Question-Driven Purchasing Propensity Analysis for Recommendation

Long Chen,[1] Ziyu Guan,[2*] Qibin Xu,[3] Qiong Zhang,[4] Huan Sun,[5] Guangyue Lu,[1] Deng Cai[3]
[1]Xi'an University of Posts and Telecommunications, Xi'an, China
[2]Northwest University, Xi'an, China
[3]Zhejiang University, Hang Zhou, China
[4]Alibaba Group, Hang Zhou, China
[5]Ohio State University, Columbus, USA
[1]{chenlong, gylu}@xupt.edu.cn, [2]ziyuguan@nwu.edu.cn
[3]21751012@zju.edu.cn, dengcai78@qq.com [4]qz.zhang@alibaba-inc.com [5]sun.397@osu.edu

Abstract

Merchants of e-commerce Websites expect recommender systems to entice more consumption which is highly correlated with the customers' purchasing propensity. However, most existing recommender systems focus on customers' general preference rather than purchasing propensity often governed by instant demands which we deem to be well conveyed by the questions asked by customers. A typical recommendation scenario is: Bob wants to buy a cell phone which can play the game PUBG. He is interested in HUAWEI P20 and asks "can PUBG run smoothly on this phone?" under it. Then our system will be triggered to recommend the most eligible cell phones to him. Intuitively, diverse user questions could probably be addressed in reviews written by other users who have similar concerns. To address this recommendation problem, we propose a novel Question-Driven Attentive Neural Network (QDANN) to assess the instant demands of questioners and the eligibility of products based on user generated reviews, and do recommendation accordingly. Without supervision, QDANN can well exploit reviews to achieve this goal. The attention mechanisms can be used to provide explanations for recommendations. We evaluate QDANN in three domains of Taobao. The results show the efficacy of our method and its superiority over baseline methods.

Introduction

Online shopping has become our daily routine. Merchants of e-commerce Websites expect recommender systems to entice more consumption which is highly correlated with the customers' purchasing propensity. Purchasing propensity is mainly governed by a customer's "local" demands when looking for a specific kind of products. Such instant demands are not explicitly reflected by user behaviors exploited by traditional recommender systems, such as purchases, clicks and review writing. Moreover, most previous recommendation algorithms take the paradigm of summarizing users' preference from the above behaviors and using this "global" preference to generate recommendations (Wan and McAuley 2018). Although there are some studies trying to capture the dynamic preference by temporal models (Liu 2015), they still cannot well detect purchasing propensity, not to mention leveraging it for recommendation.

Today's e-commerce Websites offer an online Q&A system where users can freely ask questions about a product and receive answers from the merchant or users who bought that product. The questions are a good indicator of users' local demands. For instance, a user wants to buy a *cellphone* (a specific product domain) which can run *the PUBG* smoothly (an explicit instant demand), and he/she asks this question in the Q&A system. Providing recommendations based on this local demand from the cell phone domain is usually more effective for stimulating consumption than relying on general preference of the user. In this work, we propose and study this question-driven recommendation problem. We believe we are the first to explore this problem.

The basic idea of generating question-driven recommendations is that we evaluate purchasing propensity based on the eligibility of products for the user's local demands and generate recommendation accordingly. Then the critical question is, how we can assess the eligibility of products given a user and a question. Since users can ask a variety of questions, heuristic methods based on enumerating product features cannot well address this problem. Recently, reviews have become a focus for recommender systems since they not only reflect user preference but also show pros/cons of items (Zheng, Noroozi, and Yu 2017; Tay, Tuan, and Hui 2018). Intuitively, diverse user questions could probably be addressed in reviews written by other users who have similar concerns (Chen et al. 2019). We therefore resort to user reviews to address this question-driven recommendation problem.

In this work, we propose a Question-Driven Attentive Neural Network (QDANN) to evaluate purchasing propensity based on which personalized recommendations can be generated. To assess the eligibility of a product given a user question, QDANN analyzes reviews of the product to find supporting evidences. Relevance and sentiment orientation are the two important ingredients for this analysis. For example, in Figure 1 the sentence "It runs the PUBG smoothly." is a proper supporting evidence for the question "can the PUBG run smoothly on it?". This sentence contains both a relevance evidence ("runs the PUBG") and a positive sen-

*Corresponding author

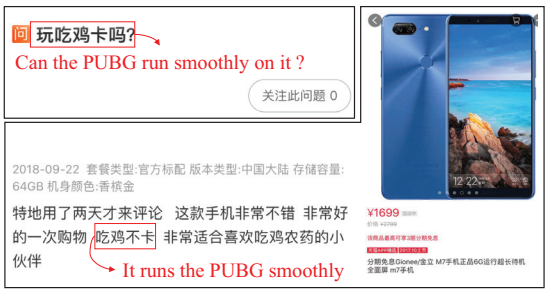

Figure 1: A relevant item review for a given question.

timent evidence ("smoothly"). To assess relevance and sentiment of reviews, we design two kinds of subnets, namely, Relevance Net (R-Net) and Sentiment Net (S-Net), respectively. R-Net employs the co-attention mechanism (dos Santos et al. 2016) to capture the relevance matching patterns between questions and reviews and is pre-trained by aspect relevance prior knowledge. S-Net uses a two-level attention mechanism to capture the sentiment evidences of reviews related to relevance matching and is pre-trained by user ratings as weak-supervision signals (Guan et al. 2016). The outputs of R-Net and S-Net are then combined through nonlinear transformation to predict purchase. In addition, we also take the target user's historical reviews to assess the eligibility of a candidate product for the user in a global sense. This complements the eligibility evaluation based on local demands (the question) and accounts for other factors leading to a purchase. The whole model is trained with user purchase information and no manual labeling work is needed.

