning over the information of reviews and product attributes in combination. Whereas, processing multi-source information is non-trivial since the product attributes and reviews usually are different types of text and come from completely different sources – product attributes are key-value pairs given by the sellers while reviews are raw text written by customers.

To sufficiently understand and reason the relation information and its inner logic on multi-type texts, we propose a review-attribute heterogeneous graph neural network (RAHGNN) for product-aware answer generation. RAHGNN differentiate existing methods in its ability to understand and reason the relation between multi-type texts. Specifically, in RAHGNN, nodes represent the entity of information (e.g., reviews and attributes), and edges represent the relation between entities. To generate the answer to a product-related question, we first employ an attention mechanism to model the interactions between the question and its reviews, which is later used as the review node's initial embedding. Besides, we introduce a key-value memory network to extract the relevant information between question and attribute as the initial embedding of the attribute node. Then, RAHGNN learns the logical relation between entities in heterogeneous graph by subgraph representation and subgraph integration. To capture the complex structure in each subgraph, we employ the node-level attention to discriminate the importance of meta-path based neighbors. To fuse the semantic information of all subgraphs, we introduce the semantic-level attention to learn the corresponding attention values of different subgraphs concerning a specific question. Through information propagation in the heterogeneous graph, logical relation information of entities exchanges and reasons sufficiently. Finally, inspired by the sequence-to-sequence architecture with the attention mechanism, we propose a recurrent neural network (RNN) based decoder, which uses the relation representation as the context vector to generate answers. Our experiments demonstrate the effectiveness and efficiency of RAHGNN in generating both accurate and informative answers to product-related questions in e-commerce. In summary, our contributions of this paper are three-fold:

- To our best knowledge, RAHGNN is the first algorithm which can understand/reason/inference relations between multi-type texts for generating complex product-aware answers.
- RAHGNN uncovers the intrinsic relations between a specific question and the corresponding multi-type texts (reviews and attributes), which shows its good interpretability.
- Extensive experiments on a large-scale real-world dataset show that RAHGNN outperforms state-of-the-art baselines in generating natural-sounding answers to product-related questions in e-commerce scenarios.

2 RELATED WORK

Product-aware question answering in e-commerce is an important task that has drawn much attention in recent years. Many works focus on aspect-based extraction and opinion mining from user reviews to answer the given question. Yu *et al.* [48] propose a new framework which can accurately identify aspects in the questions with the help of the hierarchy and retrieve the corresponding review fragments relevant to the aspects from the hierarchy as the answer. Yu *et al.* [49] propose to learn latent aspect-specific embeddings of reviews by aspect analytic model to predict the answer. McAuley *et al.* [23] propose a relevance function to determine which reviews contain relevant information, and a prediction function to vote on the correct answer aspect based on the relevant reviews. Recently, several ranking based models also have been developed for e-commerce question answering. Cui *et al.* [5] propose to get candidate sentences by searches the related queries and rank all candidate sentences with a regression based ranking framework. Yu *et al.* [47] propose an online system for question answering which the goal is to retrieval the nearest question in the knowledge base for a given customer question. With the emergence of neural networks, some generative product-aware question answering models, which generate an answer from scratch, have also been proposed. Gao *et al.* [14, 15] use the Wasserstein distance based adversarial learning method to learn to denoise the review text and product attribute to generate answers. Chen *et al.* [3] propose a review-driven answer generation framework which automatically generates answer in natural language based on a noise-tolerant solution. Zhang *et al.* [50] propose a conformal prediction based framework which can reject unreliable answers to improve concise and accurate for answering the product question.

However, above models generate answers utilize simple review analysis methods which deal with each review independently. These methods are unable to grasp sufficient relational and logical information among reviews. To the best of our knowledge, RAHGNN is the first to model the relation of unstructured reviews and structured product attributes to facilitate natural answer generation.

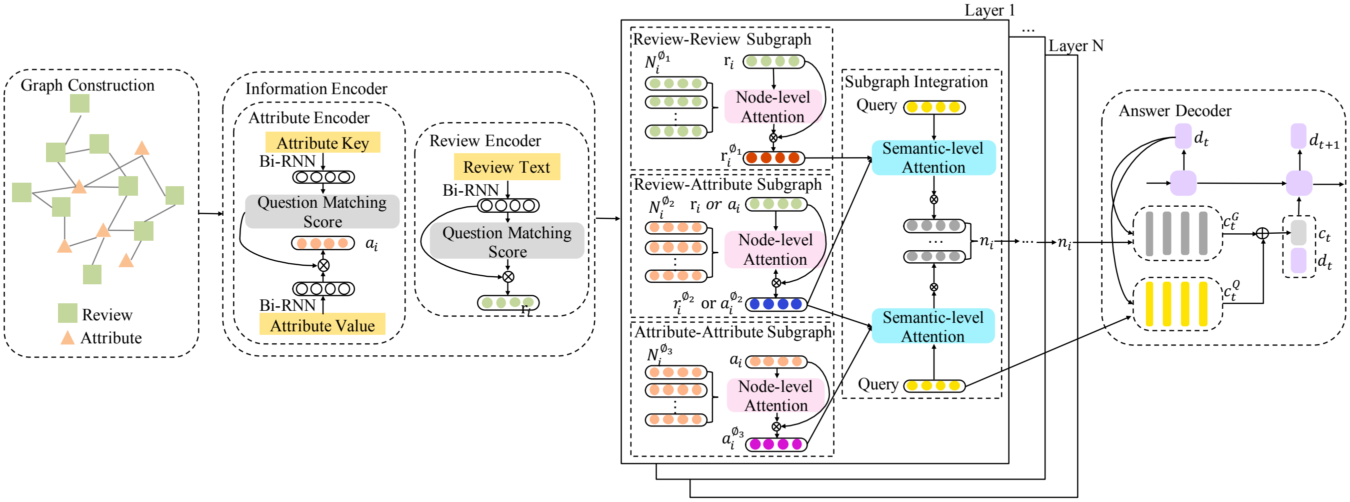


Figure 2: Overview of RAHGNN. We divide RAHGNN into four parts: 1) Graph constructor creates a heterogeneous graph from a set of reviews and product attributes. 2) Information encoder learns the embeddings of reviews and product attributes as nodes initial representation by using question matching attention. 3) Multi-type text reasoning module grasps logical relation of heterogeneous graph by subgraph structure and subgraph semantic integration. 4) Answer decoder generates answers according to the logical relation and question context.

