Distantly Supervised Transformers For E-Commerce Product QA

Happy Mittal
India Machine Learning
Amazon
mithappy@amazon.com

Aniket Chakrabarti
India Machine Learning
Amazon
chakanik@amazon.com

Belhassen Bayar*
Digital Video
Amazon
bayarb@amazon.com

Animesh Anant Sharma
Community Shopping
Amazon
shanimes@amazon.com

Nikhil Rasiwasia
India Machine Learning
Amazon
rasiwasi@amazon.com

Abstract

We propose a practical instant question answering (QA) system on product pages of e-commerce services, where for each user query, relevant community question answer (CQA) pairs are retrieved. User queries and CQA pairs differ significantly in language characteristics making relevance learning difficult. Our proposed transformer-based model learns a robust relevance function by jointly learning unified syntactic and semantic representations without the need for human labeled data. This is achieved by distantly supervising our model by distilling from predictions of a syntactic matching system on user queries and simultaneously training with CQA pairs. Training with CQA pairs helps our model learning semantic QA relevance and distant supervision enables learning of syntactic features as well as the nuances of user querying language. Additionally, our model encodes queries and candidate responses independently allowing offline candidate embedding generation thereby minimizing the need for real-time transformer model execution. Consequently, our framework is able to scale to large e-commerce QA traffic. Extensive evaluation on user queries shows that our framework significantly outperforms both syntactic and semantic baselines in offline as well as large scale online A/B setups of a popular e-commerce service.

1 Introduction

Product pages on an e-commerce service (eg. Amazon) are often overloaded with information. Customers wanting to search for a piece of specific information about a product find it difficult to sift through. To address this issue most services provide an instant QA system on the product pages enabling users to type their query and get instant answers curated from various sources present on the page. Figure 1 shows the QA widget on Amazon, and the three sources viz. Product information (eg: bullet points, technical specifications etc.), Customer Q&A's (where customers/sellers provide an answer to the posted questions by customers, henceforth called community QA or CQA section), and Customer reviews from where a response is generated. In this paper, we focus on retrieving responses

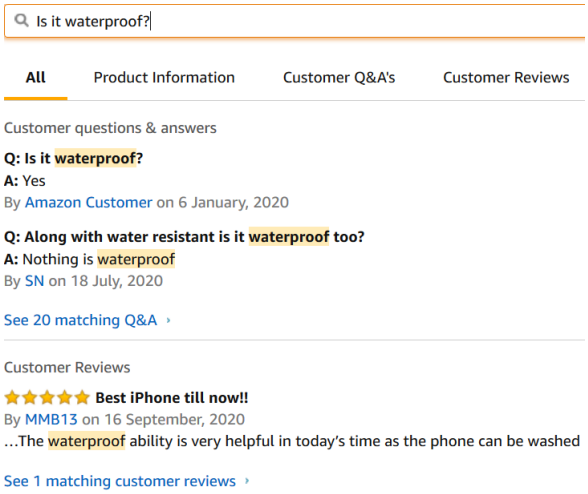

Figure 1: Instant QA widget on Amazon

from the CQA section. Hence our goal is to learn a robust relevance function between user queries and CQA pairs. Notably, these two domains differ significantly in language characteristics. User queries are typically short, often ill-formed and incomplete, whereas CQA pairs tend to be more complete and well-formed. For example, "Bettry perfon" is a user

*This work was done while author was in Community Shopping team.

query where the intended question probably was "how is the battery performance?". Furthermore, we analyzed CQA section along with 3 months user query logs of a popular e-commerce service and found that the data statistics such as length, vocabulary overlap (between user queries and CQA) indicate that the domains are quite different. Consequently, relevance learning for this task is difficult. Table 1 characterizes these differences for 4 different locales: Canada (CA), Germany (DE), France (FR), and IN (India).

	Vocab Overlap Percentage	Avg. Length	
		User Query	CQA Question
CA	55.4	3.26	11.79
DE	59.4	2.58	12.05
FR	59.1	4.21	13.79
IN	39.6	3.06	8.56

Table 1: Differences in user queries and CQA

Existing QA systems typically work by retrieving a set of candidates for a user query using syntactic features (eg. BM25 that uses bag of words features) followed by a semantic answer selection/re-ranking step (Chen et al., 2017). Some approaches include semantic features in the candidate generation step (Mitra and Craswell, 2019). Syntactic systems fail in two cases: (1) when there are no word overlaps (a likely scenario as user queries have limited vocabulary overlap with CQA pairs), and (2) when the word overlaps are semantically irrelevant. While adding semantic features or semantic re-ranking models mitigate some of the drawbacks, however, training a robust semantic relevance model to match user queries with CQA pairs is difficult due to the lack of human-labeled data. An additional challenge is that the instant QA system needs to provide real-time responses to users and must scale to the very large traffic of modern e-commerce systems. Running deep models online (typical in case of re-ranking) is prohibitive for such a system.

