Effectiveness of Data Augmentation to Identify Relevant Reviews for Product Question Answering

Kalyani Roy
Indian Institute of Technology
Kharagpur, India
kroy@iitkgp.ac.in

Avani Goel
Indian Institute of Technology
Kharagpur, India
avanigoel89@gmail.com

Pawan Goyal
Indian Institute of Technology
Kharagpur, India
pawang@cse.iitkgp.ac.in

ABSTRACT

With the rapid growth of e-commerce and an increasing number of questions posted on the Question Answer (QA) platforms of e-commerce websites, there is a need for providing automated answers to questions. In this paper, we use transformer-based review ranking models which provide a ranked list of reviews as a potential answer to a new question. Since no explicit training data is available, we exploit the product reviews along with available QA pairs to learn a relevance function between a question and a review sentence. Further, we present a data augmentation technique by fine-tuning the T5 model to generate new questions from customer reviews by considering the summary of the review as an answer and the review as the document. We conduct experiments on a real-world dataset from three categories in Amazon.com. To assess the performance of the models, we use the annotated question review dataset from RIKER [13]. Experimental results show that Deberta-RR model with the augmentation technique outperforms the current state-of-the-art model by 5.84%, 4.38%, 3.96%, and 2.96% on average in nDCG@1, nDCG@3, nDCG@5, and nDCG@10, respectively.

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*; *Question answering*; • **Applied computing** → *Online shopping*.

KEYWORDS

Product Question Answering, Review Ranking, Data Augmentation

ACM Reference Format:
Kalyani Roy, Avani Goel, and Pawan Goyal. 2022. Effectiveness of Data Augmentation to Identify Relevant Reviews for Product Question Answering. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524261

1 INTRODUCTION

There is a rising interest in people for online shopping. Before making any purchase decisions, people generally have some queries regarding that product. So, most e-commerce websites provide a Question-Answer (QA) platform for each product, wherein customers can post their questions. Sellers cannot answer every question due to the large volume of questions posted. Other consumers who have already purchased that product can answer that question, but in this case, the customer needs to wait to get the reply. On the other hand, the existing reviews of the product may already contain the answer, but it is infeasible to go through these enormous reviews. If we can provide an instant response to the question, it will considerably enhance the customer's online shopping experience. A potential solution would be to automatically identify the answer to the new question from the already posted reviews.

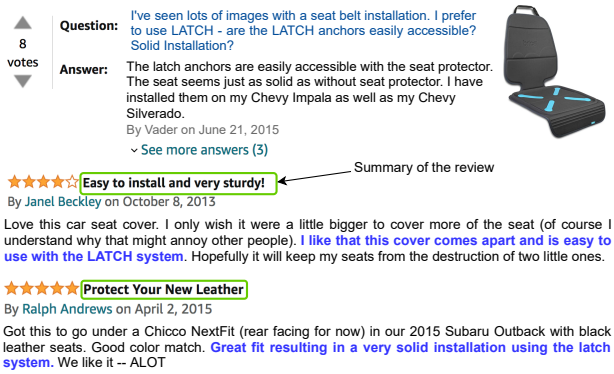

Figure 1: Snapshot of the product page of a car seat protector with a question, its answer, and two relevant reviews. The plausible answers to the question from the reviews are highlighted in blue. The texts inside the green rectangles are the summaries of the reviews.

Figure 1 depicts a snapshot of a product page for a car seat protector. It shows a question-answer pair from the QA section and two relevant reviews of the product. Each review also contains a summary written by the customer and it is shown within a green rectangle. Consider the question, *"Are the LATCH anchors easily accessible? Solid Installation?"*. We can find the answers to this question in the reviews posted earlier. We observe that the review snippet from the first review *" I like that this cover comes apart and is easy to use with the LATCH system"* and from the second review *"Great fit resulting in a very solid installation using the latch system."* can answer the question. So, these relevant review sentences can serve as responses to the newly posted question.