Graph neural networks (GNNs) which aim to extend the deep neural network to deal with arbitrary graph-structured data are introduced in [30]. Kipf *et al.* [19] propose a graph convolutional network via a localized first-order approximation to encode both local graph structure and features of nodes. With the development of attention mechanisms, Petar *et al.* [38] propose to introduce the attention mechanism to learn the importance between nodes and its neighbors for graph based applications. Graph neural networks have been successfully applied into recommender systems [41]. Social-aware links have also been combined into graph neural networks to model user behaviors [45]. Recently, there are also some preliminary works of applying graph neural network for question answering. Song *et al.* [32] propose a graph-based method to better connect global evidence of passage for reading comprehension. Cao *et al.* [2] propose a bi-directional attention graph network to leverage relations between nodes in an entity graph to answer question. Besides, graph neural networks also have been used in recommendations for inference [9, 10, 44]. However, the above graph neural network cannot deal with various types of nodes and edges and can only be applied to the homogeneous graphs.

3 PRELIMINARIES

In this section, we formulate heterogeneous graph for reviews and attributes. And define the product-aware question-answering task.

3.1 Problem Formulation

In this paper, we propose a **heterogeneous graph** to represent information, such as reviews and product attributes. The structure of a heterogeneous graph is shown in Figure 2. For a heterogeneous graph $G = (V, E)$, V and E denote the set of nodes and the set of edges in the graph. The nodes V consist of the set of reviews X^r, and the set of product attributes X^a. The edges E have three types, Review-Review edges E_{rr} that reflect the relations between reviews, Review-Attribute edges E_{ra} that describe the interaction between reviews and product attributes, and Attribute-Attribute edges E_{aa} that express the relation between attributes.

In our e-commerce question answering heterogeneous graph, review nodes are provided by other consumers who purchased the same product before, and attribute nodes are key-value pairs to describe the product. The Review-Review edge describes common words appearing in both reviews. The Review-Attribute edge means that the review contains the attribute keywords. The Attribute-Attribute edge represents that the attributes contain the same words.

According to the types of edges, the heterogeneous graph can be divided into three subgraphs: Review-Review subgraph, Review-Attribute subgraph, and Attribute-Attribute subgraph. 1) **Review-Review subgraph**: The nodes in Review-Review subgraph are reviews in the heterogeneous graph. The edges in the subgraph are the relation between reviews. 2) **Review-Attribute subgraph**: The nodes in Review-Attribute subgraph are reviews and attributes in the heterogeneous graph. The edges in the subgraph are the interactions between reviews and attributes. 3) **Attribute-Attribute subgraph**: The nodes in Attribute-Attribute subgraph are attributes in the heterogeneous graph. The edges in the subgraph are the relation between attributes.

3.2 Product-aware Question Answering

When people purchase products offline, questions about diverse aspects of the product are often issued before they make the final decision. To cater users' need, e-commerce services assist users in posing product-specific questions to other consumers who have previously purchased the same product before. For e-commerce scenarios, The reviews of products are a fruitful resource to answer these product-aware questions, because opinions from many users can be well revealed through the reviews. Product attributes can also help provide answers, as they describe the characteristics of the product more detailed. Thus the product-aware question answering can be formulated as automatically generating accurate and informative answers in natural language for a product-aware question based on the corresponding reviews and product attributes.

For a product, we assume there is a question $X^q = \{x_1, x_2, ..., x_{T_q}\}$, T_r reviews $X^r = \{R_1, R_2, ..., R_{T_r}\}$, and T_a key-value pairs of product attributes $X^a = \{(A^k_1, A^v_1), (A^k_2, A^v_2), ..., (A^k_{T_a}, A^v_{T_a})\}$, where A^k is the name of i-th attribute and A^v is the attribute content. Given a question X^q, an answer generator reads the reviews X^r and attributes X^a, then generates an answer $\hat{Y} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_{T_y}}\}$. The goal is to generate an answer $\hat{Y}$ that is not only grammatically correct but also consistent with opinions in the reviews and product attributes. Essentially, the generator tries to optimize the parameters to maximize the probability $P(Y|X^q, X^r, X^a) = \prod_{t=1}^{T_y} P(y_t|X^q, X^r, X^a)$, where $Y = \{y_1, y_2, ..., y_{T_y}\}$ is the ground truth answer.

4 MODEL

RAHGNN is designed to grasp sufficient relational and logical information to reason answers. As depicted in Figure 2, RAHGNN is split into three components: an information encoder, a multi-type text reasoning component, and an answer decoder.

4.1 Information Encoder

In RAHGNN, the input information includes reviews and product attributes. We encode the review text into vector representations by matching the relevance of the given question. And for the attributes, we store the product attributes into a key-value memory network according to the correlation score between the key and the question.