Our work has three major contributions: 1) we propose question-driven recommendation, a novel recommendation problem which could be more effective to stimulate consumption in e-commerce platforms than previous recommendation paradigms. 2) We develop a neural model, QDANN, for this problem. QDANN does not need supervision and can provide explanations for recommendation via the attention mechanisms. 3) We empirically compare QDANN with state-of-art recommendation algorithms on a real-world dataset collected from three domains of Taobao, one of the largest e-commerce Website in China. The results show the efficacy of our method and its superiority over baseline methods.

Related Work

Review-based Recommendation System. Recent popular deep learning-based recommendation methods have taken customer reviews into consideration. DeepCoNN (Zheng, Noroozi, and Yu 2017) used dual CNNs with factorization machine (Rendle 2012) to capture the latent representations of users and items from the text of all reviews related to them, respectively. Later, Catherine and Cohen (Catherine and Cohen 2017) found that much of the predictive power of DeepCoNN came from the review written by the target user for the target item which is not available at test time. They proposed transformational neural networks extending DeepCoNN by an additional hidden layer which represented an approximation of the review corresponding to the target (user, item) pair. (Lu, Dong, and Smyth 2018) presented a deep learning recommendation model which co-learned user and item information from ratings and customer reviews by jointly optimizing matrix factorization and an attention-based GRU network. It achieved explainable recommendations through the attention mechanism. (Tay, Tuan, and Hui 2018) designed a variant of DeepCoNN with a Gumbel-Max (Maddison, Tarlow, and Minka 2014) co-attention module to extract review pointers that could indicate the important matching relations between review pairs. However, no previous work has considered exploiting reviews for the proposed question-driven recommendation problem.

Question Answering by Reviews. Our work is also related to some question answering works that leverage reviews. McAuley and Yang (McAuley and Yang 2016) proposed a model called MOQA to predict the answer for a question from candidate answers by using review sentences as supporting "experts". Based on this work, Yu and Lam (Yu and Lam 2018) proposed a framework including an aspect analytics model and a predictive answer model learned jointly from existing questions, answers, and reviews to predict the answers for yes-no questions. Chen et al. (Chen et al. 2019) proposed a multi-task attentive network for plausible answer identification from reviews. Our work is intrinsically different from the above works from two aspects: (1) the goal of those works is to automatically answer user submitted questions, while our goal is providing product recommendations to questioners via purchasing propensity analysis; (2) they only consider relevance (correctness for the yes-no case) of candidate answers to the concerned question, while we need to further consider another important factor, sentiment orientation, in order to identify eligible products for the questioner.

Sentiment Analysis. Sentiment orientation is another important supporting factor for our recommendation problem. Hyun et al. (Hyun et al. 2018) proposed a sentiment-based recommendation system. A two-step scheme is proposed: they first extracted sentiment vectors of the reviews through a single CNN; Then they merged the sentiment vectors to the review embeddings as the input of a simple dual CNNs for the rating prediction task. Experimental results proved that the sentiment representations can boost recommendation performance. However, performing the sentiment analysis task needs a large amount of labeled data, which is a bottleneck problem. (Tang et al. 2014) used the emoticons as the weak training signals for the sentiment classification on tweets, but this kind of signals has a strong connection with microblogging environments. Customer reviews seldom contain emoticons. (Guan et al. 2016) utilized the review ratings as the weak training signals to implement an anti-noise weakly-supervised training method for sentiment analysis. In this work, we adopt this training method to pre-train our Sentiment Net (S-Net).

The Method

In this section, we first clarify the problem definition and then introduce our proposed framework QDANN.

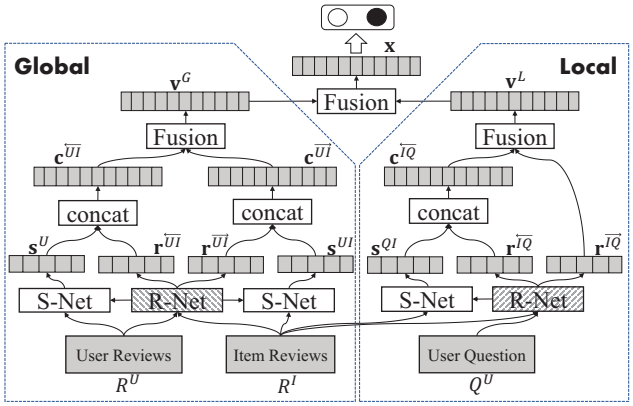

Figure 2: The Framework of QDANN.

