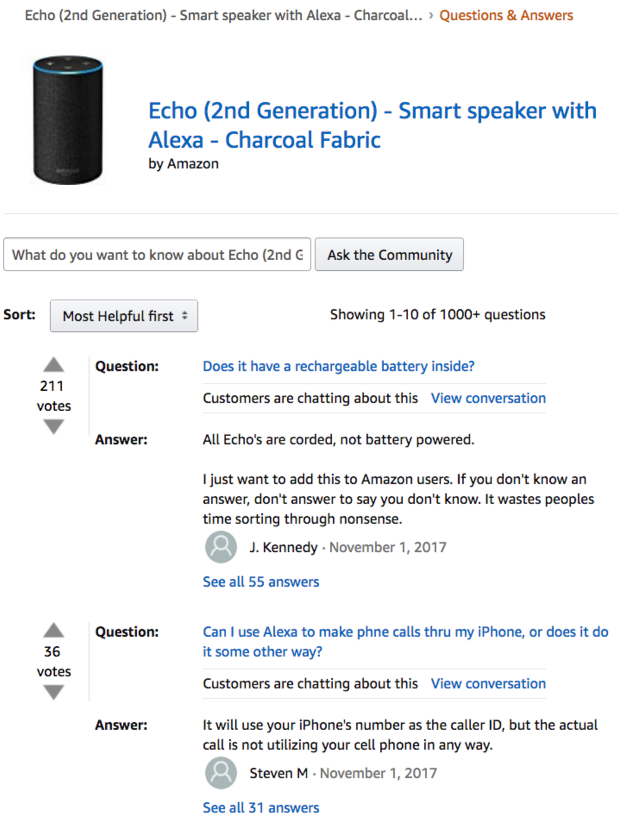
Reading Customer Reviews to Answer Product-related Questions

Miao Fan   Chao Feng   Mingming Sun   Ping Li   Haifeng Wang

Cognitive Computing Lab (CCL), Baidu Research, Baidu Inc.
{fanmiao,v_fegchao,sunmingming01,liping11,wanghaifeng}@baidu.com

Abstract

The e-commerce websites are ready to build the community question answering (CQA) service, as it can facilitate questioners (potential buyers) to obtain satisfying answers from experienced customers and furthermore stimulate consumption. Given that more than 50% product-related questions only anticipate a binary response (i.e., “Yes” or “No”), the research on product-related question answering (PQA), which aims to automatically provide instant and correct replies to questioners, emerges rapidly. The mainstream approaches on PQA generally employ customer reviews as the evidence to help predict answers to the questions which are product-specific and concerned more about subjective personal experiences. However, the supportive features either extracted by heuristic rules or acquired from unsupervised manners are not able to perform well on PQA. In this paper, we contribute an end-to-end neural architecture directly fed by the raw text of product-related questions and customer reviews to predict the answers. Concretely, it teaches machines to generate and to synthesize multiple question-aware review representations in a reading comprehension fashion to make the final decision. We also extract a real-world dataset crawled from 9 categories in Amazon.com for *PQA* to assess the performance of our neural reading architecture (NRA) and other mainstream approaches such as COR-L [12], MOQA [12], and AAP [21]. Experimental results show that our NRA sets up a new state-of-the-art performance on this dataset, significantly outperforming existing algorithms.



Echo (2nd Generation) - Smart speaker with Alexa - Charcoal... › Questions & Answers

Echo (2nd Generation) - Smart speaker with Alexa - Charcoal Fabric
by Amazon

What do you want to know about Echo (2nd G | Ask the Community

Sort: Most Helpful first   Showing 1-10 of 1000+ questions

211 votes
Question: Does it have a rechargeable battery inside?
Customers are chatting about this View conversation
Answer: All Echo's are corded, not battery powered.

I just want to add this to Amazon users. If you don't know an answer, don't answer to say you don't know. It wastes peoples time sorting through nonsense.
J. Kennedy · November 1, 2017
See all 55 answers

36 votes
Question: Can I use Alexa to make phne calls thru my iPhone, or does it do it some other way?
Customers are chatting about this View conversation
Answer: It will use your iPhone's number as the caller ID, but the actual call is not utilizing your cell phone in any way.
Steven M · November 1, 2017
See all 31 answers

Figure 1: A snapshot of the QA interactions talking about the *Echo (2nd Generation)* sold on Amazon.com.

1 INTRODUCTION

The e-commerce websites such as `www.amazon.com` and `www.ebay.com` are ready to build the community question answering (CQA) service (see Figure 1), as it can facilitate questioners (potential purchasers) to obtain satisfying answers from experienced customers. Satisfying answers can help build the confidence of those potential buyers and stimulate consumption. A recent study [12] conducted on these product-associated QA interactions in Amazon.com indicates that more than 50% product-related questions just anticipate a binary answer (i.e., “Yes” or “No”). It leads to an emerging and promising study on automated *product-related question answering* (PQA) which attempts to teach machines to automatically provide instant and correct replies to the questioners (potential buyers) on e-commerce websites. The instant and correct replies can help the potential buyers to avoid the time-consuming waiting for a simple Yes/No response, and hence are able to drive a higher conversion rate for e-commerce.