Review Encoder: Given a question $X^q = \{x_1, x_2, ..., x_{T_q}\}$, and a review $R_i = \{r_1, r_2, ..., r_{T^i_r}\}$. Following [4], we first employ bi-directional recurrent neural network (Bi-RNN) to model the temporal interactions between words in question and review, respectively:

$$h^q_t = \text{Bi-RNN}_q(x_t, h^q_{t-1}), \qquad h^r_t = \text{Bi-RNN}_r(r_t, h^r_{t-1}),$$

where x_t is the embedding of t-th word in question X^q, r_t is the embedding of t-th word in review R_i. And the h^q_t, h^r_t denote the hidden state of t-th step in Bi-RNN. We use the final hidden state $h^q_{T_q}$ of Bi-RNN$_q$ to represent the question sentence X^q. Following [31, 43], we here choose the LSTM as the cell of Bi-RNN.

In order to demote the question-oriented salient part of the review, we proposed an attention-based method to consider the focus point of question into the review representation. Specifically, we add an additional gate to learn the relevance between question X^q and review R_i via a soft-alignment.

$$s^n_j = v^\top \tanh(W_q h^q_n + W_r h^r_j),$$
$$s_j = \max(s^1_j, s^2_j, ..., s^{T_q}_j),$$
$$\alpha_j = \exp(s_j) / \sum_{t=1}^{T^i_r} \exp(s_t),$$

where W_q, W_r, v are all trainable parameters, T_q is the text length of question X^q. α_j refers to the importance score of the j-th word in the review text given the question X^q.

Thereafter, we adopt an attention-pooling operation on each review hidden state h^r_t to produce question-aware review representation. So we get the semantic representation r_i of review R_i as

$$r_i = \sum_{t=1}^{T^i_r} \alpha_t h^r_t,$$

where T^i_r refers to the text length of review R_i. We use r_i as the initial node representation of review node R_i in the review-attribute heterogeneous graph.

Attribute Encoder: The attributes of a product are key-value pairs which can be seen as structured knowledge data in our task. As key-value memory network (KVMN) has shown effective on structured data utilization tasks [18, 24, 36], in our framework, we employ KVMN to analyze product attributes for answer generation. Correspondingly, we store the representation of each attribute's key and value in the KVMN. The read operation in our KVMN is divided into two steps: key matching and value combination.

Specifically, given attribute key $A^k_i = \{a^k_1, a^k_2, ..., a^k_{T^i_k}\}$ and attribute value $A^v_i = \{a^v_1, a^v_2, ..., a^v_{T^i_v}\}$, following [4], we use Bi-RNN to extract features of key and value to get semantic representation:

$$h^k_t = \text{Bi-RNN}_k(a^k_t, h^k_{t-1}), \qquad h^v_t = \text{Bi-RNN}_v(a^v_t, h^v_{t-1}),$$

where a^k_t, and a^v_t is the embedding of t-th word in attribute key A^k_i and attribute value A^v_i, respectively. h^k_t, h^v_t is the hidden state of t-th step of Bi-RNN.

The goal of the key matching step is to calculate the relevance between each word of attribute key to the given question X^q. The matching score $P(a^k_j|X^q)$ for j-th word of key is calculated as

$$P(a^k_j|X^q) = \frac{\exp(h^q_{T_q} W_k h^k_j)}{\sum_{t=1}^{T^i_k} \exp(h^q_{T_q} W_k h^k_t)},$$

where W_k is a trainable key matching parameter to transform question representation and key representation into a same space, and T^i_k is the text length of attribute key A^k_i.

In the value combination step, we need to extract the related value information for each word of the key, firstly. Following [28], we use the bi-linear layer to obtain the relevant information representation of value for keyword a^k_j.

$$v_j = h^k_j W_v h^v_{T^i_v},$$

where W_v is the matching parameter. Then we use keyword matching score $P(a^k_j|X^q)$ to produce a weighted sum of all related information representation of value since a keyword with high matching score is more related to the question, so should take a larger proportion in overall attribute representation. The question-aware attribute representation a_i, for initializing the attribute node in the review-attribute heterogeneous graph, can be calculated as

$$a_i = \sum_{j=1}^{T^i_k} P(a^k_j|X^q) v_j.$$

4.2 Multi-type Text Reasoning

To comprehend the relational and logical information among reviews and product attributes, we propose an attention-based heterogeneous graph neural network to track the information propagation among different types of text. Our attention-based heterogeneous graph neural network is composed of three major steps: subgraph representation, subgraph integration, and information propagation.

Subgraph Representation: We initialize the review node R_i and product attribute node (A_i^k, A_i^v) by the review representation r_i and the product attribute representation a_i, respectively. Due to the heterogeneity of nodes, different types of nodes have different feature spaces. Following [16], for each type (ϕ) of nodes, we design the type-specific transformation matrix M_ϕ to project the features of different types of nodes into the same feature space.

$$g_i = M_\phi n_i',$$

where n_i' and g_i are respectively the original and projected representations of node i, and ϕ is the type of node i, which can be review or product attribute.

To inject the structural information in each subgraph, we leverage self-attention [37] to learn the weight among neighbors. Given a node i, and its neighbors $N_i^{\Phi_m}$ which are connected in the subgraph Φ_m, the node-level attention $\alpha_{i,j}^{\Phi_m}$ can be learned. The $\alpha_{i,j}^{\Phi_m}$ means how important the neighbor node j will be for the node i. The node-level attention can be calculated as follows,

$$\alpha_{i,j}^{\Phi_m} = \frac{\exp(g_j W_N g_i)}{\sum_{n \in N_i^{\Phi_m}} \exp(g_n W_N g_i)},$$

where W_N is relation parameter. Then, the subgraph-based neighbor embedding of node i can be aggregated by the neighbor's projected features with the corresponding coefficients as follows:

$$u_i^{\Phi_m} = \sum_{j \in N_i^{\Phi_m}} \alpha_{i,j}^{\Phi_m} g_j.$$

To capture the node and it's neighbor information, we concatenate the representations to obtain the subgraph-based node embedding:

$$n_i^{\Phi_m} = [g_i \times u_i^{\Phi_m}; g_i; u_i^{\Phi_m}],$$

where × denotes the element-wise multiplication, and $[\cdot;\cdot]$ denotes the concatenation of two vectors.

