Using Large Pretrained Language Models for Answering User Queries from Product Specifications

Kalyani Roy[1], Smit Shah[1]*, Nithish Pai[2], Jaidam Ramtej[2], Prajit Prashant Nadkarn[2],
Jyotirmoy Banerjee[2], Pawan Goyal[1], and **Surender Kumar[2]**
[1]Indian Institute of Technology Kharagpur
[2]Flipkart
kroy@iitkgp.ac.in, smitsunny11@gmail.com, {nithish.p, jaidam.ramtej, prajit.pn }@
flipkart.com, jyoban@gmail.com, pawang@cse.iitkgp.ac.in, surender.k@flipkart.com

Abstract

While buying a product from the e-commerce websites, customers generally have a plethora of questions. From the perspective of both the e-commerce service provider as well as the customers, there must be an effective question answering system to provide immediate answers to the user queries. While certain questions can only be answered after using the product, there are many questions which can be answered from the product specification itself. Our work takes a first step in this direction by finding out the relevant product specifications, that can help answering the user questions. We propose an approach to automatically create a training dataset for this problem. We utilize recently proposed XLNet and BERT architectures for this problem and find that they provide much better performance than the Siamese model, previously applied for this problem (Lai et al., 2018). Our model gives a good performance even when trained on one vertical and tested across different verticals.

1 Introduction

Product specifications are the attributes of a product. These specifications help a user to easily identify and differentiate products and choose the one that matches certain specifications. There are more than 80 million products across 80+ product categories on Flipkart [1]. The 6 largest categories are - *Mobile*, *AC*, *Backpack*, *Computer*, *Shoes*, and *Watches*. A large fraction of user queries ($\sim 20\%$)[2] can be answered with the specifications. Product specifications would be helpful in providing instant responses to questions newly posed by users about

the corresponding product. Consider a question "What is the fabric of this bag?" This new question can be easily answered by retrieving the specification "Material" as the response. Fig. 1 depicts this scenario.

Most of the recent works on product related queries on e-commerce leverage the product reviews to answer the questions (Gao et al., 2019; Zhao et al., 2019; McAuley and Yang, 2016). Although reviews are a rich source of data, they are also subject to personal experiences. People tend to give many reviews on some products and since it is based upon their personal experience, the opinion is also diverse. This creates a massive volume and range of opinions and thus makes review systems difficult to navigate. Sometimes products do not even have any reviews that can be used to find an answer, also the reviews do not mention the specifications a lot, but mainly deal with the experience. So, there are several reasons why product specifications might be a useful source of information to answer product-related queries which does not involve user experience to find an answer. As the specifications are readily available, users can get the response instantly.

Figure 1: Snapshot of a product with its specifications.

---

* Work done while author was at IIT Kharagpur.

[1]Flipkart Pvt Ltd. is an e-commerce company based in Bangalore, India.

[2]We randomly sampled 1500 questions from all these verticals except *Mobile* and manually annotated them as to whether these can be answered through product specifications.
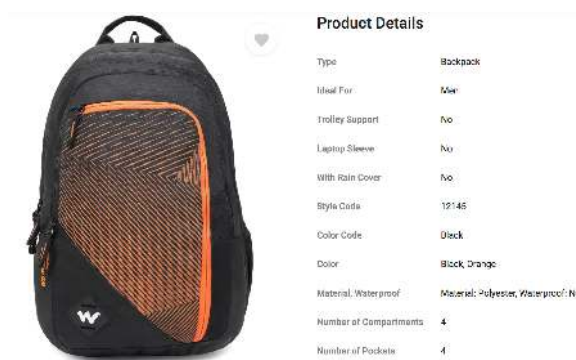
Dataset	Products	Questions	Avg. Specs
Mobile	1,175	260,529	55
AC	300	16,545	35
Backpack	300	16,878	17
Computer	300	93,589	60
Shoes	300	5,812	10
Watches	300	21,392	50

Table 1: Statistics of 6 largest categories.