Given a product-related question, the mainstream approaches on PQA, including COR-L [12], MOQA [12], and AAP [21], generally employ customer reviews as the evidence to help predict the answer. The reasons for exploiting reviews are that those questions are product-specific and many of them are concerned about user experiences and subjective opinions, which are mostly stated in customer reviews. To find out the supportive features for the answer prediction in PQA, COR-L [12] and MOQA [12] adopt several standard functions including Cosine similarity, Okapi BM25 [10] and Rough-L [9] to measure the similarity between product-related questions and customer reviews. The state-of-the-art approach AAP [21] argues that these heuristic functions tend to bring in irrelevant features in terms of the various aspects implied in customer reviews and product-related questions. Therefore, AAP [21] proposes a three-order AutoEncoder [4] to discover the hidden aspects at first in an unsupervised manner and to select a subgroup of customer reviews which have the same aspect with the corresponding product-related question as the prior knowledge. However, the supportive features either extracted by heuristic rules [12] or acquired from unsupervised manners [21] are not able to perform well on PQA, as they cannot be directly optimized by the learning target (i.e., the correct answer).

To address the problem, we contribute an end-to-end neural reading architecture (NRA) for PQA in this paper which can be directly fed by the raw text of product-related questions and customer reviews to predict the answers in a reading comprehension fashion [3, 16]. Concretely, it teaches machines to adapt the underlying feature representations, so as to generate and synthesize multiple question-aware embeddings of reviews as evidence to make the correct decision. We also build a real-world dataset crawled from 9 categories in Amazon.com for successive studies on *PQA*. It has been used as the benchmark dataset to assess the performance of our neural reading architecture (NRA) and other mainstream approaches such as COR-L [12], MOQA [12], and AAP [21]. Experimental results demonstrate that our NRA sets up a new state-of-the-art performance at 77.35% accuracy@50% and 61.76% AUROC on the dataset, surpassing the modern methods with a significant improvement by **5.70%** accuracy@50% and **8.46%** AUROC.

The rest of this paper is organized as follows. Section 2 reviews related work on PQA, and we present our neural reading architecture in Section 3. Section 4 showcases the effectiveness of the proposed model in the experiments. Finally, Section 5 concludes the paper and leaves several open questions for successive research.

2 RELATED WORK

To the best of our knowledge, the mainstream approaches proposed for automated PQA includes COR-L [12], MOQA [12], and AAP [21]. Generally speaking, all these methods on PQA aim at modeling the conditional probability, denoted as $Pr(a|q, R^q)$, of predicting a binary answer $a \in \{1, 0\}$ (where $1 = Yes$ and $0 = No$), given a product-related question q associated with the set of customer reviews R^q about the product.

Consider a training set Δ where $(q_i, R_i^q, a_i) \in \Delta$ containing n training instances. The likelihood $\mathcal{L}$ of the whole training set is formulated as follows,

$$\mathcal{L} = \prod_{i=1}^{n} [Pr(a_i = 1|q_i, R_i^q)]^{a_i} [1 - Pr(a_i = 1|q_i, R_i^q)]^{1-a_i}. \tag{2.1}$$

And it is easier to set the learning objective as maximizing $\log \mathcal{L}$ with the help of the mini-batch gradient ascend algorithm [19].

2.1 COR-L and MOQA

The baseline methods on PQA, i.e., COR-L [12] and MOQA [12], generally adopt a classical framework named *Mixtures of Experts* (MoEs) [6] which combines the outputs of several classifiers (or "experts") by associating weighted confidence scores with each classifier.

In COR-L [12] and MOQA [12], each customer review $r^q \in R^q$ is regarded as an "expert" to answer the product-related question q. More concretely, each "expert" helps to predict the final answer a to the question q by defining a "relevance" function S and a "voting" function V respectively. The function S is the critical parameter in a softmax classifier to produce the normalized score of "relevance" for each customer review and the function V controls the parameter of logistic regression to "vote" for the orientation of the final answer to the question q.

The difference between COR-L and MOQA is in the way of defining the "relevance" function S and the "voting" function V:

- For COR-L, the "relevance" function is determined by a set of existing pairwise similarity measures for q and r^q, including the Cosine similarity, Okapi BM25 [10] and Rough-L [9][1]. The "voting" function of COR-L is concerned about the bag-of-words (BOW) feature of the text of a product-related question and its corresponding reviews.
- The "relevance" and "voting" function of MOQA[2],

[1] COR-L is an abbreviation of the Cosine similarity, Okapi BM25 and Rough-L.