Subgraph Integration: Inspired by Wang et al. [42], each node in a heterogeneous graph contains multiple types of semantic information and each semantic information is revealed by subgraphs. To learn a more comprehensive node embedding, we proposed a semantic-level attention to automatically learn the importance of different subgraphs and aggregate them for the semantic fusion.

To learn the importance each subgraph, we first transform the subgraph-based node embedding through a nonlinear transformation. Then we measure the importance of subgraph as the similarity of transformed embedding with question X^q. The importance of subgraph Φ_m, denoted as $w_i^{\Phi_m}$, is shown as follows,

$$w_i^{\Phi_m} = h_{T_q}^q \tanh(W_s n_i^{\Phi_m} + b),$$

where W_s is the weight matrix, b is the bias vector, $h_{T_q}^q$ is the semantic representation of question X^q. All above parameters are shared for all subgraphs. After obtaining the importance of each subgraph, we normalize them via softmax function. The weight of subgraph Φ_m for node i, denoted as $\beta_{i,m}$ can be obtained as follows,

$$\beta_{i,m} = \exp(w_i^{\Phi_m}) / \sum_{j=1}^{M} \exp(w_i^{\Phi_j}),$$

which can be interpreted as the contribution of the subgraph for question-aware semantic representation. Obviously, the higher $\beta_{i,m}$, the more important subgraph Φ_m is. With the learned weights as coefficients, we can fuse these subgraph-base embeddings to produce the finial semantic embedding n_i as follows,

$$n_i = \sum_{m=1}^{M} \beta_{i,m} n_i^{\Phi_m},$$

where M is the number of subgraphs.

Semantic Information Propagation: To explore the higher-order connectivity information of reviews and product attributes, following [40], we stack T layers of subgraph representation and subgraph integration. Each layer k takes the semantic node embedding from the previous layers as input, and outputs the updated node semantic embedding after the current diffusion process finishes. The updated node semantic embeddings are sent to the $k+1$ layer for the next diffusion process.

4.3 Answer Decoder

Our proposed model RAHGNN generates an answer based on the question and the logical information extracted from the review-attribute heterogeneous graph. Following [22], we adopt an attention-based RNN answer decoder to generate answers. At each decoding step, a context vector summarizing the input question and the logical information is fed into the decoder.

We initialize the decoder state d_0 with the question and logical information. We concatenate the question and the sum of node semantic embedding and apply a linear transform to map these features into the same space. The t-th decoding step is shown as

$$d_0 = W_d[h_{T_q}^q; \sum_{j=1}^{T_r+T_a} n_j] + b_d, \qquad d_t = \text{Bi-RNN}_d(d_{t-1}, [c_{t-1}; y_{t-1}]),$$

where W_d, b_d are the trainable parameters, d_t is the hidden state of t-th decoding step, and c_{t-1} is the context vector. We concatenate the question context c_t^Q and the logical information context c_t^G with a balance gate γ which is determined by decoder state d_t to obtain the context vector c_t:

$$c_t = [\gamma c_t^G; (1-\gamma) c_t^Q], \qquad \gamma = \sigma(W_c d_t + b_c).$$

Similar with the seq2seq with attention mechanism [1], we use the hidden state of previous step d_{t-1} to attend the question hidden states to get the question context c_t^Q:

$$q_{i,t}' = z_q^\top \tanh(W_s^q h_i^q + W_d^q d_t),$$

$$q_{i,t} = \exp(q_{i,t}') / \sum_{j=1}^{T_q} \exp(q_{j,t}'),$$

$$c_t^Q = \sum_{i=1}^{T_q} q_{i,t} h_i^q,$$

where W_s^q, W_d^q, and z_q are trainable parameters. The algorithm of attending node semantic embeddings is same as attending question hidden states, we produce logical information context c_t^G as follows,

$$e'_{i,t} = z_g^\intercal \tanh(W_s^g n_i + W_d^g d_t).$$

$$e_{i,t} = \exp(e'_{i,t}) / \sum_{j=1}^{T_r+T_a} \exp(e'_{j,t}).$$

$$c_t^G = \sum_{i=1}^{T_r+T_a} e_{i,t} n_i,$$

where W_s^g, W_d^g, and z_g are trainable parameters.

The context vector c_t is denoted as a representation of semantic meaning of question and logical information of reviews and product attributes. We concatenated with the decoder state d_t with the context vector c_t and then fed into a linear transformation layer to produce the generated word distribution P_v over the vocabulary:

$$o_t = W_o[d_t; c_t] + b_o, \qquad P_v = \text{softmax}(W_v o_t + b_v),$$

where W_o, W_v, b_o, b_v are transformation parameters. At the t decoding step, the loss is the negative log likelihood of the target word y_t:

$$l = -\frac{1}{T_y} \sum_{t=1}^{T_y} \log P_v(y_t).$$

5 EXPERIMENTS

To evaluate our proposed method in reasoning on reviews and product attributes, following [15], we conduct our experiments on the large-scale datasets from JD (https://github.com/gsh199449/productqa), which is one of the largest e-commerce websites in China. In the collected data, each question-answer pair is associated with the reviews and attributes of the corresponding product. Most questions in the dataset are about personal user experience and product characteristics. We use BM25 to evaluate the relevance between reviews and questions, and remove all questions without any related reviews. In our experiments, if the related review number is more than 50 for one question, we need to filter out the top related 50 reviews to join the training process. We split the whole dataset into training set, validation set, and testing set. In total, JD dataset contains 469,953 products and 38 product categories. Detailed statistics of the datasets are given in Table 1.

Table 1: Statistics of JD question answering datasets. #(q,a): the number of question-answer pairs; Len(q): the average length of question; Len(a): The average length of answer; Avg(|attr|): the average number of attributes for a question; Avg(|r|): the average number of reviews for a question.