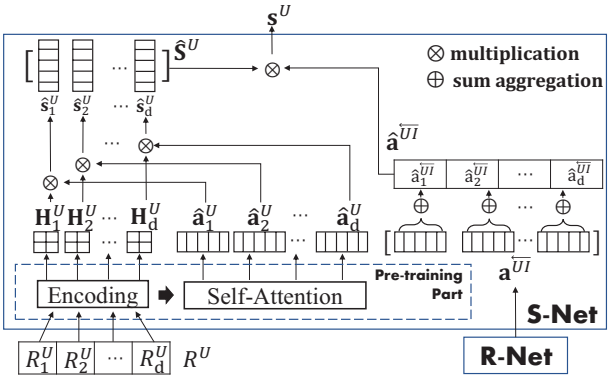

Figure 3: The sentiment net. Inputs are for the leftmost S-Net copy in Figure 2.

Problem Definition

Given a questioner U and an item I, let R^U denote U's historical reviews for products in the same domain as I. R^I represents the reviews of the item I. Q^U is a submitted question by U. $y \in \{0, 1\}$ is the binary label indicating whether U will purchase I or not. Given a set of tuples $T = \{(R^U, Q^U, R^I, y)\}$, our goal is to develop a model that can accurately predict y for a given (R^U, Q^U, R^I).

Framework

The architecture of QDANN is shown in Figure 2. Its key components from bottom to top include Relevance Net (R-Net), Sentiment Net (S-Net), Concatenation, Fusion and Output. Intuitively, Q^U conveys user instant demand; R^I contains I's pros/cons in various aspects; R^U represents U's general preference. The general idea of QDANN is to extract matched relevance information and the corresponding sentiment information between Q^U and R^I (eligibility of I w.r.t. Q^U), and between R^U and R^I (global eligibility of I for user U) respectively. The extracted useful information is then fused for making purchase prediction. In the next, we introduce each component in detail.

Relevance Net. The R-Nets are depicted in Figure 2, where different texture patterns are used to differentiate the two independent R-Nets. R-Nets are used to extract the representation of the relevance evidence. We use Q^U & R^I and R^U & R^I as the inputs of the two different R-Nets to capture the relevance matching patterns between R^I and R^U and between R^I and Q^U, respectively. We use separate R-Nets since the distribution of questions is intrinsically different from that of reviews (Chen et al. 2019). The R-Nets include two operations: text encoding and co-attention. Since the operations are the same, in the following we take R^U & R^I as input for illustration.

Text Encoding. We first concatenate all reviews of U (or I) as a sequence. The input sequences are treated as word sequences where each word is embedded as a 300-dimensional vector, i.e., $R^U =< \mathbf{w}_t^U >_{t=1}^n$, $R^I =< \mathbf{w}_t^I >_{t=1}^m$, according to the Chinese-Word-Vectors[1] (Li et al. 2018). Then we apply Bi-GRU as the encoding layer to extract low-level representations of the inputs. Compared to other RNN models such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), GRU is computationally efficient and can achieve competitive performance (Wang et al. 2017). A Bi-GRU is computed as follows:

$$\overrightarrow{\mathbf{h}_t} = \overrightarrow{GRU}(\overrightarrow{\mathbf{h}_{t-1}}, \mathbf{w}_t). \quad \overrightarrow{\mathbf{h}_t} \in \mathbb{R}^u \tag{1}$$

$$\overleftarrow{\mathbf{h}_t} = \overleftarrow{GRU}(\overleftarrow{\mathbf{h}_{t+1}}, \mathbf{w}_t). \quad \overleftarrow{\mathbf{h}_t} \in \mathbb{R}^u \tag{2}$$

where $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ represent forward and backward hidden states of Bi-GRU respectively. Then we concatenate $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ for each word $\mathbf{w}_t$ to get the complete hidden state $\mathbf{h}_t \in \mathbb{R}^{2u}$. We store $\mathbf{h}_t$'s of a sequence as column vectors of a matrix, i.e. $\mathbf{H}^U \in \mathbb{R}^{2u \times n}$ and $\mathbf{H}^I \in \mathbb{R}^{2u \times m}$ respectively.

Co-Attention. After the encoding layer, we use co-attention to capture the relevance matching patterns between the two low-level representations. The co-attention mechanism was first proposed in (dos Santos et al. 2016) and has been shown to be able to capture matching patterns between two distributions. It has three steps: First, we compute the affinity matrix between $\mathbf{H}^U$ and $\mathbf{H}^I$ as follows:

$$\mathbf{G} = \tanh((\mathbf{H}^I)^T \mathbf{M} \mathbf{H}^U). \quad \mathbf{G} \in \mathbb{R}^{m \times n} \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{2u \times 2u}$ is a parameter matrix. Second, we use row-wise summation and column-wise summation on $\mathbf{G}$ followed by a softmax function to generate attention vectors $\mathbf{a}^{\overrightarrow{UI}}$ and $\mathbf{a}^{\overleftarrow{UI}}$:

$$\mathbf{a}^{\overrightarrow{UI}} = \text{softmax}(\text{RowSum}(\mathbf{G})). \quad \mathbf{a}^{\overrightarrow{UI}} \in \mathbb{R}^m \tag{4}$$

$$\mathbf{a}^{\overleftarrow{UI}} = \text{softmax}(\text{ColSum}(\mathbf{G})). \quad \mathbf{a}^{\overleftarrow{UI}} \in \mathbb{R}^n \tag{5}$$

Last, the attention-weighted representations of R^U and R^I are calculated as:

$$\mathbf{r}^{\overleftarrow{UI}} = \mathbf{H}^U \mathbf{a}^{\overleftarrow{UI}}. \quad \mathbf{r}^{\overleftarrow{UI}} \in \mathbb{R}^{2u} \tag{6}$$

$$\mathbf{r}^{\overrightarrow{UI}} = \mathbf{H}^I \mathbf{a}^{\overrightarrow{UI}}. \quad \mathbf{r}^{\overrightarrow{UI}} \in \mathbb{R}^{2u} \tag{7}$$

For the submitted question and item reviews, we use another R-Net and perform Eq. (1) - Eq. (7) to generate the attention weighted representations, i.e. $\mathbf{r}^{\overleftarrow{IQ}} \in \mathbb{R}^{2u}$ and $\mathbf{r}^{\overrightarrow{IQ}} \in \mathbb{R}^{2u}$.