[2] MOQA is short for *Mixtures of Opinions for Question Answering.*

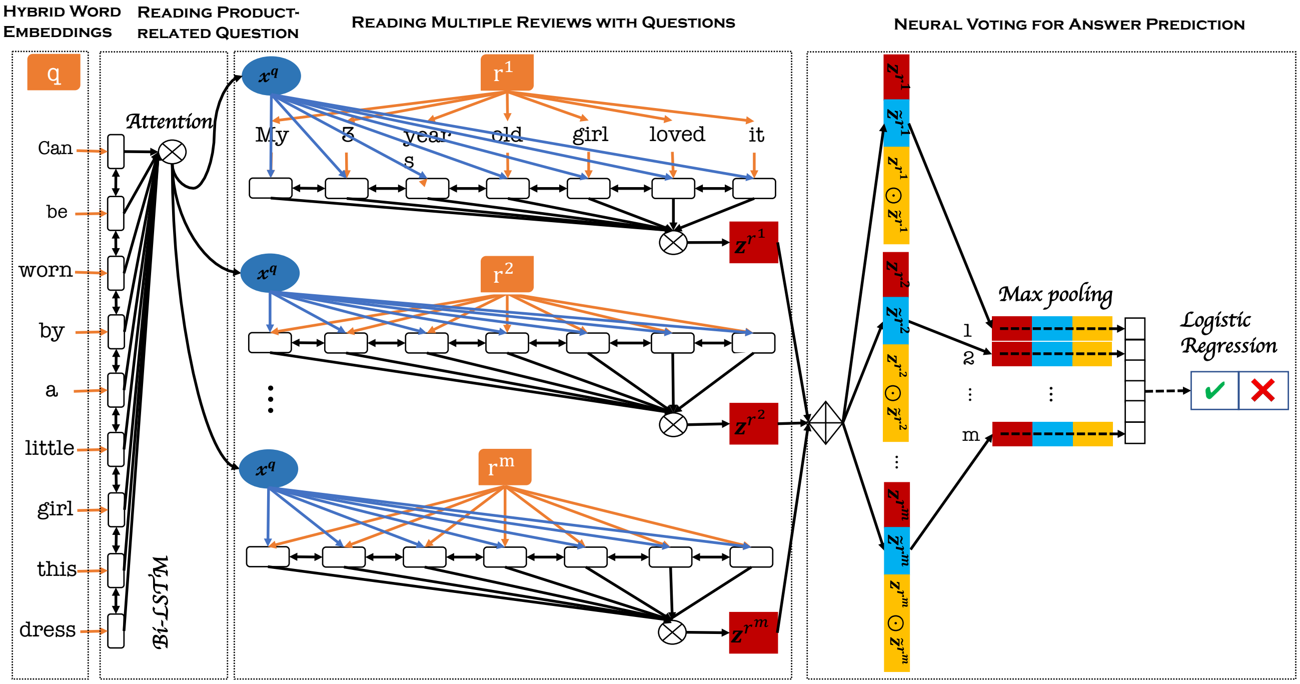


Figure 2: Illustration of the proposed end-to-end neural reading architecture (NRA) for the problem of product-related question answering (PQA).

goes one more step forward by exploring the bilinear relationship between the combined features (i.e., pairwise similarity measures and BOW features) of questions and customer reviews.

2.2 Aspect-based Answer Prediction (AAP)

Superior to the heuristic features such as the cosine similarity, Okapi BM25 [10], Rough-L [9] and bag-of-words (BOW) which are leveraged by the baseline approaches (COR-L [12] and MOQA [12]) on PQA, AAP [21] proposes to automatically acquire the aspect-specific embeddings of customer reviews and product-related questions via a 3-order AutoEncoder [4].

After we have acquired the aspect-specific embeddings of a product-related question and the product-associated review, AAP [21] improves the framework of MoEs [6] by taking the aspects into consideration. Specifically, it selects a subgroup of customer reviews which share the same aspect with the corresponding product-related question as the “experts”. The “relevance” function S is defined as the cosine similarity between the aspect distribution of a product-related question and its associated reviews, and the “voting” function V models the bilinear relationship between the aspect-specific embeddings of the question and reviews.

3 NEURAL READING ARCHITECTURE

In this section we present the detail of our end-to-end *neural reading architecture* (NRA) for PQA, which is composed of 4 neural layers from bottom to top illustrated by Figure 2: 1) It starts with a neural layer which produces a hybrid word embedding for each word in product-related questions and customer reviews; 2) The hybrid embeddings of the words in a product-related question are fed into a “Bi-LSTM [5,20] + Attention [1]” module for machines to read and achieve the contextual embedding of the question; 3) The contextual embedding of the question is concatenated with each hybrid embedding of word in multiple customer reviews as the question-aware word embeddings, which are fed into the same neural reading module “Bi-LSTM + Attention” (with different parameters) to generate multiple question-aware review representations; 4) A neural voting layer is designed for those question-aware review representations to mutually verify the most significant evidence for answer prediction.

3.1 Starting with Hybrid Word Embeddings

Either a product-related question q or its associated customer review r^q is a sequence of words. As the input of NRA, we first map the words into their adaptable embeddings. For a word u, we can initialize the embedding of u with a d-dimensional pre-trained word vector $\mathbf{u} \in \mathbb{R}^d$ such as Word2vec [13], Glove [17] or ELMo [18] in the training phase, and re-use the fine-tuned word vector if the word u also appears in the test set.