Subset	#(q,a)	Len(q)	Len(a)	Avg(|attr|)	Avg(|r|)
Train	460000	13.82	11.22	15.03	2364.50
Validation	4955	13.94	11.53	15.04	2721.76
Test	5000	13.83	11.62	15.22	1922.46

5.1 Baselines

We will evaluate our models against popular and state-of-the-art review driven product question answering baselines and several alternative techniques, including:

S2SA [34]: The Seq2Seq model with attention mechanism has been proposed for language generation task. We use seq2seq framework as baseline method. No product-related review information is provided, the input sequence is question and ground truth output sequence is the answer.

S2SAR [34]: A simple method which can incorporate the review information when generating the answer. Different from the S2SA, an RNN is used to read all the reviews and concatenate the final state of this RNN with encoder final state as the initial state of decoder RNN.

SNet [35]: A two-stage state-of-the-art model which extracts some text spans and synthesis the answer from those spans. Our dataset does not have text span label ground truth for training the evidence extraction module, so we only use the passage ranking loss, the ground truth rankings are produced by BM25 score. We use the predicted extraction probability to do weighted sum of the original review word embeddings, and use this representation to feed into the answer generation module.

IMCP [50]: A conformal prediction based framework, which rejects unreliable answers so that the returned results are more concise and accurate at answering the product question.

QS [17]: This is a model for generating summaries of documents with respect to a query, which is a sequence-to-sequence model with attention and a pointer mechanism. Both encoders, as well as the decoder, use RNNs with LSTM. We use the product reviews as original passages and the answer as a summary to train the model.

RAGE [3]: A noise-tolerant solution based on convolutional neural networks to generate natural answers for product-related questions. The relevant review snippets are extracted from the reviews of the corresponding product. The attention and gate mechanism are utilized to highlight the relevant part presented in the auxiliary review snippets against the question context.

PAAG [15]: An end-to-end learning method to extract the fact that is helpful for answering questions from reviews and attributes. Attention based review reader and attribute reader are combined with the Wasserstein distance based adversarial learning method to learn to denoise the review text.

5.2 Evaluation Metrics

Following previous work, we employ BLEU [19] to measure the quality of generated sentences by computing overlapping lexical units (e.g., unigram, bigram) with the reference sentence. A BLEU score can range from 0 to 1, where higher scores indicate closer matches to the reference answer, and where a score of 1 is assigned to a generated answer which exactly matches the reference answer.

To examine whether the methods can generate diverse answers, we also evaluate generated responses by calculating the number of distinct n-grams [21].

We also consider three embedding-based metrics to compute the semantic relevance between the generated and reference answer. An answer representation can be obtained by averaging the embeddings of all the words in that answer, of which the cosine similarity

Table 2: BLEU, Distinct, and embedding scores comparison between baselines on JD question answering dataset. Boldface scores indicate best results, and significant improvements over the best baseline are marked with $*$ (t-test, $p < 0.05$).

Model	BLEU	BLEU2	BLEU4	Distinct-2	Distinct-4	Average	Greedy	Extrema
S2SA	1.6186	3.1437	0.1706	0.0847	0.3014	0.410013	98.653415	0.269461
S2SAR	1.7549	3.2156	0.2142	0.0928	0.3236	0.419979	99.742679	0.278666
SNet	0.9550	2.5374	0.0597	0.0633	0.2572	0.397162	95.791356	0.277781
IMCP	1.5632	2.8890	0.1572	0.0938	0.3092	0.378439	94.834892	0.264771
QS	1.6848	2.9508	0.2119	0.1092	0.3273	0.400291	93.255031	0.252164
RAGE	1.9934	3.4034	0.2518	0.1329	0.4218	0.422442	99.987672	0.303666
PAAG	2.0189	3.5711	0.2787	0.1129	0.3495	0.424218	103.912364	0.288321
RAHGNN	**3.1600***	**6.3847***	**0.6831***	**0.1743***	**0.5388***	**0.506788***	**131.233625***	**0.395748***

gives the Average metric [25]. Alternatively, greedy metric [29] is to greedily match words in two given answers based on the cosine similarities of their embeddings. In addition, one can also achieve an answer representation by taking the largest extreme values among the embedding vectors of all the words it has, before computing the cosine similarities between answer vectors, which yields the Extreme metric [12].

5.3 Implementation Details

For the proposed RAHGNN, we randomly initialize parameters. The model is trained with a batch size of 32 and a dimension size of 256 for word, POS and position embeddings. The randomly sampled validation set is used for early stop and parameter selection. Specifically, we limit the max length for reviews to 128 and the max length for product attributes to 10. For RNN based encoder and decoder models, the hidden dimension size is set to be 512. If two texts have the common word more than 3, there will have an edge between them to transfer the information. The propagation layer of the heterogeneous graph is 3. To produce better answers, we use beam search with beam size 4. Adagrad [6] with learning rate 0.1 is used to optimize the parameters.

5.4 Experimental Results

To demonstrate the effectiveness of RAHGNN, we examine the overall performance in terms of the automatic evaluation metrics. Table 2 lists performances of RAHGNN and all baselines in terms of BLEU, Distinct and embedding metrics on the JD dataset. In these experimental results, we see that RAHGNN outperforms all the baselines significantly. Specifically, RAHGNN achieves a %56.52, %78.78, %145.10, %31.15, and %27.73 increment, compared with the state-of-the-art model PAAG in terms of BLEU, BLEU2, BLEU4, Distinct2, Distinct4, respectively. For embedding metrics, RAHGNN outperforms PAAG by %19.46, %26.29, %30.32 in terms of Average, Greedy, and Extrema metrics, respectively. We conducted significant testing (t-test) on the improvements of our approaches over the best baseline. The results indicate that all the improvements are significant (p-value < 0.05 and denoted with '*' in Table 2). The results conclude that RAHGNN can perform not only better in word coverage, but also better in semantic accuracy and diversity.