[1] https://github.com/Embedding/Chinese-Word-Vectors. It keeps fixed during the training

Sentiment Net. We design the S-Net to identify sentiment evidences related to local/global relevance matches from reviews to facilitate purchasing propensity analysis. We train a single S-Net for R^U and R^I since they both consist of reviews. Two copies of S-Net in Figure 2 are for matching between R^U and R^I, i.e., extracting sentiment information in user/item reviews for the global match. One copy is for the matching between Q^U and R^I. The computation flow of S-Net is shown in Figure 3. Intuitively, we want to identify sentiment evidences related to relevance matching. For example, in Figure 1 we want to identify the positive evidence "smoothly" in sentence "It runs the PUBG smoothly", so we would ignore irrelevant sentences in the review. This indicates that the co-attention weights generated by R-Nets should be considered when calculating sentiment representations of reviews. Thus, we propose a two-level attention mechanism for S-Net, where "two-level" means sentence-level and review-level. Since the operations of the three copies are the same, we take R^U & $\mathbf{a}^{\overleftarrow{UI}}$ as input for illustration. We first calculate the low-level representation $\mathbf{H}_i^U$ for each sentence $R_i^U \in R^U$ and then feed each $\mathbf{H}_i^U$ into a self-attention module (Lin et al. 2017) to generate its sentiment attention vector $\hat{\mathbf{a}}_i^U$. The design of the encoding layer is the same as in R-Nets. The attention vectors are computed as follows:

$$\hat{\mathbf{a}}_i^U = \text{softmax}(\mathbf{p}^T \tanh(\mathbf{U}^s \mathbf{H}_i^U)). \quad \hat{\mathbf{a}}_i^U \in \mathbb{R}^l \tag{8}$$

where $\mathbf{U}^s \in \mathbb{R}^{k \times 2u}$, $\mathbf{p} \in \mathbb{R}^k$ are parameters of the self-attention function with hyperparameter k, and l denotes the sentence length. Next, we obtain attention-weighted vector $\hat{\mathbf{s}}_i^U = \mathbf{H}_i^U \hat{\mathbf{a}}_i^U$ for each sentence R_i^U. These vectors are stored into a matrix $\hat{\mathbf{S}}^U \in \mathbb{R}^{2u \times d}$ as the sentence-level sentiment representation, where d is the total number of sentences in R^U. To make the sentiment representation of R^U conform to the relevance match between R^U and R^I, we perform aggregation in $\mathbf{a}^{\overleftarrow{UI}}$ sentence-wise to obtain a sentence-level relevance attention vector $\hat{\mathbf{a}}^{\overleftarrow{UI}} \in \mathbb{R}^d$ (see Figure 3). The review-level sentiment representation of R^U is then calculated:

$$\mathbf{s}^U = \hat{\mathbf{S}}^U \hat{\mathbf{a}}^{\overleftarrow{UI}}. \quad \mathbf{s}^U \in \mathbb{R}^{2u} \tag{9}$$

In this way, $\mathbf{s}^U$ is forced to concentrate on sentences with salient relevance scores in matching. The outputs of the other two copies of S-Net (i.e. $\mathbf{s}^{UI} \in \mathbb{R}^{2u}$ and $\mathbf{s}^{QI} \in \mathbb{R}^{2u}$) are calculated in the same way.

Concatenation, Fusion & Output. After obtaining the attention-weighted relevance representations and sentiment representations, we concatenate them to form informative feature vectors:

$$\mathbf{c}^{\overleftarrow{UI}} = [\mathbf{r}^{\overleftarrow{UI}}, \mathbf{s}^U]. \quad \mathbf{c}^{\overleftarrow{UI}} \in \mathbb{R}^{4u} \tag{10}$$

$$\mathbf{c}^{\overrightarrow{UI}} = [\mathbf{r}^{\overrightarrow{UI}}, \mathbf{s}^{UI}]. \quad \mathbf{c}^{\overrightarrow{UI}} \in \mathbb{R}^{4u} \tag{11}$$

$$\mathbf{c}^{\overleftarrow{IQ}} = [\mathbf{r}^{\overleftarrow{IQ}}, \mathbf{s}^{QI}]. \quad \mathbf{c}^{\overleftarrow{IQ}} \in \mathbb{R}^{4u} \tag{12}$$