What if u is an OOV (out-of-vocabulary) word in the test set? To address this issue, we adopt the convolutional neural network (CNN) [8] to generate word embeddings from characters [22] which is also successfully used by [2]. Suppose that the word u is composed of a sequence of l characters denoted by $v_{1:l}$:

$$v_{1:l} = [v_1, v_2, ..., v_l]. \tag{3.2}$$

The vector representations of these characters are:

$$\mathbf{v}_{1:l} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_l], \tag{3.3}$$

where we use d' to denote the dimension of character-level embeddings: $\mathbf{v}_j \in \mathbb{R}^{d'}$. If we apply a filter $\mathbf{w}_i \in \mathbb{R}^{d' \times l'}$ which stands for the i-th filter with a window covering l'-length characters into a sequence of characters $\mathbf{v}_{j:j+l'-1} \in \mathbb{R}^{d' \times l'}$, this filter can help produce a feature map $f_{i,j}$ of the character l'-grams with the technique of narrow convolution [7]:

$$f_{i,j} = \text{ReLU}(\mathbf{w}'_i \cdot \mathbf{v}_{j:j+l'-1} + b'), \tag{3.4}$$

where $b' \in \mathbb{R}$ is a bias term corresponding to a series of filters $\mathbf{W}' = [\mathbf{w}'_1, \mathbf{w}'_2, ..., \mathbf{w}'_{k'}]$ and ReLU [14] is the rectified activation function.

The feature map $\mathbf{F} = [\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, ..., \mathbf{f}_{k'}^{\mathrm{T}}]^{\mathrm{T}}$ encodes k' kinds of l'-gram character embeddings of the word u. In order to acquire the fix-dimension character-level encoding (denoted by $\mathbf{h}$) of the word, we then apply the max-pooling strategy to each row of $\mathbf{F}$:

$$\mathbf{h} = \left[\max(\mathbf{f}_1^{\mathrm{T}}), \max(\mathbf{f}_2^{\mathrm{T}}), ..., \max(\mathbf{f}_{k'}^{\mathrm{T}})\right]^{\mathrm{T}}, \tag{3.5}$$

where $\mathbf{h} \in \mathbb{R}^{k'}$ is the k'-dimension encoding of the word u regardless of its length.

After obtaining both word-level (e.g., $\mathbf{u} \in \mathbb{R}^d$) and character-level (e.g., $\mathbf{h} \in \mathbb{R}^{k'}$) encodings of the word u, we concatenate them to generate the hybrid word embedding $\mathbf{e}$:

$$\mathbf{e} = \mathbf{u} \oplus \mathbf{h}, \tag{3.6}$$

where $\mathbf{e} \in \mathbb{R}^{d+k'}$ and we use $\mathbf{E}$ to represent the set of hybrid word embeddings.

3.2 Reading Product-related Questions

We combine a Bi-LSTM network [5, 20] with a single-time attention mechanism [1] as the module for machine reading. In this part, we use this module first to "read" the product-related question q. Suppose that the length of question q is n, then we map each word q_i $(i \in [1, n])$ with the set of hybrid word embeddings $\mathbf{E}$ to obtain its embedding $\mathbf{q}_i$. With the help of the Bi-LSTM as follows,

$$\overrightarrow{\mathbf{s}_i^q} = \overrightarrow{\text{LSTM}}(\mathbf{q}_i, \overrightarrow{\mathbf{s}_{i-1}^q}) \tag{3.7}$$

and

$$\overleftarrow{\mathbf{s}_i^q} = \overleftarrow{\text{LSTM}}(\mathbf{q}_i, \overleftarrow{\mathbf{s}_{i+1}^q}), \tag{3.8}$$

we can encode the past and future information (i.e., $\overrightarrow{\mathbf{s}_i^q} \in \mathbb{R}^d$ and $\overleftarrow{\mathbf{s}_i^q} \in \mathbb{R}^d$) of the whole question into each word q_i by

$$\mathbf{s}_i^q = \overrightarrow{\mathbf{s}_i^q} \oplus \overleftarrow{\mathbf{s}_i^q}, \tag{3.9}$$

where $\mathbf{s}_i^q \in \mathbb{R}^{2d}$ stands for the contextual embedding of the word q_i.