From Table 2, we also find that PAAG and RAHGNN, which combine structured product attribute data with review information, outperform RAGE and IMCP. RAGE and IMCP are also a review-driven model to generate answers for product-related questions, which neglects the attribute information. This observation demonstrates the effectiveness of incorporating structured product attribute data in answer generation. Besides, we still find a noticeable gap between PAAG and RAHGNN. This result demonstrates that RAHGNN makes better use of relational and logical information of review and attribute than the simple method PAAG. As our task definition has some similarities in some way with reading comprehension and query-based text summarization, we also employ SNet and QS to tackle this task, which see the reviews as original passages. Since SNet and QS are not defined to tackle QA task in e-commerce scenario, they cannot fully utilize the interactions between question, reviews, and product attributes. These methods also lack reasoning component to extract logical information from the structured knowledge information and unstructured text information. So RAHGNN outperforms these models significantly. From the experiment results, we also find that S2SA and S2SAR, which only utilizes question text as the input of encoder, outperforms SNet, which demonstrates that the question information is essential for answer generation, so taking attention for question at every decoding step can help to generate better answers.

6 ANALYSIS AND DISCUSSION

6.1 Ablation Study

We conduct ablation tests on the usage of the attention-based heterogeneous graph neural network to verify the effectiveness of reasoning. The ablation models are shown as follows,

RAG: It is a variant of RAHGNN, but only uses the semantic representation of reviews and product attributes as the final node representation of the heterogeneous graph. That is, RAG does not have the multi-type text reasoning component.

RASG: It is a variant of RAHGNN, but only utilizes the concatenation of subgraph node representations as the final node representation of the heterogeneous graph. RASG contains the subgraph representation component.

RASIG: It is a variant of RAHGNN, but only uses the integration representation of subgraphs as the final node representation of the heterogeneous graph. RASIG contains the subgraph representation and subgraph integration components.

RAHGNN: Our proposed model, which contains subgraph representation, subgraph integration, and semantic information propagation components.

Table 3: Comparison results of BLEU, Distinct, and embedding scores for different ablation models on JD datasets. Boldface scores indicate best results.

Model	BLEU	BLEU2	BLEU4	Distinct-2	Distinct-4	Average	Greedy	Extrema
RAG	1.7853	3.1529	0.2322	0.1176	0.3862	0.413239	95.842932	0.278387
RASG	2.5738	4.8973	0.4262	0.1437	0.4525	0.453896	113.453525	0.316377
RASIG	2.8734	5.4289	0.5688	0.1563	0.4982	0.472446	125.642522	0.357338
RAHGNN	**3.1600**	**6.3847**	**0.6831**	**0.1743**	**0.5388**	**0.506788**	**131.233625**	**0.395748**

Table 4: BLEU, Distinct, and Embedding scores comparison between different multi-type text integration variants. Boldface scores indicate best results.

Model	BLEU	BLEU2	BLEU4	Distinct-2	Distinct-4	Average	Greedy	Extrema
RAHGNN-R	2.4362	4.5238	0.3884	0.1388	0.4132	0.428389	108.324893	0.283754
RAHGNN-A	0.2738	0.5288	0.0204	0.0348	0.1237	0.178938	25.429847	0.098736
RAHGNN-R&A	2.7098	5.1172	0.5368	0.1429	0.4735	0.463760	120.836675	0.334681
RAHGNN	**3.1600**	**6.3847**	**0.6831**	**0.1743**	**0.5388**	**0.506788**	**131.233625**	**0.395748**

We evaluate RAHGNN and these variants on the JD dataset in terms of BLEU, Distinct, and embedding metrics. As shown in Table 3, there is a significant increase from RAG to RAHGNN, which verifies the effectiveness of the attention-based heterogeneous graph neural network. Comparing with RAG and RASG, we find that the relational information of reviews and attributes improve the performance significantly. The performance of RASG and RASIG reflects that the relational information fusion based on the relevance of question achieves better performance. A slight increment from RASIG to RAHGNN demonstrates that modeling high-order connectivity helps relational information transfer on the graph.

6.2 Effect of Information Relations

To verify the effectiveness of information relation for product-aware question answering, we vary the max relation numbers of each node and the propagation layer depth. In particular, we explore the layer numbers in the range of {1, 2, 3, 4} and the max relation numbers per node in {10, 100, 1000, 2000, Unlimited} for JD QA dataset. If the relation number of a node exceeds the max requirement, top relations, which have more common words, will be used. Figure 3 illustrates the results in terms of BLEU in different settings.

We find increasing the depth of RAHGNN substantially enhances the generation accuracy. Clearly, 2 layers and 3 layers achieve consistent improvement over 1 layer. When further stacking a propagation layer on the top, we find that 4 layer leads to overfitting on JD dataset. This might be caused by applying a too deep architecture that might introduce noises to the representation learning. These results verify that conducting suitable propagation layers, which help to capture the high-order connectivity relations, can greatly facilitate the answer performance. In addition, the unlimited relation number of each node consistently superior to all other numbers. It illustrates interactions of reviews or attributes are significant to generate high-quality logical representations, since the high-density relation strengthens the expressiveness of graph. It hence also verifies the effectiveness of the information relation.

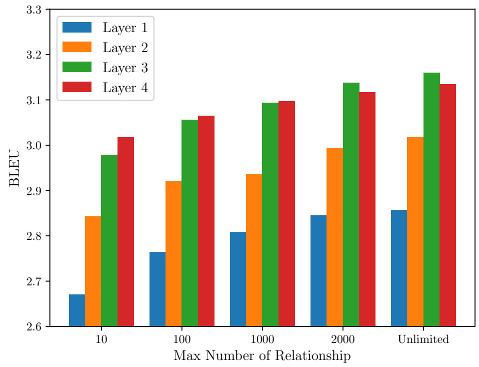


Figure 3: BLEU performance comparison of RAHGNN over propagation layer numbers and the max relation numbers on JD question answering dataset.