This paper attempts to retrieve the product specifications that would answer the user queries. While solving this problem, our key contributions are as follows - (i) We demonstrate the success of XLNet on finding product specifications that can help answering product related queries. It beats the baseline Siamese method by $0.14 - 0.31$ points in HIT@1. (ii) We utilize a method to automatically create a large training dataset using a semi-supervised approach, that was used to fine-tune XLNet and other models. (iii) While we trained on *Mobile* vertical, we tested on different verticals, namely, *AC*, *Backpack*, *Computer*, *Shoes*, *Watches*, which show promising results.

2   Background and Related Work

In recent years, e-commerce product question answering (PQA) has received a lot of attention. Yu et al. (2018) present a framework to answer product related questions by retrieving a ranked list of reviews and they employ the Positional Language Model (PLM) to create the training data. Chen et al. (2019) apply a multi-task attentive model to identify plausible answers. Lai et al. (2018) propose a Siamese deep learning model for answering questions regarding product specifications. The model returns a score for a question and specification pair. McAuley and Yang (2016) exploit product reviews for answer prediction via a Mixture of Expert (MoE) model. This MoE model makes use of a review relevance function and an answer prediction function. It assumes that a candidate answer set containing the correct answers is available for answer selection. Cui et al. (2017) develop a chatbot for e-commerce sites known as SuperAgent. SuperAgent considers question answer collections, reviews and specifications when answering questions. It selects the best answer from multiple data sources. Language representation models like BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019) are pre-trained on vast amounts of text and then fine-tuned on task-specific labelled data. The resulting models have achieved

state of the art in many natural language processing tasks including question answering. Dzendzik et al. (2019) employ BERT to answer binary questions by utilizing customer reviews.

In this paper, unlike some of the previous works (Lai et al., 2018; Chen et al., 2019) on PQA that solely rely on human annotators to annotate the training instances, we propose a semi-supervised method to label training data. We leverage the product specifications to answer user queries by using BERT and XLNet.

3   Problem Statement

Here, we formalize the problem of answering user queries from product specifications. Given a question Q about a product P and the list of M specifications $\{s_1, s_2, ..., s_M\}$ of P, our objective is to identify the specification s_i that can help answer Q. Here, we assume that the question is answerable from specifications.

4   Model Architecture

Our goal is to train a classifier that takes a question and a specification as input (e.g., "Color Code Black") and predicts whether the specification is relevant to the question. We take Siamese architecture (Lai et al., 2018) as our baseline method. We fine-tune BERT and XLNet for this classification task.

Siamese: We train a 100-dimensional word2vec embedding on the whole corpus (all questions and specifications as shown in Table 1.) to get the input word representation. In the Siamese model, the question and specification is passed through a Siamese Bi-LSTM layer. Then we use max-pooling on the contextual representations to get the feature vectors of the question and specification. We concatenate the absolute difference and hadamard product of these two feature vectors and feed it to two fully connected layers of dimension 50 and 25, subsequently. Finally, the softmax layer gives the relevance score.

BERT and **XLNet :** The architecture we use for fine-tuning BERT and XLNet is the same. We begin with the pre-trained BERT$_{\text{Base}}$ and XLNet$_{\text{Base}}$ model. To adapt the models for our task, we introduce a fully-connected layer over the final hidden state corresponding to the [CLS] input token. During fine-tuning, we optimize the entire model end-to-end, with the additional softmax classifier parameters $W \in R^{K \times H}$, where H is the dimen-

Dataset	# que-spec pairs	Answer type (in %)		
		Num	Y/N	Other
AC	3693	0.27	0.52	0.21
Backpack	2693	0.29	0.48	0.23
Computer	2718	0.04	0.78	0.18
Shoes	999	0.09	0.49	0.42
Watches	1700	0.17	0.59	0.24

Table 2: Test datasets statistics.

sion of the hidden state vectors and K is the number of classes.

5 Experimental Setup

5.1 Dataset Creation

The Statistics for the 6 largest categories used in this paper are shown in Table 1, containing a snapshot of product details up to January 2019. Except for mobiles, for other domains, 300 products were sampled. As the number of question-specification pairs is huge, manually labelling a sufficiently large dataset is a tedious task. So, we propose a semi-supervised method to create a large training dataset using Dual Embedding Space model (DESM) (Mitra et al., 2016).