Previous works on Product-related Question Answering (PQA) [6] rely on using traditional word-based relevance function and manual feature engineering to rank reviews. Yu et al. [11] uses existing QA pairs as distant supervision and Positional Language Model (PLM)

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524261

to rank the reviews. Chen et al. [3] formulates PQA as a classification problem to predict whether a question-review pair is correct or not. Zhao et al. [13] proposes interpretable PQA model RIKER that uses rich keyword representations to answer product related questions. Transformer-based models like BERT [4], T5 [8], DeBERTa [5] are applied to wide range of NLP tasks. Zhang et al. [12] presents a BERT-based model that discovers relevant reviews for PQA by leveraging the question-answer pairs to learn the relevance between question and reviews. However, it does not utilize the large set of reviews to create more coherent question-answer data that could be used in addition to the available data for effective training. Recently, various data augmentation (DA) techniques [1, 10] have shown improved performance in many NLP tasks like question-answering, summarization, sequence tagging, etc.

Motivated by the success of Zhang et al. [12] on learning relevance between question and reviews via the question-answer pairs, and the DA methods in effectively augmenting the available training dataset, we attempt to improve the review ranking task with a) better transformer-based model and b) DA technique using T5. More specifically, we utilize the question answers related to a product to learn a relevance function between a question and a review sentence, while using the recently proposed DeBERTa [5] pretrained model. We also present a DA technique by fine-tuning the T5 model to generate new questions from customer reviews by considering the summary of the review as an answer and the review as the document.

To assess the performance of the models, we use the annotated question review dataset from RIKER [13]. We observe that (i) DeBERTa improves over BERT by 2.64%, 2.03%, 2.38%, and 1.06% on average in nDCG@1, nDCG@3, nDCG@5, and nDCG@10, respectively, (ii) Our Deberta-RR model trained with the augmented data outperforms the current state-of-the-art model by 5.84%, 4.38%, 3.96%, and 2.96% on average in nDCG@1, nDCG@3, nDCG@5, and nDCG@10, respectively.

2 DATASET

We use the Amazon Question Answer (QA) dataset [6] and the Amazon review dataset [7] for our experiment. The Amazon QA dataset contains questions with multiple customer written answers and a helpfulness rating for each of the answers. The Amazon Review dataset includes users' reviews along with a summary of the review and overall rating of the product. The unique Product IDs in each dataset are used to align the question with its reviews. We take three product categories, namely, *Baby*, *Tools & Home Improvement*, and *Patio Lawn & Garden*, for our experiment.

2.1 Synthetic Training Data Creation

Due to lack of a labeled dataset which contains question along with its relevant reviews that can be used for training, we resort to synthetic training setting similar to Zhang et al. [12], where each training data instance consists of a question Q, a positive answer A_p, a negative answer A_n, and a list of review sentences R. Since short answers do not convey much information, we filter out the answers having less than five tokens (without counting stop-words and yes/no). Because of this filtering, many questions did not have any answers remaining. Further, we remove the products with less than two questions. We assign each answer a score where the score is the ratio of the helpful votes to the total votes. For every question Q, we select the positive response A_p based upon this score. We take the response with the highest score (> 0.5) as the positive answer, and in the case of a tie, we choose the one with the highest number of total votes. When there is no vote for any answer of a question, we randomly select any of its answers as the gold answer. We randomly select an answer from a different question of the same product as the negative answer A_n. The dataset contains thousands of reviews for each product, and the entire review may not be relevant for a particular question. So, we split reviews into a set of review sentences. We discard the sentences having less than five tokens. For each question Q, we take a positive answer A_p, a negative answer A_n, and we train a classifier with BERT that predicts whether the answer is relevant to that question or not. Initially, we choose the top 100 review sentences using BM25. We refine this list with the trained classifier, and we take the top 10 reviews as the set R for each question. Table 1 shows the train and validation split for each vertical. We will refer to this dataset as QAR dataset.

Table 1: Statistics of the QAR dataset and the additional train data in the QAR-aug dataset

Category	Train	Valid	Augmented
Baby	9,189	1,148	2,756
Tools & Home	29,764	3,720	8,929
Patio Lawn & Garden	17,784	2,223	5,335

2.2 Data Augmentation

While the synthetic training data leverages the available question-answer pairs, these may not be sufficient for many verticals. In most of the e-commerce platforms like Amazon, eBay, Flipkart etc., customers write review and a short summary of that review. As these review summaries are not explored in PQA, we further attempt to augment the synthetic training data by utilizing the large number of reviews available to us. We consider the review of a product as "context" and the summary as the "answer". We attempt to generate a question, given a context and an answer, by utilizing Text-to-Text Transfer Transformer (T5) [8].