In this paper, we present an instant QA system with two main contributions: (1) our framework is able to learn a robust relevance function between user queries and CQA pairs by jointly learning semantic and syntactic features-aware representations without the need for explicit human-labeled data, and (2) our framework minimizes the need for real-time model execution by encoding the CQA pairs offline, enabling large scale online deployment.

We chose BERT (Devlin et al., 2019) as our transformer encoder due to its recent success in various natural language understanding (NLU) tasks including QA. To address the lack of labeled training data challenge, we use the QA pairs from the CQA section of each product page as training data. However, as shown in our evaluation (section 4.3), such a model does not work well on the user queries asked on the instant QA system on the product pages. We propose a distillation-based distantly supervised training algorithm where we use the answers retrieved by a syntactic match system on a set of user queries asked on the instant QA system. This training helps the model adapt to the specific task at hand by learning the user query distribution as well as the strengths of a traditional syntactic match system. This coupled with training on CQA pairs helps our model learn a robust semantic model that is task aware. Our training data does not require any explicit human labeling.

To make our system work in real-time we train the BERT model in Siamese style (Reimers and Gurevych, 2019) with triplets consisting of query, relevant candidate (+ve sample), and irrelevant candidate (-ve sample). Hence the query and candidate responses are encoded independently using the same transformer encoder enabling embedding computation of all candidates (across all products) offline. At real-time, only the user query needs to be embedded using the heavy semantic model resulting in a significant reduction of online compute cost. In contrast, the common practice of using BERT in QA problems is to concatenate the query and a candidate response and run BERT on the fused input. This would require BERT to run on all query, candidate CQA pairs on product pages real-time making it prohibitive for online deployment. Additionally, we combine the two embeddings (question and answer) in each CQA pair to form one embedding per pair allowing us to reduce the offline storage significantly.

We extensively evaluate our framework on user queries asked on the instant QA system at a popular e-commerce system in 4 locales spanning 3 languages. Offline evaluation shows that our proposed framework is able to increase the area under the precision-recall curve (PR-AUC) by up to 12.15% over the existing system. Also in an online A/B test, our system is able to improve coverage by up to 6.92% by complementing the existing system.

2 Related Works

QA Systems: Question Answering (QA) is a fundamental task in the Natural Language Understanding (NLU) domain. Broadly QA systems can be categorized into open-domain QA and closed-domain QA. Open-domain QA involves answering questions related to all topics from a huge repository of information such as the Web (Voorhees and Tice, 1999), Wikipedia corpus (Yang et al., 2015), Knowledge Bases (Bollacker et al., 2008). Closed-domain QA systems usually deal with a specific domain such as medical, sciences etc. The main steps of a QA system are candidate retrieval followed by answer selection/re-ranking (Chen et al., 2017). Some systems do answer generation (Lewis et al., 2020) instead of selection.

Semantic Text Encoders: Recently, QA systems have significantly evolved from syntax based (eg. BM25) systems to leverage the power of semantic text representation models. Recurrent Neural Networks (RNN) such as Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks were defacto for semantic text representation. Recently proposed self attention based transformer (Vaswani et al., 2017) models show consistent improvement over RNNs on a multitude of NLU tasks such as Machine Translation (MT) (Vaswani et al., 2017), Machine Reading Comprehension (Rajpurkar et al., 2016), GLUE (Devlin et al., 2019) and Natural Language Generation (NLG) tasks (Radford et al., 2019).