where [*,*] represents the concatenation operation (the concat box in Figure 2). $\mathbf{c}^{\overleftarrow{UI}}$ and $\mathbf{c}^{\overrightarrow{UI}}$ encode sentiment information of user/item reviews and also their relevance matching patterns, while $\mathbf{c}^{\overleftarrow{IQ}}$ contains the relevance matching patterns in item reviews w.r.t. the question, in addition to the associated sentiment information. We perform fusion (Rocktäschel et al. 2015) which has shown good performance for NLP tasks. We impose this operation on $\mathbf{c}^{\overleftarrow{IQ}}$ & $\mathbf{r}^{\overrightarrow{IQ}}$ and $\mathbf{c}^{\overleftarrow{UI}}$ & $\mathbf{c}^{\overrightarrow{UI}}$ to generate the final representations of eligibility evaluations from the questioner's local demands and global preference respectively:

$$\mathbf{v}^L = \tanh(\mathbf{W}^{\overleftarrow{IQ}} \mathbf{c}^{\overleftarrow{IQ}} + \mathbf{W}^{\overrightarrow{IQ}} \mathbf{r}^{\overrightarrow{IQ}}). \quad \mathbf{v}^L \in \mathbb{R}^{2u} \tag{13}$$

$$\mathbf{v}^G = \tanh(\mathbf{W}^{\overleftarrow{UI}} \mathbf{c}^{\overleftarrow{UI}} + \mathbf{W}^{\overrightarrow{UI}} \mathbf{c}^{\overrightarrow{UI}}). \quad \mathbf{v}^G \in \mathbb{R}^{2u} \tag{14}$$

where $\mathbf{W}^{\overleftarrow{IQ}}$, $\mathbf{W}^{\overrightarrow{IQ}}$, $\mathbf{W}^{\overleftarrow{UI}}$ are $2u \times 4u$ parameter matrices, and $\mathbf{W}^{\overrightarrow{UI}}$ is a $2u \times 2u$ parameter matrix. We then further fuse $\mathbf{v}^L$ and $\mathbf{v}^G$:

$$\mathbf{x} = \tanh(\mathbf{W}^L \mathbf{v}^L + \mathbf{W}^G \mathbf{v}^G). \quad \mathbf{x} \in \mathbb{R}^{2u} \tag{15}$$

where $\mathbf{W}^L$ and $\mathbf{W}^G$ are $2u \times 2u$ parameter matrices. Finally, we compute the output prediction based on the synthesized information as:

$$\hat{y} = \sigma((\mathbf{w})^T \mathbf{x} + b). \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function. The output $\hat{y}$ represents the probability of U purchasing I. An item with a higher probability score has higher priority to be recommended to the corresponding questioner.

Loss Function. We use the standard cross-entropy function as the loss function for the purchase prediction task:

$$\mathcal{L} = -\sum y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \tag{17}$$

Training Strategy

We design a transfer training strategy: we first perform pre-training of R-Nets and S-Net by using aspect relevance prior knowledge (the relevance task) and review ratings (the sentiment task) respectively. Then the pre-trained subnets are used to initialize the QDANN and we fine-tune the whole network on the tuple dataset T (the purchase prediction task). Next we first describe the pre-training of R-Net and S-Net and then give the details of fine-tuning QDANN.

Pre-training for R-Net. In order to pre-train R-Nets, we employ aspect keywords as prior knowledge to assess the coarse-grained aspect relevance among reviews. Specifically, we first collect aspect keywords from Taobao's database[2], and then sample (question, review sentence) pairs (QR) and (review sentence, review sentence) pairs (RR) where pairs with/without overlapping aspect keywords are deemed to be positive/negative instances for the respective relevance tasks. For training, the two outputs (Eqs. (6) and (7)) of R-Net are fused by the fusion operation and then employed for 0/1 relevance prediction. The cross-entropy loss is used. Although the pre-training can only capture the coarse-grained aspect relevance between the input pairs, the purchase prediction task could further fine-tune the R-Nets

Table 1: Dataset statistics

Domain	Pre-training for R-Net		Pre-training for S-Net	Main Task		
	QR	RR	Review sentence	Positive Tuple	Item	Questioner
Cellphone	120,000	120,000	250,000	2,500	397	2,109
Smart television	100,000	100,000	150,000	2,100	303	1,923
Washing machine	80,000	80,000	100,000	1,900	286	1,768

to capture the fine-grained relevance patterns that can explain the training purchase behaviors.

Pre-training for S-Net. The parameters of S-Net affected by pre-training is contained in the dotted-line box of Figure 2. Since the ratings of customer reviews can reflect the overall sentiment orientation of the corresponding reviews, we use them as the training labels for the sentiment task and follow the weakly-supervised training strategy proposed in (Guan et al. 2016). Specifically, we assign the rating (1 - 5 stars) of a review (R) to each sentence (R_i) of it as the weak sentiment label. We adopt a simple rule to achieve weak binary labels: ratings higher than 3-star are treated as positive labels and those lower than 3-star are treated as negative labels. We discard reviews with 3-star ratings since those reviews could contain diverse sentiments and degenerate the quality of weak labels. The following triple-based ranking loss is adopted as the training criterion:

$$\mathcal{L}_s = \sum_{<R_1,R_2,R_3>} \max\left(0, \lambda - \|\hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_3\|_2 + \|\hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_2\|_2\right), \tag{18}$$

where $< R_1, R_2, R_3 >$ denotes a triple with weak labels $\ell(R_1) = \ell(R_2) \neq \ell(R_3)$, λ is the margin parameter, and $\|\cdot\|_2$ represents the Euclidean distance based on the sentence sentiment representation $\hat{\mathbf{s}}_i$ calculated by S-Net. This equation means we require the distance between same-label sentences to be shorter than that between the opposite-label sentences by at least λ. Compared to using weak labels directly in supervised-style training, this training strategy can mitigate the negative effects of wrong-labeled sentences and lead to a good sentiment representation (Guan et al. 2016).