We take the T5 model that is already fine-tuned on the question generation task on the SQuAD dataset and we will refer to this model as FT1. We use FT1 to further fine-tune the model on the SubjQA dataset [2] and we denote this model as FT2. We select four categories, namely, *Electronics*, *Movies*, *Books*, *Grocery* from the SubjQA dataset since those belong to the Amazon dataset. We drop those questions that do not have the answer contained within the review. We employ both FT1 and FT2 to generate new questions. We discard the questions that have words such as "I", "me", "mine", "we", "our" etc. in them, i.e., the questions containing possessive pronouns since such questions do not ask about the characteristics of the product. Also, we drop the generated questions not ending in '?'. Table 2 lists two examples of generated questions. The first example is generated using FT1 and the second example is generated using FT2. The generated question from these models is our question Q,

Table 2: Example of generated questions with FT1 and FT2.

Review: Since it's a long hose there was a bit of kink that I didn't notice. Turned the water on and it instantly blew a large hole in the hose, rendering it useless. The plastic is pretty thin, so be warned to carefully check for any kinks. **Summary:** On 3rd time used it had a huge hole in it. **Gen. Que.:** What was the problem with the hose?
Review: This is a very cute decal for my baby girl's room, but it is not as tall as the picture depicts. I would suggest putting it behind the baby bed or dresser to hide how short it really is. **Summary:** Cute, but not as big as in the photo. **Gen. Que.:** How is the decal?

we split the review into sentences to form the review list R, we take the review summary as our positive answer A_p, and we get the negative answer A_n by randomly selecting any other question's gold answer. The last column of the Table 1 shows the statistics of the augmented data points for each vertical. We combine this generated data with the QAR dataset and we denote it as the QAR-aug dataset. [1]

3 BASE MODEL ARCHITECTURE

Similar to [12], our base model simultaneously learns relevance functions between 'question and review' and 'review and answer' such that at the time of inference, the learned relevance function between 'question and review' can be used to rank the reviews related to a question. Figure 2 illustrates the model's architecture.

For each data instance as prepared in Section 2 we have a question Q, a list of review sentences R, a gold answer A_p, and a negative answer A_n. We concatenate each reviews r_i with Q, A_p,and A_n as Qr_i, A_pr_i, and A_nr_i, respectively. After tokenizing the sentences, we pass it through a transformer layer to get the sentence representation from the CLS token.

Then, we use linear layers followed by non-linear activation functions to get $P(r_i|Q)$, $P(A_p|r_i)$ and $P(A_n|r_i)$ as follows:

$$P(r_i|Q) = softmax(W^T\ Tran(Qr_i)) \tag{1}$$

$$P(A_p|r_i) = \sigma(W^T\ Tran(A_pr_i)) \tag{2}$$

$$P(A_n|r_i) = \sigma(W^T\ Tran(A_nr_i)) \tag{3}$$

where W is a learnable matrix, and $Tran(XY)$ denotes the representation of the paired sentences X and Y using transformer. Now, $P(A_p|Q)$ and $P(A_n|Q)$ are computed as follows:

$$P(A_p|Q) = \sum_{r_i} P(A_p|r_i)\ P(r_i|Q) \tag{4}$$

$$P(A_n|Q) = \sum_{r_i} P(A_n|r_i)\ P(r_i|Q) \tag{5}$$

The objective of the model is to rank gold answer higher than the negative answer. For this, we use the margin ranking loss function.

$$loss = max(0, P(A_p|Q) - P(A_n|Q) - \delta) \tag{6}$$

Equation 6 signifies that $P(A_p|Q)$ should be greater than $P(A_n|Q)$ at least by a difference of δ. This way, the model learns the function $P(r|Q)$ that can be used at inference time to rank the review sentences.

[1] https://github.com/kalyani-roy/DARR.