Suppose a product P has S specifications and Q questions. For a question $q_i \in Q$ and a specification $s_j \in S$, we find dual embedding score $DUAL(q_i, s_j)$ using Equation 1, where t_q and t_s denote the vectors for the question and specification terms, respectively. We consider (q_i, s_j) pair positive if $DUAL(q_i, s_j) \geq \theta$ and negative if $DUAL(q_i, s_j) < \theta$.

$$DUAL(q_i, s_j) = \frac{1}{|q_i|} \sum_{t_q \in q_i} \frac{t_q^T \overline{s_j}}{\| t_q \| \| \overline{s_j} \|} \quad (1)$$

where

$$\overline{s_j} = \frac{1}{|s_j|} \sum_{t_s \in s_j} \frac{t_s}{\| t_s \|} \quad (2)$$

We take $Mobile$ dataset to create labelled training data since most of the questions come from this vertical. We choose the threshold value (θ) which gives the best accuracy on manually labelled balanced validation dataset consisting of 380 question and specification pairs. We train a word2vec (Mikolov et al., 2013) model on our training dataset to get the embeddings of the words. The word2vec model learns two weight matrices during training. The matrix corresponding to the input space and the output space is denoted as IN and OUT word embedding space respectively.

Dataset	Model	HIT@1	HIT@2	HIT@3
	BERT-380	0.05	0.09	0.14
	XLNet-380	0.20	0.32	0.39
AC	Siamese	0.38	0.53	0.61
	BERT	0.62	**0.77**	**0.81**
	XLNet	**0.69**	**0.77**	0.80
	BERT-380	0.17	0.27	0.34
	XLNet-380	0.27	0.41	0.48
Backpack	Siamese	0.35	0.53	0.65
	BERT	**0.50**	0.66	0.69
	XLNet	0.49	**0.67**	**0.70**
	BERT-380	0.14	0.16	0.22
	XLNet-380	0.06	0.16	0.18
Computer	Siamese	0.5	0.6	0.72
	BERT	0.68	0.80	0.90
	XLNet	**0.70**	**0.86**	**0.92**
	BERT-380	0.22	0.40	0.55
	XLNet-380	0.25	0.45	0.60
Shoes	Siamese	0.42	0.55	0.62
	BERT	0.60	0.72	0.84
	XLNet	**0.63**	**0.77**	**0.88**
	BERT-380	0.05	0.09	0.15
	XLNet-380	0.24	0.36	0.45
Watches	Siamese	0.42	0.65	0.69
	BERT	0.54	0.60	0.74
	XLNet	**0.60**	**0.76**	**0.84**

Table 3: Performance comparison of different models.

Word2vec leverages only the input embeddings (IN), but discards the output embeddings (OUT), whereas DESM utilizes both IN and OUT embeddings. To compute the DUAL score of a question and specification, we take OUT-OUT vectors as it gives the best validation accuracy. We find that for $\theta = 0.34$, we gain maximum accuracy value of 0.72 on the validation set. This creates a labelled training dataset $\mathcal{D}$ with $57,138$ positive pairs and $655,290$ negative pairs. For training, we take all the positive data from $\mathcal{D}$ and we randomly sample an equal number of negative examples from $\mathcal{D}$.

To create the test datasets, domain experts manually annotate the correct specification for a question. As the test datasets come from different verticals, there is no product in common with the training set. The details of different test datasets are shown in Table 2. We analyze the questions in the test datasets and find that the questions can be roughly categorized into three classes - numerical, yes/no and others based upon the answer type of the questions. For a question, we have a number of specifications and only one of them is correct.