6.3 Performance of Multi-type Text Integration

We now check whether RAHGNN can effectively integrate multi-type text and its effects on relation reasoning, via experiments on three variants of RAHGNN. The differences are which type of text is used and whether integrate multi-type text.

RAHGNN-R: It only uses review texts as input. It contains the review encoder, review-review subgraph, and answer decoder.

RAHGNN-A: Similar to RAHGNN-R, this model only uses attribute texts as input. It contains the attribute encoder, attribute-attribute subgraph, and answer decoder.

RAHGNN-R&A: It uses review texts and attribute texts without integration. It means that this model contains the review encoder, the attribute encoder, review-review subgraph, attribute-attribute subgraph, and answer decoder.

RAHGNN: Our proposed model, which contains review encoder, attribute encoder, all three subgraphs, subgraph integration, and answer decoder.

Table 4 displays the performance of all different variants. To examine whether the RAHGNN method can effectively integrate multi-type texts to reason, we compare to RAHGNN-R&A which doesn't use review-attribute subgraph and subgraph integration. The performance indicates that RAHGNN can effectively integrate multi-type information and benefit tremendously by conducting relation reasoning on multi-type text. Compared to RAHGNN-R and RAHGNN-A, the performance of RAHGNN-A drops most dramatically in terms of BLEU score. This observation suggests that review text understanding is very helpful in product QA, and our model successfully learns how to utilize review text.

6.4 Case Study

Table 5 shows examples in our experiments which needs multiple pieces of information to generate the right answer.

Table 5: Case study of the product related question that requires integrating multiple information to answer.

(a) Example 1.

Question:
Can this computer **change the system** to **WIN7**?

Generated Answer:
Yes. You can change the WIN10 to WIN7.

Review or Product Attribute:
(1) The boot is fast and light, but I am not used to the windows 10 system.
(2) We can **install some other operating systems** by ourselves.
(3) I want to **install win7 and win8**. I don't like win10.
(4) Category: Laptop. (5) **System: WIN10**.

Truth Answer:
Yes, you can change it by yourself. The initial system is WIN10.

(b) Example 2.

Question:
How hard are the **soles** and **leather**?

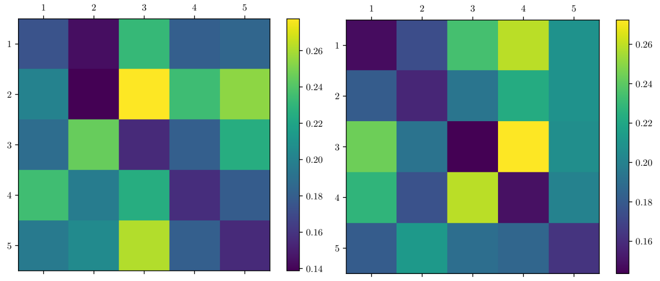
Generated Answer:
The sole is hard and the leather is soft.

Review or Product Attribute:
(1) The quality is good, but the **sole is hard**.
(2) Comfortable, soft and good!!!
(3) I felt the leather is a little hard when I first wore. But **it will be much better** after I wear it for a few days! Feeling very comfortable now!
(4) **Leather Feature: Soft**. (5) Heel Shape: Flat.

Truth Answer:
The sole is a bit hard and the leather is soft. So it's very comfortable to wear.

For the first example, the ground truth answer needs logical relation between the second review, the third review, and the fifth attribute. For the second case, the ground truth needs to summarize logical information among the first review, the third review, and the fourth attribute.

In order to verify the logical relation representation ability of RAHGNN, we plot the logical relation attention map from RAHGNN with one propagation layer in Figure 4. We utilize node-level attention to time semantic-level attention as the normalized logical relation attention for each node. From the experimental results, we observe that there is a very strong interaction between the high related information nodes. Concretely, it demonstrates that the multi-type text reasoning module can effectively capture the logical relation among multi-type information. Moreover, we also find that most information nodes tend to attend the high related node, which provides the most useful information in each case. And the attention weights for other information nodes are pretty low. It also indicates that RAHGNN has the ability to extract useful relation from multiple pieces of information.



(a) Example 1.

(b) Example 2.

Figure 4: Visualizations of logical relation attention map. The numbers on the left and up are the information IDs. The color corresponds to the importance of relation.

7 CONCLUSION

By reading reviews and product attributes, e-commerce question-answering task aims to automatically generate natural-sounding answers for product-related questions. One major limitation of existing approaches is that they usually analyze each review and the corresponding attribute of the product individually, i.e., they neglect the relationship between different reviews/attributes of the product. To answer specific questions in the wild, it is necessary to take the text information from different reviews and attributes into consideration.

In this paper, we propose a heterogeneous graph neural network (named RAHGNN) for generating natural-sounding answers to product-related questions in e-commerce. The proposed RAHGNN solves two main challenges in the question-answering area, i.e., 1) how to analyze the relations between diverse information, and 2) how to combine the unstructured information (reviews) and the structured information (product attributes) together to form useful knowledge. The extensive experiments on the JD benchmark dataset demonstrate that the proposed RAHGNN significantly outperforms the state-of-the-art baselines in generating informative, accurate, and natural-sounding answers to testing questions.

Next, we also discuss limitations of RAHGNN. We observe that the degree of relation reasoning is sensitive for the generation performance of RAHGNN. In addition to that, due to the number of reviews is huge, the efficiency of RAHGNN has yet to be improved. As a future work research direction, we will improve our algorithm to generate better answers by rational reasoning and compression relation graph. We also plan to include more product-related information, for example, the viewpoints of users, the product's categories, and the sentiments of reviews, into the considerations.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA.