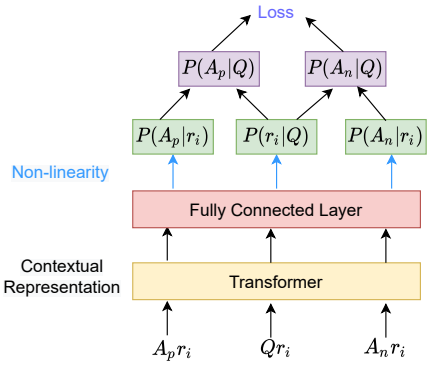

Figure 2: The architecture of the model. $r_i \in R, |R| = 10$. After the non-linear layer, the reviews are combined according to Equ. 4 and Equ. 5 to get the relevance scores of the positive and the negative answers with the question.

We use two transformer-based models - BERT and DeBERTa, and we denote the two models with this architecture as **Bert R**elevant **R**eview (Bert-RR) and **Deberta R**elevant **R**eview (Deberta-RR), respectively.

4 EXPERIMENTAL SETUP

Given a question Q about a product P and a set of reviews $R = \{r_1, r_2, r_3, \dots\}$ for that question, our aim is to provide a ranked list of reviews $R' = \{r'_1, r'_2, r'_3, \dots\}$ where r'_i are ranked in order of decreasing relevance with question Q.

4.1 Evaluation Metrics

For evaluating the models, we use the annotated dataset from RIKER [13]. It has 40 annotated questions per vertical. There are on average 17 reviews as answer candidates for each question and each question-review pair is annotated by 3 annotators, based on relevance to the question, which can be 2/1/0 pertained to being relevant, partly relevant, and irrelevant. Consistent with the previous approaches [11, 13], we use the Normalized Discounted Cumulative Gain (nDCG) as our evaluation metric. We show nDCG@k, $k \in \{1, 3, 5, 10\}$, averaged across the three annotators, for all the models.

4.2 Competing Models

We evaluate the proposed framework by taking various traditional and state-of-the-art methods as competing models. **(i) BM25 [9] :** It is a popular retrieval model for ranking candidate answers of a question. **(ii) RIKER [13] :** It is an interpretable PQA model. It improves keyword-based search by learning rich keyword representations for questions. [2] **(iii) Bert-RR [12] :** It is a state-of-the-art model for discovering relevant reviews of a question, as described in Section 3. **(iv) Deberta-RR :** It is similar to Bert-RR, but instead of BERT, it uses the CLS token of the DeBERTa model for contextual

[2] Since the available code repository for RIKER is missing information required to run their model on our training datasets, and the same Amazon dataset is used to create the training dataset, we compare the models with the reported numbers in RIKER [13]. RIKER reports only nDCG@10.

Table 3: Performance of all the models in three categories. The cross and the checkmark symbols indicate that the model is trained with the QAR dataset and the augmented dataset QAR-aug, respectively.

	nDCG	BM25	RIKER [13]	Bert-RR [12]		Deberta-RR	
	(%)	-	-	×	✓	×	✓
Baby	@1	42.08	-	55.00	62.08	59.17	**64.16**
	@3	38.64	-	57.95	61.32	60.70	**63.28**
	@5	44.34	-	61.60	63.79	64.71	**66.85**
	@10	52.76	64.80	66.95	69.31	67.51	**70.78**
Tools & Home	@1	37.50	-	40.83	**46.25**	41.25	44.17
	@3	38.44	-	45.32	**45.82**	44.14	45.46
	@5	38.36	-	45.76	46.60	46.68	**46.90**
	@10	43.81	45.12	49.67	50.69	50.13	**50.75**
Patio Lawn & Garden	@1	31.25	-	45.00	49.16	48.33	**50.00**
	@3	34.70	-	44.46	47.13	48.96	**52.11**
	@5	36.40	-	46.88	50.52	49.99	**52.35**
	@10	44.04	55.91	55.01	58.30	57.17	**58.99**
Average	@1	36.94	-	46.94	52.50	49.58	**52.78**
	@3	37.26	-	49.24	51.42	51.27	**53.62**
	@5	39.70	-	51.41	53.64	53.79	**55.37**
	@10	46.87	55.28	57.21	59.43	58.27	**60.17**

representation of sentences. We first train Bert-RR and Deberta-RR with QAR dataset. To further test the effectiveness of augmented dataset, these are trained with the QAR-aug dataset.