E-commerce Product QA Systems: E-commerce Product QA systems are similar to domain specific systems. Recently product QA systems are receiving a lot of attention due to their growing usage and unique characteristics such as the search space being specific to each product. Product QA systems are real-time systems where a user types a query and expect instant answers, the queries of such systems are typically short, prone to errors and even incomplete in nature. This coupled with product specific limited search space, often results in no syntactic match between the query and candidate answers, making semantic matching essential. In contrast, the retrieval set for websearch and traditional IR typically is huge and there are always bag-of-words matches that are used to filter down the candidates before running subsequent deep models. Additionally, search and IR systems in e-commerce/web domains get powerful implicit supervision signals through user clicks, however, instant QA on product pages only show the answer with no option to click making it hard to get user feedback based labels. Finally, QA relevance is different from traditional IR relevance (eg. for query "what is the material?", the response "made of stainless steel" is relevant and doesn't require bag-of-words or even synonym matches) making domain specific semantic matching critical. Kulkarni et al. (Kulkarni et al., 2019) propose an embedding based semantic matching model to find relevant answers. Additionally, it uses a query category classifier and an external ontology graph both of which require human generated labels. There are several proposed works (Zhang et al., 2020b, 2019, 2020a; Chen et al., 2019a; McAuley and Yang, 2016; Burke et al., 1997; Gupta et al., 2019) that improve the QA relevance models (usually learned from CQA pairs) by enriching them using information from reviews of the product and capturing their relation with the CQA pairs. Natural language answer generation models are also used in the context of product QA (Deng et al., 2020; Chen et al., 2019b; Gao et al., 2019; Bi et al., 2019). They are typically encoder-decoder architectures and their variants. These models are hard to generalize and often result in factually incorrect text generation. The aforementioned works use reviews and other product information along with CQA section to guide the models to generate answers.

In this paper, we take the approach of answer retrieval (instead of generation). We solve the orthogonal problem of how to adapt the relevance model to be aware of the user query characteristics (significantly different from the well formed questions posted in the CQA section) in the absence of human labeled data. The improvement in relevance models (between user queries and CQA pairs) proposed can be easily complemented with the existing review awareness models. A drawback of the aforementioned models is they comprise of multiple deep neural components, many of which need to be run real-time making online model deployment and computation cost prohibitive for large scale deployment. Our framework only needs to encode the user query realtime, all candidate responses are precomputed stored in an index making it amenable to real-time deployment.

3 Semantic QA System

In this section, we describe our proposed semantic QA system for e-commerce services. Unlike tradi-

tional QA systems where multiple models are used sequentially to surface the final response (eg. candidate retrieval, followed by answer selection/re-ranking, followed by span selection), here we use a semantic index and the top results retrieved from the index are the final answers shown to the users. Below we describe the problem definition followed by individual components of our system:

3.1 Problem Statement

Given a set of N products, a user query u_q on product p and the set of CQA pairs for all products $C = \{\{Q, A\}_p\}$ where $p \in \{1, N\}$ and $\{Q, A\}_p = \{\{q, a\}_p^1, \{q, a\}_p^2, ...\{q, a\}_p^n\}$ are the set of n QA pairs for product p, the goal is to find the relevant QA pairs set $R \subseteq C$ such that $\forall\{q, a\} \in C, \{q, a\}$ can answer u_q.

3.2 Model Architecture

We chose the transformer network (Vaswani et al., 2017) as our core text representation model. Transformers are largely successful in QA systems (eg. BERT for MRC (Devlin et al., 2019)), however, the typical approach to use transformers in a QA setting is to create a single input concatenating both the user query and a candidate response, enabling transformers to leverage a full contextual representation through attention mechanisms. Since transformer models are usually very large (hundreds of millions of parameters), this makes it infeasible to run the model real-time on a large candidate set. Our goal in this work is to leverage the strengths of the deep representational power of transformers while being able to scale to a real-time system with large candidate sets. Hence we propose to use transformers in a Siamese network setting similar to Sentence BERT (Reimers and Gurevych, 2019) to embed the query and the candidate responses independently. The same transformer encoder is used to encode both the query as well as the candidate responses (CQA pairs). This enables offline encoding of all CQA pairs and at real-time only, the user query needs to be encoded making the model productionizable at scale.

In our model, a sequence of text is encoded first by passing it through the transformer network that generates embeddings for each token in the input sequence. A mean pool (average) of the output token embeddings represents the full sequence.

$$e(text) = meanpool(transformer(text)) \quad (1)$$

We train our transformer based QA system using the triplet loss (Chechik et al., 2009) that tries to learn a similarity measure between user query, CQA pairs while maximizing the margin between relevant pairs and irrelevant pairs. Such ranking loss has proven effective at numerous ranking tasks (Chechik et al., 2009; Schroff et al., 2015; Wang et al., 2014). The triplet loss for a query q (also known as anchor), a relevant candidate response c_{+ve}, and an irrelevant candidate response c_{-ve}, is formally defined as:

$$\max\big(||e(q)-e(c_{+ve})|| - ||e(q) - e(c_{-ve})|| + \epsilon, 0\big) \quad (2)$$

where $||\cdot||$ is the Euclidean distance, and ϵ is the margin. The goal is to maximize the loss over the triplets of the training set.