5.2 Training and Evaluation

We split the *Mobile* dataset into 80% and 20% as training set and development set, respectively. The Siamese model is trained for 20 epoch with Stochastic Gradient Descent optimizer and learn-

Question	Siamese	BERT	XLNet
Is it single core or multi core?	processor name core i3	internal mic single digital microphone	**number of cores 2**
	processor variant 7100u	processor name core i3	processor name core i3
	os architecture 64 bit	**number of cores 2**	processor brand intel
Does 16 inch laptop fit in to it?	depth 13 inch	**compatible laptop size 15.4 inch**	**compatible laptop size 15.4 inch**
	width 9 inch	laptop sleeve no	depth 13 inch
	height 19 inch	depth 13 inch	height 19 inch

Table 4: Top three specifications returned by different models for two questions. Correct specification is highlighted in bold.

ing rate 0.01. The fine-tuning of BERT and XL-Net is done with the same experimental settings as given in the original papers. In all the models, we minimize the cross-entropy loss while training. BERT-380 and XLNet-380 models are fine-tuned on the 380 labeled validation dataset that was used for creating the training dataset in Section 5.1.

During evaluation, we sort the question specification pairs according to their relevance score. From this ranked list, we compute whether the correct specification appears within top $k, k \in \{1, 2, 3\}$ positions. The ratio of correctly identified specifications in top 1, 2, and 3 positions to the total number of questions is denoted as HIT@1, HIT@2 and HIT@3 respectively.

6 Results and Discussion

Table 3 shows the performance of the models on different datasets[3]. BERT-380 and XLNet-380 perform very poorly, but when we use the train dataset created with DESM, there is a large boost in the models' performance and it shows the effectiveness of our semi-supervised method in generating labeled dataset. Both BERT and XLNet outperform the baseline Siamese model (Lai et al., 2018) by a large margin, and retrieve the correct specification within top 3 results for most of the queries. For *Backpack* and *AC*, both BERT and XLNet are very competitive. XLNet outperforms BERT in *Computer*, *Shoes*, and *Watches*. Only in HIT@1 of *AC*, BERT has surpassed XLNet with 0.07 points. We see that all the models have performed better in *Computer* compared to the other datasets. *Computer* has the highest percentage of yes/no questions and this might be one of the reasons, as some questions might have word overlap with correct specification. Table 4 shows the top three specifications returned by different models for some

questions. We see that Siamese architecture returns results which look similar to naïve word match, and retrieve wrong specifications. On the other hand, BERT and XLNet are able to retrieve the correct specifications.

Error Analysis: We assume that for each question, there is only one correct specification, but the correct answer may span multiple specifications and our models can not provide a full answer. For example, in *Backpack* dataset, the dimension of the backpack, i.e., its height, weight, depth is defined separately. So, when user queries about the dimension, only one specification is provided. Some specifications are given in one unit, but users want the answer in another unit, e.g., "what is the width of this bag in cms?". Since the specification is given in inches, the models show the answer in inches. So, the answer is related, but not exactly correct. Users sometimes want to know the difference between certain specification types, what is meant by some specifications. For example, consider the questions "what is the difference between inverter and non-inverter AC?", "what is meant by water resistant depth?". While we can find the type of inverter, the water resistant depth of a watch etc. from specifications, the definition of the specification is not given. As we have generated train data labels in semi-supervised fashion, it also contributes to inaccurate classification in some cases.

7 Conclusion and Future Work

In this paper, we proposed a method to label training data with little supervision. We demonstrated that large pretrained language models such as BERT and XLNet can be fine-tuned successfully to obtain product specifications that can help answer user queries. We also achieve reasonably good results even while testing on different verticals.

We would like to extend our method to take into

---

[3]Unsupervised DUAL embedding model gave very similar results to Siamese model, and is not reported.

account multiple specifications as an answer. We also plan to develop a classifier to identify which questions can not be answered from the specifications.

References

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*, volume 33, pages 45–52.

Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A customer service chatbot for e-commerce websites. pages 97–102.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2019. Is it dish washer safe? automatically answering "yes/no" questions using customer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 1–6, Minneapolis, Minnesota. Association for Computational Linguistics.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 429–437, New York, NY, USA. Association for Computing Machinery.

Tuan Lai, Trung Bui, Sheng Li, and Nedim Lipka. 2018. A simple end-to-end question answering model for product information. In *Proceedings of the First Workshop on Economics and Natural Language Processing*, pages 38–43, Melbourne, Australia. Association for Computational Linguistics.

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*, WWW '16, page 625–635, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

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*, pages 2192–2203, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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*, KDD '19, page 1389–1398, New York, NY, USA. Association for Computing Machinery.