4.3 Implementation Details

We use pytorch to implement the models. To train the initial classifier with Bert in Section 2.1, we use batch size of 64, Adam optimizer with learning rate 2e-5. For augmenting the dataset, we fine-tune the FT2 model for 10 epochs with learning rate 1e-4, batch size 32. To train the Bert-RR and Deberta-RR models, we use a maximum sequence length of 64, batch size 3, dropout of 0.3, and Adam optimizer with a learning rate of 2e-5. We empirically fix δ to 0.3. All our experiments are run on Tesla P100-PCIE 16GB GPU.

5 RESULTS

Table 3 summarizes the review ranking results among three product categories in nDCG@1, nDCG@3, nDCG@5, and nDCG@10 scores. The Average group shows the average score across all the verticals for each method at different nDCG@k. The cross and the checkmark symbols indicate that the models are trained with the QAR dataset and the QAR-aug dataset, respectively. Compared to the basic BM25 model, the deep learning models generally provide stronger baselines for this task. RIKER [13], Bert-RR [12], and Deberta-RR perform better than the unsupervised BM25. We achieve better performance than RIKER [13] for all the three verticals. Without any augmentation of data, Deberta-RR outperforms Bert-RR by 2.64%, 2.03%, 2.38%, and 1.06% on average in nDCG@1, nDCG@3, nDCG@5, and nDCG@10, respectively.

In all three categories, the DA technique improves performance. The DA method results in performance gains of 5.56%, 2.18%, 2.23%, and 2.22% in Bert-RR, and performance gains of 3.20%, 2.35%, 1.58%, and 1.90% in Deberta-RR. With the DA method, Deberta-RR performs the best in *Baby* and *Patio Lawn & Garden* verticals. In *Tools & Home Improvement*, Bert-RR with augmentation outperforms the other methods in nDCG@1 and nDCG@3, but for nDCG@5 and nDCG@10, Deberta-RR with augmentation is the best performing model. Compared to the previous state-of-the-art method Bert-RR [12], Deberta-RR with DA shows large improvements by 5.84%, 4.38%, 3.96%, and 2.96% on average in nDCG@1, nDCG@3, nDCG@5, and nDCG@10, respectively. The overall result shows that the augmentation technique yields superior performance to the baseline methods in all product categories.

6 CONCLUSION

We utilize transformer-based models to provide relevant reviews to a new question by exploiting the question-answer collections and review collections. Experimental results show substantial improvements over the existing approaches using the data augmentation technique. Retrieving relevant reviews is essential to generate natural answers to the product questions. It would be interesting to check whether the reviews retrieved by Deberta-RR with augmented data can improve the performance of the generative models.

REFERENCES

[1] Akari Asai and Hannaneh Hajishirzi. 2020. Logic-Guided Data Augmentation and Regularization for Consistent Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[2] Johannes Bjerva, Nikita Bhutani, Behzad Golshan, Wang-Chiew Tan, and Isabelle Augenstein. 2020. SubjQA: A Dataset for Subjectivity and Review Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 5480–5494.

[3] Long Chen, Ziyu Guan, Wei Zhao, Wanqing Zhao, Xiaopeng Wang, Zhou Zhao, and Huan Sun. 2019. Answer Identification from Product Reviews for User Questions by Multi-Task Attentive Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 45–52.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[5] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv* abs/2006.03654 (2020).

[6] Julian 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*. 625–635.

[7] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. 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)*. 188–197.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[9] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389.

[10] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and Accurate Reading Comprehension by Combining Self-Attention and Convolution. In *International Conference on Learning Representations*, Vol. 2.

[11] Qian Yu, Wai Lam, and Zihao Wang. 2018. Responding E-commerce Product Questions via Exploiting QA Collections and Reviews. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2192–2203.

[12] Shiwei Zhang, Jey Han Lau, Xiuzhen Zhang, Jeffrey Chan, and Cécile Paris. 2019. Discovering Relevant Reviews for Answering Product-Related Queries. In *2019 IEEE International Conference on Data Mining (ICDM)*. 1468–1473.

[13] Jie Zhao, Ziyu Guan, and Huan Sun. 2019. Riker: Mining Rich Keyword Representations for Interpretable Product Question Answering. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1389–1398.