[2] Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional Attention Entity Graph Convolutional Network for Multi-hop Reasoning Question Answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Minneapolis, MN, 357–362.

[3] Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. 2019. Review-Driven Answer Generation for Product-Related Questions in E-Commerce. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM)*. Melbourne, Australia, 411–419.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 1724–1734.

[5] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), System Demonstrations*. Vancouver, Canada, 97–102.

[6] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* 12 (2011), 2121–2159.

[7] Miao Fan, Chao Feng, Mingming Sun, and Ping Li. 2019. Reinforced Product Metadata Selection for Helpfulness Assessment of Customer Reviews. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 1675–1683.

[8] Miao Fan, Yue Feng, Mingming Sun, Ping Li, Haifeng Wang, and Jianmin Wang. 2018. Multi-Task Neural Learning Architecture for End-to-End Identification of Helpful Reviews. In *Proceedings of the IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Barcelona, Spain, 343–350.

[9] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering, Early Access* (2019).

[10] Fuli Feng, Weiran Huang, Xiangnan He, Xin Xin, Qifan Wang, and Tat-Seng Chua. 2021. Should Graph Convolution Trust Neighbors? A Simple Causal Inference Method. In *Proceedings of the 44rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*.

[11] Yue Feng, Miao Fan, Mingming Sun, and Ping Li. 2020. A Reinforced Semi-supervised Neural Network for Helpful Review Identification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*. Virtual Event, Ireland, 2021–2024.

[12] Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *NIPS, Modern Machine Learning and Natural Language Processing Workshop*.

[13] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and Challenges in Conversational Recommender Systems: A Survey. *arXiv preprint arXiv:2101.09459* (2021).

[14] Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2021. Meaningful Answer Generation of E-Commerce Question-Answering. *ACM Trans. Inf. Syst.* 39, 2 (2021), 18:1–18:26.

[15] Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-Aware Answer Generation in E-Commerce Question-Answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*.

[16] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*. Montréal, Canada, 2030–2041.

[17] Johan Hasselqvist, Niklas Helmertz, and Mikael Kågebäck. 2017. Query-based abstractive summarization using neural networks. *arXiv preprint arXiv:1712.06100* (2017).

[18] Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating Natural Answers by Incorporating Copying and Retrieving Mechanisms in Sequence-to-Sequence Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada, 199–208.

[19] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France.

[20] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, and Wei Chu. 2017. *AliMe Assist* : An Intelligent Assistant for Creating an Innovative E-commerce Experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*. Singapore, 2495–2498.

[21] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. San Diego CA, 110–119.

[22] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal, 1412–1421.

[23] Julian J. McAuley and Alex Yang. 2016. Addressing Complex and Subjective Product-Related Queries with Customer Reviews. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*. Montreal, Canada, 625–635.

[24] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Austin, TX, 1400–1409.

[25] Jeff Mitchell and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*. Columbus, OH, 236–244.

[26] Samaneh Moghaddam and Martin Ester. 2011. AQA: Aspect-based Opinion Question Answering. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)*. Vancouver, Canada, 89–96.

[27] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada, 498–503.

[28] Laura Rimell, Amandla Mabona, Luana Bulat, and Douwe Kiela. 2017. Learning to Negate Adjectives with Bilinear Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Valencia, Spain, 71–78.

[29] Vasile Rus and Mihai C. Lintean. 2012. An Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics. In *Proceedings of the 11th International Conference on Intelligent Tutoring Systems (ITS)*. Chania, Crete, Greece, 675–676.

[30] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Trans. Neural Networks* 20, 1 (2009), 61–80.

[31] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France.

[32] Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv preprint arXiv:1809.02040* (2018).

[33] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *Proceedings of the Web Conference (WWW)*. Taipei, 837–847.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*. Montreal, Canada, 3104–3112.

[35] Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-Net: From Answer Extraction to Answer Synthesis for Machine Reading Comprehension. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. New Orleans, LA, 5940–5947.

[36] Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to Remember Translation History with a Continuous Cache. *Trans. Assoc. Comput. Linguistics* 6 (2018), 407–420.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 5998–6008.

[38] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. (2018).

[39] Meng Wang, Weijie Fu, Xiangnan He, Shijie Hao, and Xindong Wu. 2020. A Survey on Large-scale Machine Learning. *IEEE Transactions on Knowledge and Data Engineering, Early Access* (2020).

[40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *International Conference on Research on Development in Information Retrieval (SIGIR)* (2019).

[41] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In *The Web Conference*.

[42] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *Proceedings of the World Wide Web Conference (WWW)*. San Francisco, CA, 2022–2032.

[43] Zhen Wang, Jiachen Liu, Xinyan Xiao, Yajuan Lyu, and Tian Wu. 2018. Joint Training of Candidate Extraction and Answer Selection for Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Melbourne, Australia, 1715–1724.

[44] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*.

[45] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the Evolution of Users' Preferences and Social Links in Social Networking Services. *IEEE Trans. Knowl. Data Eng.* 29, 6 (2017), 1240–1253.

[46] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Vancouver, Canada.

[47] 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 *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*. Marina Del Rey, CA, 682–690.

[48] Jianxing Yu, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Answering Opinion Questions on Products by Exploiting Hierarchical Organization of Consumer Reviews. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Jeju Island, Korea, 391–401.

[49] Qian Yu and Wai Lam. 2018. Review-Aware Answer Prediction for Product-Related Questions Incorporating Aspects. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*. Marina Del Rey, CA, 691–699.

[50] Shiwei Zhang, Xiuzhen Zhang, Jey Han Lau, Jeffrey Chan, and Cécile Paris. 2020. Less Is More: Rejecting Unreliable Reviews for Product Question Answering. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), Part III*. Ghent, Belgium, 567–583.