Fine-tuning for QDANN. After pre-training, we initialize QDANN by the pre-trained subnets and then use supervised-style learning to train the whole network for purchase prediction. In experiments, we will investigate the impact of pre-training on model performance and demonstrate its usefulness.

Experiments

Dataset and Preprocessing

In this section, we evaluate our method on a dataset collected from Taobao.com. The dataset covers three domains including "cellphones", "smart televisions","washing machines". We choose the above three domains because the products of them have many attributes and aspects to consult about, and customers often consult others for more details when buying high price products, so that we could get enough and diverse questions. The dataset statistics are shown in Table 1. For pre-training, we randomly collect 300,000 QR pairs and 300,000 RR pairs in total for the two relevance tasks, and collect 500,000 review sentences for the sentiment task. To obtain the dataset for the main task, we first collect questions submitted between May 1st, 2017 to May 1st, 2018. Then we set a rule to collect positive training tuples (i.e. looking for product purchases related to the questions): the purchase time must be after the question's time and the distance in time must not exceed 2 months. The reason for such a long distance constraint is that some customers looking for the above products (e.g. cellphones) may not be able to afford instantly or would be in hesitation for a while, since they cost a lot of money. In this way, we obtain positive tuples (i.e. $(R^U, Q^U, R^I, 1)$) where each question only has one positive tuple because most consumers do not frequently buy the products of the chosen domains. We collect reviews for the involved users and products, and design a simple preprocessing scheme: we first discard short reviews (i.e. fewer than 5 words) and reviews without any aspect keywords, and then concatenate reviews for each user/item. Most concatenated sequences are not too long ($\leq$500 words) and can be used as the input for the network. For a few too long sequences (>500 words, tail in the power law distribution), we conduct multiple under-sampling on each sequence to generate duplicate tuples with sampled 500-word[3] subsequences. All the preprocessed positive tuples are split into training set (70%), validation set (10%) and test set (20%). For the training set, we randomly replace the items in positive tuples by different items in other positive tuples in the same domain to generate negative tuples. Note that different items from other positive tuples actually means 396/302/285 (i.e., just minus 1) of all the involved items in the three domains. Hence, this would not lead to a bias in the training set. The positive/negative ratio of all the training sets are 1:1.

Implementation Details

We use the Adam (Kingma and Ba 2014) for training. We set the learning rate to 1e-6. The first and second momentum coefficients are set to 0.9 and 0.999 respectively. We follow the empirical conclusion in (Guan et al. 2016) to set λ in Eq. (18) to 0.5. Both the length of the hidden state vectors of GRUs (u) and the hyperparameter (k) are both set to 64 according to the parameter study. The mini-batch size for SGD is set to 32.

[2]Aspect keyword can also be mined from reviews by state-of-art opinion mining techniques (Poria, Cambria, and Gelbukh 2016; Xiao et al. 2018; Liu et al. 2015). In this work, we simply use Taobao's database since it is of high quality.

[3]Approximately 500, truncated at sentence boundaries.

Table 2: Performance comparison

Method	Cellphones			Smart televisions			Washing machines		
	succ@1	succ@5	succ@10	succ@1	succ@5	succ@10	succ@1	succ@5	succ@10
Match	0.0737	0.1552	0.3237	0.0693	0.1452	0.3201	0.0594	0.1293	0.3042
FM	0.0789	0.1710	0.3395	0.0957	0.1782	0.3564	0.0664	0.1399	0.3217
NeuroMF	0.1026	0.1658	0.3684	0.1155	0.2013	0.3828	0.0804	0.1643	0.3636
DeepCoNN	0.1105	0.1789	0.3895	0.1353	0.2244	0.3960	0.0944	0.1958	0.4021
SentiRec	0.1120	0.1837	0.3921	0.1518	0.2475	0.4224	0.1119	0.2272	0.4301
MPCN	0.1158	0.1842	0.4079	0.1683	0.2640	0.4389	0.1224	0.2378	0.4441
AP	0.1053	0.1763	0.4211	0.1815	0.2607	0.4554	0.1399	0.2517	0.4755
QDANN-rand	0.1289	0.2105	0.4079	0.1881	0.2706	0.4488	0.1258	0.2483	0.4510
QDANN	**0.1658**	**0.2368**	**0.4789**	**0.1980**	**0.2904**	**0.4950**	**0.1643**	**0.2727**	**0.5175**

Evaluation

Since the chosen products are not a cheap daily consumable as food, most consumers do not frequently buy the chosen products during a period of time. In our collected 1-year purchase data, there is only one relevant purchase for each user submitted question. Hence, our task is more difficult than traditional retrieval or recommendation problems where there are multiple relevant items. For evaluation, we take each positive tuple in the test set as a test case (and ground truth target), and rank all the involved products in the same domain according to their model-estimated scores with the corresponding user and question (if the model concerns the question). Since each test case only has one relevant result, we propose a metric similar to average of precision@N called success@N (succ@N for short):

$$succ@N = \frac{\sum_{i=1}^{S} TP_i^N}{S} \tag{19}$$

where S is the total number of test cases. TP_i^N is a binary indicator: if the ground truth product is in the top N ranked candidates, then $TP_i^N = 1$; otherwise, $TP_i^N = 0$. We set N to 1, 5 and 10 to evaluate our method under varying difficulty.