To re-weight the contribution of each contextual embedding to the whole question q, the machine reading module adopts a single-time attention mechanism formulated as follows,

$$\alpha_i^q = \text{softmax}(\mathbf{v}_q^{\mathrm{T}} \tanh(\mathbf{W}^q \mathbf{s}_i^q)), \tag{3.10}$$

in which $\mathbf{W}^q \in \mathbb{R}^{2d \times 2d}$ and $\mathbf{v}_q \in \mathbb{R}^{2d}$ are tunable parameters. We use $\mathbf{x}^q$ to denote the distributed representations of the product-related question q and it equals to the weighted sum of the contextual embeddings:

$$\mathbf{x}^q = \sum_i \alpha_i^q \mathbf{s}_i^q. \tag{3.11}$$

3.3 Reading Multiple Reviews with Questions

Suppose that there are m customer reviews associated with the product-related question q. We use r^i $(i \in [1, m])$ to represent the i-th customer review. The same machine reading module with different parameters could be exploited to obtain the distributed representations $\mathbf{z}^{r^i}$ of i-th customer review r^i. However, for the sake of making $\mathbf{z}^{r^i}$ fully aware of the product-related question q, we feed each hybrid embedding of word $\mathbf{r}_j^i$ in the review r^i together with the product-related question embedding $\mathbf{x}^q$ in each step in the Bi-LSTM network as follows,

$$\overrightarrow{\mathbf{s}_j^{r^i}} = \overrightarrow{\text{LSTM}}(\mathbf{r}_j^i \oplus \mathbf{x}^q, \overrightarrow{\mathbf{s}_{j-1}^{r^i}}), \tag{3.12}$$

$$\overleftarrow{\mathbf{s}_j^{r^i}} = \overleftarrow{\text{LSTM}}(\mathbf{r}_j^i \oplus \mathbf{x}^q, \overleftarrow{\mathbf{s}_{j+1}^{r^i}}), \tag{3.13}$$

and

$$\mathbf{s}_j^{r^i} = \overrightarrow{\mathbf{s}_j^{r^i}} \oplus \overleftarrow{\mathbf{s}_j^{r^i}}. \tag{3.14}$$

The product-aware review representation (i.e., $\mathbf{z}^{r^i}$) of the i-th review r^i can be obtained via:

$$\alpha_j^{r^i} = \mathrm{softmax}(\mathbf{v}_{r^i}^{\mathrm{T}} \tanh(\mathbf{W}^{r^i} \mathbf{s}_j^{r^i})) \tag{3.15}$$

and

$$\mathbf{z}^{r^i} = \sum_j \alpha_j^{r^i} \mathbf{s}_j^{r^i}. \tag{3.16}$$

3.4 Neural Voting for Answer Prediction

Given that not all customer reviews are useful to help predict the correct answer a to the product-related question q, we need to design a function which can vote for the significant evidence for answer prediction in terms of the multiple product-related review representations. The intuition comes from the idea that a customer review may benefit from other reviews represented by similar embeddings and the review receiving a majority vote tends to contain the supportive evidence for predicting the correct answer. Therefore, we use $c_{i,j}$ to denote the similarity between the i-th and the j-th reviews, and $c_{i,j}$ is defined as:

$$c_{i,j} = \begin{cases} 0, & \text{if } i = j \\ \mathbf{z}^{r^i} \cdot \mathbf{z}^{r^j}, & \text{if } i \neq j \end{cases} \tag{3.17}$$

Given the i-th review, the distribution of supporting scores from the other reviews is

$$\beta_{i,j} = \frac{\exp(c_{i,j})}{\sum_{t=1}^{m} \exp(c_{i,t})}, \tag{3.18}$$

where m is the number of reviews. We use $\widetilde{\mathbf{z}^{r^i}}$ to denote another embedding for the i-th review which collects the supportive information from the other reviews based on the supporting scores $\beta_{i,j}$:

$$\widetilde{\mathbf{z}^{r^i}} = \sum_{j=1}^{m} \beta_{i,j} \mathbf{z}^{r^j}. \tag{3.19}$$

Hence, we can represent a product-aware review from the perspectives of $\mathbf{z}^{r^i}$, $\widetilde{\mathbf{z}^{r^i}}$ and $\mathbf{z}^{r^i} \odot \widetilde{\mathbf{z}^{r^i}}$.

In other to select the most significant feature of each review, we apply the max pooling strategy into the vector $[\mathbf{z}^{r^i}, \widetilde{\mathbf{z}^{r^i}}, \mathbf{z}^{r^i} \odot \widetilde{\mathbf{z}^{r^i}}]$ and finally generate a feature vector $\mathbf{g} \in \mathbb{R}^m$ to represent the m customer reviews as the evidence to predict the answer via a logistic regression classifier.

4 EXPERIMENTS

4.1 Dataset

Thanks to [11] who started a crawl which contains various information about the products in Amazon.com. Among the data collections, we notice that three groups could be used to build a useful dataset for PQA: the product-related QA data, the meta-data (objective descriptions) of products and a vast collection of customer reviews (subjective opinions). Since each of them is classified into individual product categories, what we need to do is to align these subsets first by the categories, and then to associate both objective descriptions and subjective opinions with product-related questions by the common product id, which is denoted by "ASIN"[3] in the three data collections.

So far we have built an extensive dataset which contains 224,382 labeled Yes/No product-related questions, 124,074 products and 25,604,447 customer reviews spreading over 9 categories in `Amazon.com`. In order to fairly assess the performance of successive approaches on automated PQA, we divide the whole dataset into two parts: i.e., randomly sampling 80% data instances as the training set and leaving 20% data instances for held-out testing. Table 1 elaborates the statistics of the training and test sets. It is also a real-world dataset in terms of the two perspectives as follows:

- We sort the numbers of product-related questions over 9 categories in descending order, which are shown by Figure 3. It indicates that the distribution of the numbers of the product-related questions follows the Zipf's Law [15].