3.3 Distantly Supervised Training

One of the biggest challenges in training the instant QA system for an e-commerce service is the lack of task specific labeled data. One source of labeled data is the CQA pairs. To create the relevant pairs (positive samples) and irrelevant pairs (negative samples) we adopt the following sampling strategy: (1) we sample user questions (as anchors) from all product pages' CQA section. This ensures the diversity of products in the training data. (2) For each question, we pick a paired answer to that question as the relevant pair. (3) For the same user question, we randomly select negative samples (answers from different user questions) both from the same product page and from other product pages. The negatives from the same product page are the hard negatives (as these answers are related to the current product whereas answers from other product pages likely are completely unrelated and easy to distinguish). In future, we wish to explore advanced negative sampling strategies such as Kumar et al. (Kumar et al., 2019) for answer sampling. However, for pages having very few CQA pairs, the number of negative samples becomes small, and adding negative samples from other product pages is useful in such scenarios even though those may be easy negatives. We show (in section 4.3) that such a model learns a good QA relevance function (between community questions and answers), however, it fails to learn a robust relevance function between the typical user queries asked on the instant QA widget and the CQA pairs (candidate responses). The underlying reason is the difference

in characteristics of the questions/answers posted in CQA forum (typically long, well-formed, and complete) and the queries asked on the instant answer widget (often short, grammatically incorrect, and ill-formed). Consequently, a model trained to learn relevance between community questions and answers performs very well when the queries are long and well-formed, however, they perform poorly on the queries typically asked by a user on the instant answer widget.

To address the aforementioned challenge, we propose a knowledge distillation (Hinton et al., 2015) based training technique that acts as distant supervision on our Siamese transformer network. We collect a random set of user queries asked on the instant QA system and the responses (CQA pairs) generated by the existing syntactic match system from the query logs of a popular e-commerce service. For generating the relevant pairs we take a user query as the anchor question and the answer from the CQA pair retrieved by the existing system. For generating the irrelevant pairs we follow a similar negative strategy as before. The existing syntactic match based system can be thought of as the teacher model and the Siamese transformer model is the student model in the distillation process. This distant supervision helps our semantic model adapt to the nuances of the instant QA system where queries are often short, and incoherent. Additionally, the distant supervision system also helps the semantic model learn the strengths of syntactic match systems.

We train our Siamese transformer network with data from both the aforementioned sources (CQA pairs, distilling from predictions of syntactic match based system on real user queries). We explore two strategies for jointly training our model with the two data sources: (1) we mix the data from both sources and train our model with the single triplet loss, and (2) we train our model in a multi-task fashion where there is a task (triplet loss) for each of the two data sources. This joint training of a unified syntactic and semantic representation while adapting to the nuances of user querying language enables our instant QA system to learn a robust task specific relevance function. Hence our instant QA system serves as an end-to-end unified framework for the e-commerce product QA problem.

3.4 Model Inference

For our proposed model the input is a user query on the instant QA system. The query is embedded in real-time using equation 1 and searched against the candidate vectors (for that specific product) to retrieve the top-k most relevant candidates (where a candidate is an embedding of QA pair from the CQA section of the product). For the top-k search, we use a weighted combination of squared Euclidean distance between the query, question (of CQA pair) embeddings and query, answer (of CQA pair) embeddings. Our relevance score of a query, CQA pair is generated as follows:

$$s(q, Q, A) = \alpha||e(q) - e(Q)||^2 + (1-\alpha)||e(q) - e(A)||^2 \quad (3)$$

The above expression can be rewritten using linearity of inner products as follows:

$$||e(q)||^2 + \alpha||e(Q)||^2 + (1-\alpha)||e(A)||^2 - 2\langle e(q), \alpha e(Q) + (1-\alpha)e(A)\rangle \quad (4)$$

Here $\langle \cdot, \cdot \rangle$ denotes the inner product between vectors. From the expression in equation 4 we can see that instead of storing $e(Q)$, and $e(A)$ separately, we can store the weighted combination of the two vectors $\alpha e(Q) + (1-\alpha)e(A)$ along with two extra scalar dimensions $\alpha||e(Q)||^2$ and $(1-\alpha)||e(A)||^2$ and the rest of the terms are query related and are computed real-time. This enables us to reduce the offline index storage by half by storing only one vector per candidate QA pair. Note that to enable such relevance score computation we had to use the square of Euclidean distance (instead of vanilla Euclidean distance) as the relevance scoring function at inference time.