It is worth noting that the positive/negative ratio is 1:1 for model training. In the test phase, however, we take each positive tuple in the test set as a test case and rank all the involved products according to their model-estimated scores. Hence, the ratio is 1:(x-1), where x is the total number of involved items of one domain in our dataset, i.e. the positive/negative ratio of cellphones/smart televisions/washing machine is 1:396/1:302/1:285 respectively. This is in accordance with the real situation where positive examples are rare.

Baselines and Main Results

Match. This baseline simply performs mean pooling on the input word embeddings of R^I, Q^U and R^U to generate fixed-length vectors. Then we use dot product to compute match scores of R^I & Q^U and R^I & R^U, and take the average.

FM. Factorization machine (FM) is a popular model for recommendation (Rendle 2012). We feed the fixed-length vectors of R^I, Q^U and R^U obtained as above into the factorization machine. For the sake of fairness, we train this model (and also all the following baselines) on the training tuples for purchase prediction.

NeuroMF. This baseline is a state-of-the-art collaborative filtering method (He et al. 2017). It fuses matrix factorization into a neural network. NeuroMF takes R^I and R^U as input.

DeepCoNN. This baseline is a classic FM-based neural network for recommendation (Zheng, Noroozi, and Yu 2017). It takes R^I and R^U as input.

SentiRec. This baseline (Hyun et al. 2018) fuses sentiment information into a network with similar model structure as DeepCoNN. It takes R^I and R^U as input.

MPCN. This is the state-of-the-art review-based FM neural network with a Gumbel-Max co-attention module (Tay, Tuan, and Hui 2018). It takes R^I and R^U as input.

AP. The Attentive Pooing (AP) network (dos Santos et al. 2016) was first proposed for the question answering problem. We use (Q^U, R^I) pairs as input to train the network for purchase prediction.

QDANN-rand. This baseline uses QDANN without pretraining. All the parameters are randomly initialized.

The main results are shown in Table 2. The comparison results of the three domains show similar evidences. Match and FM perform poorly. It indicates that simple Match or FM without high-level feature extraction operations cannot handle the problem. NeuroMF applies a multilayer-perceptron with matrix factorization to extract high-level features from the input word embedding matrices. Compared to input embeddings, high-level features can represent more useful information. But it performs worse than DeepCoNN. This could be because the latter employs the convolution layers to capture the local context of texts. However, limited by the size of filters, it cannot capture long dependency in complex sequences. SentiRec employs a similar model structure as DeepCoNN but achieves better performance than DeepCoNN. This indicates that the sentiment information is useful for recommendation. MPCN performs better than SentiRec, which indicates that explicitly exploiting relevance between the target user's reviews and the target item's reviews is also useful for purchase prediction. AP achieves consistently better performance than MPCN on succ@10. The reason might be that it uses questions and item reviews as input and captures the local demands from the questions. QDANN-rand achieves even better perfor-

Table 3: Ablation study

Method	Cellphones			Smart televisions			Washing machines		
	succ@1	succ@5	succ@10	succ@1	succ@5	succ@10	succ@1	succ@5	succ@10
No PT for S-Net	0.1237	0.2211	0.4395	0.1716	0.2640	0.4653	0.1399	0.2483	0.4790
No PT for R-Net	0.1184	0.2158	0.4263	0.1584	0.2409	0.4488	0.1294	0.2343	0.4685
No LD	0.1211	0.1789	0.3342	0.1221	0.1782	0.3696	0.1119	0.1853	0.3287
No GP	0.1263	0.2052	0.4105	0.1353	0.2145	0.4323	0.1154	0.2238	0.4406
QDANN	**0.1658**	**0.2368**	**0.4789**	**0.1980**	**0.2904**	**0.4950**	**0.1643**	**0.2727**	**0.5175**

mance but is still worse than QDANN. This indicates pre-training on the relevance tasks and the sentiment task has positive influence on the recommendation performance. The similar results on the three different product domains indicate that our method does not depend on domain-specific knowledge, and therefore it can be easily generalized to different product domains.

Ablation Study

In this part, we analyze the factors affecting the recommendation performance. The ablation results are shown in Table 3, where PT, LD and GP represents "pre-training", "local demands" and "general preference" respectively. "No local demands " means we remove $\mathbf{v}^L$ (the final representation of eligibility evaluation from the questioner's local demands) before the final fusion layer, i.e., setting $\mathbf{v}^L = \mathbf{0}$. "No general preference" means we remove $\mathbf{v}^G$ (the final representation of eligibility evaluation from the questioner's global preference) before the final fusion layer. It can be seen that all of the three factors are important for the question-driven recommendation task. When removing "local demands", the performance drops more dramatically, meaning users' local demands take a more important role for purchase prediction. This well proves the importance of our proposed question-driven recommendation problem for stimulating consumption on e-commerce Websites.