- In the training set, 70.63% product-related questions have the positive answer, and the distribution of the answers in the training set is almost identical with the distribution of the answers (70.49% are positive answers) in the test set. These facts show that our dataset is imbalanced as a part of real-world e-commercial data and applicable to evaluate various learning models for PQA since the training and test sets have almost the same distribution.

4.2 Evaluation Metrics

- **Accuracy@50%**: This metric is widely adopted by all the research on PQA [12, 21] and generally provides better results than accuracy, as it ignores 50% test questions that are hardly answered (with lower confidence scores). In reality, the metric of

[3] It is an Amazon identification number that can usually be found by the page-in product descriptions.

Table 1: The number of questions (Ques.), products (Prod.), customer reviews (Rev.), positive responses (Y-Ans.), and negative responses (N-Ans.) for each category in the training and test sets, respectively.

CATEGORY	**TRAINING SET / TEST SET**				
	# (Ques.)	**# (Prod.)**	**# (Rev.)**	**# (Y-Ans.)**	**# (N-Ans.)**
Beauty	6568 / 1641	3730 / 1402	737,552 / 179,778	4397 / 1125	2171 / 516
Cell Phones and Accessories	18,658 / 4664	7561 / 3571	3,282,126 / 791,651	13,543 / 3373	5115 / 1291
Electronics	64,138 / 16,034	28,470 / 12,754	8,120,622 / 2,052,472	46,753 / 11,643	17,385 / 4391
Health and Personal Care	11,543 / 2885	6578 / 2489	1,565,222 / 410,065	7614 / 1966	3929 / 919
Home and Kitchen	26,459 / 6614	14,787 / 5602	2,857,778 / 729,078	17,824 / 4419	8635 / 2195
Musical Instruments	4706 / 1176	2268 / 954	290,723 / 81,878	3450 / 848	1256 / 328
Office Products	8444 / 2111	4028 / 1708	823,001 / 192,492	5879 / 1494	2565 / 617
Sports and Outdoors	21,328 / 5332	10,799 / 4392	1,564,201 / 416,627	15,162 / 3793	6166 / 1539
Tools and Home Improvement	17,665 / 4416	9281 / 3700	1,205,038 / 304,143	12,161 / 2971	5504 / 1445
TOTAL	179,509 / 44,873	87,502 / 36,572	20,446,263 / 5,158,184	126,783 / 31,632	52,726 / 13,241

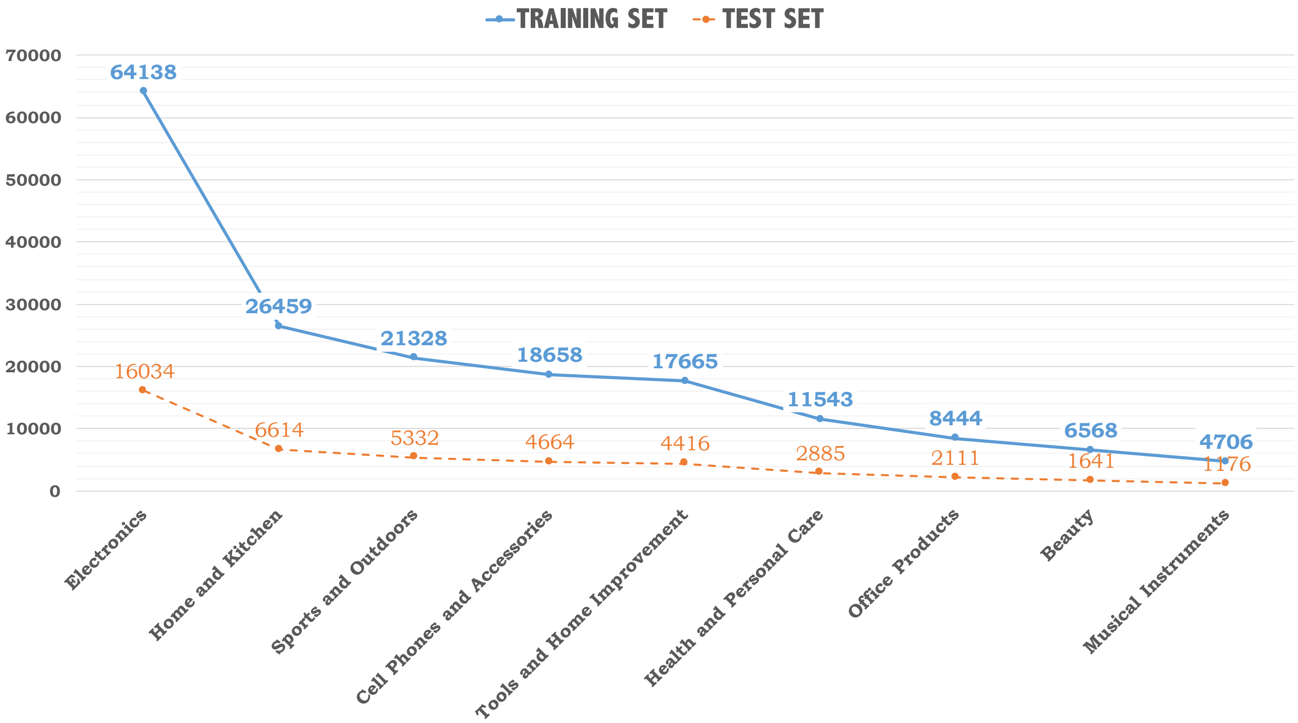


Figure 3: The number of product-related questions over 9 categories in the training and test sets.

accuracy@$x\%$ (where x ranges from 0 to 100) is more practical to assess the performance of a classifier in a real-world setting where we usually surface the top answers with higher confidence scores and leave the product-related questions that machines believe hardly to be answered for customer service agents. The metric of accuracy@$x\%$ is also capable of helping to estimate a desirable threshold of confidence score which can fulfill the requirement of a PQA system reaching a certain accuracy.

- **AUROC**: We also employ another metric: the Area Under the curve of the Receiver Operating Characteristic (AUROC), to enrich the assessment of automated PQA methods. It is due to the fact that our dataset is imbalanced. Classification models are prone to learn the bias of imbalanced data distribution. AUROC is a more stringent metric than accuracy@50% as it is sensitive to the bias and will punish the inferior models.

4.3 Experimental Results

To obtain the overall performance of each method on PQA, we adopt two ways of calculating the average results over 9 product categories in our dataset, i.e., macro average and micro average. The macro-average ignores the proportion of product-related questions in each category and simply calculates the arithmetic mean of 9 categories. The micro-average, on the other hand, takes the proportion of product-related questions in each category into account and provides the weighted mean of 9 categories. When a new product-related question comes, it is more likely to fall into the majority categories such as “Electronics” and “Home and Kitchen” shown in Figure 3. Therefore, we regard the *micro-average* as the primary way to compute the reliable results for method comparison.

Table 2 shows the experimental results of COR-L [12], MOQA [12], AAP [21], and NRA, measured by Accuracy@50% on the test set, spreading over 9 categories in Amazon.com. From the results shown by Table 2, we can easily tell that NRA consistently outperforms the other modern approaches reaching 77.35% in accuracy@50% with a significant improvement by **5.70%** on micro average. Among the other baseline methods, AAP [21] shows its superiority on 5 subsets with larger volume, including *Electronics*, *Home and Kitchen*, *Sports and Outdoors*, *Cell Phones and Accessories*, and *Office Products*. It is likely due to the latent aspects acquired by the AutoEncoder [4]. Our NRA adopts the neural architecture, and the experimental results confirm that it can fully take advantage of large-scale datasets to achieve surpassing results on PQA.

Table 3 presents the experimental results of COR-L [12], MOQA [12], AAP [21] and NRA, measured by the metric of AUROC on the test set, over the same 9 categories in Amazon.com. The implicit goal of AUROC is to deal with the skewed sample distribution, and the value of AUROC will be punished if the predicting answers overfit to a single class. The results in Table 3 show that NRA could handle the imbalanced data much better than the baseline approaches. Specifically speaking, NRA achieves 61.76% AUROC on micro average, much better than the modern approach AAP [21] (53.30%). The absolute percentage growth is **8.46%** and the relative percentage growth is 15.87%. Even for AAP [21] who leverages a neural component (AutoEncoder [4]), it performs much better than the COR-L [12] and MOQA [12] on the 7 of 9 categories in the dataset.

5 CONCLUSION AND FUTURE WORK

This paper studies the task of teaching machines to automatically answer product-related questions, i.e., PQA, on e-commerce sites. We address the problem from the perspective of machine reading comprehension. Specifically, we contribute a *neural reading architecture* (NRA) for PQA, which is fed by the raw text of product-related question and customer reviews, to produce multiple question-aware review embeddings and to synthesize the embeddings as the evidence to predict the final answer. NRA is an end-to-end trainable neural network for PQA that can directly learn the task-specific feature representations in a supervised manner. Compared with the other mainstream approaches which either leverage heuristic features or acquire hidden features via an unsupervised manner, NRA achieves the state-of-the-art performance on our dataset, surpassing the modern approaches with a significant improvement by **5.70%** accuracy@50% and **8.46%** AUROC.

Future work could aim at extending the scope of research on PQA from the perspectives of data and methodology: 1) We should enlarge the coverage of our dataset regarding its volume and the type of product-related questions it contains. It is because there are still many product-related questions looking forward to an open-ended answer besides the Yes/No-type questions; 2) To persistently make breakthroughs on our dataset of PQA, we should keep exploring more effective and efficient approaches on the task of PQA.