Parameter Study

There are two important parameters in our methods: the output dimension of Bi-GRU (u) and the hyperparameter (k) of the self-attention. We tune them on the validation set. The results are shown in Figure 4. k and u have a similar trend: the performance is the best when they are given sufficiently large values. Nevertheless, when the value is too large, we could overfit the training data with a large number of parameters.

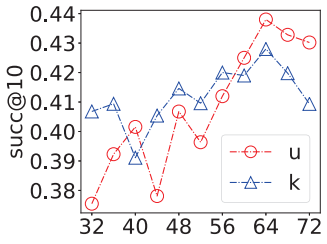

Figure 4: Parameter tuning.

Questions	Reasons
这个能玩王者荣耀吗? Can it run the Arena Of Valor?	用它玩王者荣耀毫无压力。 It runs the Arena Of Valor without any problem.
充满电能用一天吗? Can a full battery last a whole day ?	看了两个小时网络视频只用了3%的电量。 After watching the online videos for 2 hours, the battery power was reduced by only 3%.
画质音质如何 ? How about the image quality and sound quality ?	画质清晰，声音立体。 The image quality is clear and the sound is solid.
脚架是送的吗? Is the TV bracket free ?	带免费支架。 Free bracket.
有噪音吗 ? Is it noisy ?	洗衣噪声很小。 There is very little noise while washing.
烘干要多久 ? How long does it take to dry ?	烘干只要8分钟。 Drying only needs 8 minutes.

Figure 5: Case studies.

Case Studies of Recommendation Reasons

For the predictions with high estimated probabilities, we further use the attention mechanisms to identify the recommendation reason sentences from the item reviews. We select the sentences including words with the highest co-attention weights (representing relevance) as the reasons for the corresponding questions. The questions and the corresponding reasons are shown in Figure 5. The underlined words are those with the highest co-attention weights and the words in boxes are those with the highest self-attention weights. It reveals that (1) the co-attention weights indeed capture relevant review sentences to the specific local demands in questions; (2) self-attention weights can identify sentiment-related words. For another, we have investigated the global side attention results and found some of them are flat distributions, but some provide meaningful results. To decide whether the attention results can provide recommendation reasons, we could compute the entropy of co-attention distributions in reviews. Intuitively, a more skewed distribution indicates higher confidence of relevance. Hence, we could set a threshold of entropy to judge whether we can output recommendation reasons.

Conclusion

In this work, we explore a new recommendation paradigm that analyzes user purchasing propensity based on his/her submitted question to recommend products. We mine the useful local demands and general preference from the submitted question and historical reviews of the questioner to help evaluate the purchasing propensity. Empirical results show the effectiveness of our method.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61672409, 61936006, 61876144, 61522206, 61901370), the Major Basic Research Project of Shaanxi Province (Grant No. 2017ZDJC-31), Shaanxi Province Science Fund for Distinguished Young Scholars (Grant No. 2018JC-016), the Fundamental Research Funds for the Central Universities (Grant Nos. JB190301, JB190305) and the National Science Foundation (Grant IIS-1815674).

References

Catherine, R., and Cohen, W. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 288–296. ACM.

Chen, L.; Guan, Z.; Zhao, W.; Zhao, W.; Wang, X.; Zhao, Z.; and Huan, S. 2019. Answer identification from product reviews for user questions by multi-task attentive networks. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 45–52.

dos Santos, C. N.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive pooling networks. *CoRR, abs/1602.03609*.

Guan, Z.; Chen, L.; Zhao, W.; Zheng, Y.; Tan, S.; and Cai, D. 2016. Weakly-supervised deep learning for customer review sentiment classification. In *IJCAI*, 3719–3725.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182. International World Wide Web Conferences Steering Committee.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hyun, D.; Park, C.; Yang, M.-C.; Song, I.; Lee, J.-T.; and Yu, H. 2018. Review sentiment-guided scalable deep recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 965–968. ACM.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, S.; Zhao, Z.; Hu, R.; Li, W.; Liu, T.; and Du, X. 2018. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*.

Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *CoRR* abs/1703.03130.

Liu, Q.; Gao, Z.; Liu, B.; and Zhang, Y. 2015. Automated rule selection for aspect extraction in opinion mining. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Liu, X. 2015. Modeling users' dynamic preference for personalized recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Lu, Y.; Dong, R.; and Smyth, B. 2018. Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 773–782. International World Wide Web Conferences Steering Committee.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* sampling. In *Advances in Neural Information Processing Systems*, 3086–3094.

McAuley, J., and Yang, A. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, 625–635. International World Wide Web Conferences Steering Committee.

Poria, S.; Cambria, E.; and Gelbukh, A. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108:42–49.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(3):57.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kočiskỳ, T.; and Blunsom, P. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Tang, D.; Wei, F.; Qin, B.; Liu, T.; and Zhou, M. 2014. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 208–212.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. *arXiv preprint arXiv:1801.09251*.

Wan, M., and McAuley, J. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 86–94. ACM.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *Meeting of the Association for Computational Linguistics*, 189–198.

Xiao, D.; Ji, Y.; Li, Y.; Zhuang, F.; and Shi, C. 2018. Coupled matrix factorization and topic modeling for aspect mining. *Information Processing & Management* 54(6):861–873.

Yu, Q., and Lam, W. 2018. Aware answer prediction for product-related questions incorporating aspects. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 691–699. ACM.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 425–434. ACM.