Acknowledgments

This work was supported by the Joint Post-Doctoral Programme of Baidu Inc. and Tsinghua University.

Table 2: Accuracy@50% of COR-L [12], MOQA [12], AAP [21], and our neural reading architecture (NRA) on the test set (a.c.: absolute change; r.c.: relative change; *_italic fonts_: the best performance among the three baseline approaches; **bold fonts**: the state-of-the-art performance of all the approaches).

CATEGORY	ACCURACY@50%			
	COR-L	MOQA	AAP	NRA
Beauty	*_73.78%_	72.56%	71.46%	**79.14%** [a.c.: 5.36% ↑; r.c.: 7.26% ↑]
Cell Phones and Accessories	72.51%	72.64%	*_72.85%_	**80.18%** [a.c.: 7.33% ↑; r.c.: 10.06% ↑]
Electronics	71.90%	73.26%	*_74.26%_	**79.71%** [a.c.: 5.45% ↑; r.c.: 7.34% ↑]
Health and Personal Care	67.05%	*_68.93%_	68.58%	**72.05%** [a.c.: 3.12% ↑; r.c.: 4.53% ↑]
Home and Kitchen	65.73%	65.98%	*_66.91%_	**73.45%** [a.c.: 6.54% ↑; r.c.: 9.77% ↑]
Musical Instruments	73.97%	*_75.68%_	74.65%	**79.08%** [a.c.: 3.40% ↑; r.c.: 4.49% ↑]
Office Products	72.98%	72.22%	*_73.36%_	**78.19%** [a.c.: 4.83% ↑; r.c.: 6.58% ↑]
Sports and Outdoors	69.24%	72.65%	*_71.23%_	**77.49%** [a.c.: 4.84% ↑; r.c.: 6.66% ↑]
Tools and Home Improvement	68.02%	*_69.74%_	69.02%	**73.41%** [a.c.: 3.67% ↑; r.c.: 5.26% ↑]
MACRO AVERAGE	70.58%	*_71.52%_	71.37%	**76.97%** [a.c.: 5.45% ↑; r.c.: 7.62% ↑]
MICRO AVERAGE (Primary)	70.22%	71.41%	*_71.65%_	**77.35%** [a.c.: 5.70% ↑; r.c.: 7.96% ↑]

Table 3: AUROC of COR-L [12] , MOQA [12], AAP [21], and our neural reading architecture (NRA) on the test set (a.c.: absolute change; r.c.: relative change;*_italic fonts_: the best performance among the three baseline approaches; **bold fonts**: the state-of-the-art performance of all the approaches).

CATEGORY	AUROC			
	COR-L	MOQA	AAP	NRA
Beauty	*_58.69%_	57.55%	53.27%	**71.23%** [a.c.: 12.54% ↑; r.c.: 21.37% ↑]
Cell Phones and Accessories	50.82%	50.94%	*_51.79%_	**64.61%** [a.c.: 12.82% ↑; r.c.: 24.75% ↑]
Electronics	52.32%	51.19%	*_53.54%_	**62.37%** [a.c.: 8.83% ↑; r.c.: 16.49% ↑]
Health and Personal Care	51.57%	51.63%	*_52.82%_	**56.71%** [a.c.: 3.89% ↑; r.c.: 7.36% ↑]
Home and Kitchen	51.37%	51.53%	*_51.94%_	**60.09%** [a.c.: 8.15% ↑; r.c.: 15.69% ↑]
Musical Instruments	55.21%	55.43%	*_56.29%_	**61.42%** [a.c.: 5.13% ↑; r.c.: 9.11% ↑]
Office Products	53.88%	54.06%	*_55.70%_	**61.84%** [a.c.: 6.14% ↑; r.c.: 11.02% ↑]
Sports and Outdoors	52.02%	*_53.46%_	53.05%	**60.92%** [a.c.: 7.46% ↑; r.c.: 13.95% ↑]
Tools and Home Improvement	52.59%	54.45%	*_54.79%_	**59.92%** [a.c.: 5.13% ↑; r.c.: 9.36% ↑]
MACRO AVERAGE	53.16%	53.36%	*_53.69%_	**62.12%** [a.c.: 8.43% ↑; r.c.: 15.70% ↑]
MICRO AVERAGE (Primary)	52.35%	52.31%	*_53.30%_	**61.76%** [a.c.: 8.46% ↑; r.c.: 15.87% ↑]

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] Miao Fan, Yue Feng, Mingming Sun, Ping Li, Haifeng Wang, and Jianmin Wang. Multi-task neural learning architecture for end-to-end identification of helpful reviews. In *IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, Barcelona, Spain, August 28-31, 2018*, pages 343–350, 2018.

[3] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701, 2015.

[4] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[7] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

[8] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[9] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[10] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 7–16, 2011.

[11] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2015.

[12] Julian McAuley and Alex Yang. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 625–635, 2016.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[14] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814, 2010.

[15] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[16] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016.*, 2016.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[19] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[20] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[21] Qian Yu and Wai Lam. 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 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 691–699, 2018.

[22] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015.