4 Experiments

We ran experiments both in offline settings as well as in large scale online setups. We evaluated our models across 4 locales with 3 languages to test whether our distant supervision based training approach is able to generalize across languages and varying data characteristics.

4.1 Methods

In this section, we describe the methods that we compare. All methods described below can encode query and candidates independently. Consequently, the candidate index may be computed offline for all of these methods, enabling large scale deployment.
BM25: BM25 (Robertson et al., 1994) is the defacto ranking function used in retrieval systems.

It relies on a weighted combination of Term Frequency (TF) and Inverted Document Frequency (IDF) matching. The standard form of the scoring function is as follows:

$$bm25(q, D) = \sum_{i=1}^{n} IDF(q_i) \frac{TF(q_i, D)(k+1)}{TF(q_i, D) + k\left(1 - b + b\frac{|D|}{avgdl}\right)}$$

$$where,\ IDF(q_i) = ln\left(\frac{N - m(q_i) + 0.5}{m(q_i) + 0.5} + 1\right)$$

Here q is the user query consisting of n terms $(q_1, q_2, \ldots q_n)$, D is a document (or a sequence of text), $TF(q_i, D)$ denotes the number of times q_i appears in D, $|D|$ denotes the number of terms in document D, $avgdl$ is the average number of terms per document, $m(q_i)$ is the number of documents containing the term q_i, N is the total number of documents in the corpus, and k, b are tunable parameters, which we fixed to 1.5 and 0.75 respectively (Manning et al., 2008). Given the $bm25$ function above, we derive the relevance function between a user query, and a CQA pair in a similar fashion as equation 3 as follows:

$$\alpha bm25(q, Q) + (1 - \alpha) bm25(q, A)$$

E-commerce Baseline: We use the syntactic feature based existing optimized instant QA system at a popular e-commerce service as a baseline. We collect the query and responses shown by the system from query logs.

Sentence-transformers-STS-NLI: We use sentence-transformers (Reimers and Gurevych, 2019, 2020) which are state-of-the-art Siamese style trained transformer models for the general purpose semantic textual similarity (STS) and natural language inference (NLI) task. For English, we use the roberta-large-nli-stsb-mean-tokens [1] model, and for French and German we use the xlm-r-100langs-bert-base-nli-stsb-mean-tokens [1] model as we found them to be the best performing pretrained models. The relevance function is computed in a similar fashion as equation 3.

SemQA-CQA: Our proposed model trained only with CQA data as described in section 3.

SemQA-CQA-DS: Our proposed model that was trained with CQA data and distantly supervised with predictions of syntactic match system on user queries as described in section 3.

[1]https://www.sbert.net/docs/pretrained_models.html

4.2 Training Setup

We collect training data from the CQA section and user query logs for CA, DE, FR and IN locales of a popular e-commerce service. For each locale, to generate the CQA triplets and user query triplets (for distant supervision), we use data from CQA section of products, and user query logs and follow the sampling strategy described in section 3.3. The dataset statistics are described in table 2.

	CQA Triplets	User Query Triplets
CA	5,317,904	1,063,580
DE	5,000,000	4,949,766
FR	1,500,000	173,258
IN	7,176,824	10,641,498

Table 2: Training data statistics

We use the bert-base-uncased[2] as the base transformer for our English models (for CA and IN locale), camembert-base (Martin et al., 2020) [3] as the base transformer for FR locale, and bert-base-multilingual-uncased [4] as the base transformer for DE locale. We train our models upto 10 epochs, with a batch size of 16, Adam optimizer with learning rate of $2e-5$ with a schedule of linear warmup of first 10000 steps and then linear decay. We set $\epsilon = 1$ in the loss equation 2, and $\alpha = 0.4$ in the inference equation 3. For the joint training (CQA triplets and user query triplets), we have two training runs (data mixing and multi-task as described in section 3.3) per locale and picked the best models (data mixing for CA, FR and multi-task for DE, IN). We use the Pytorch[5], Huggingface (Wolf et al., 2019) and Sentence-Transformers (Reimers and Gurevych, 2019) libraries to develop our models on an Nvidia V100 GPU and hence our training time per batch and inference time per sample are same as that of Sentence-Transformers with BERT (base-model, 110M parameters).

4.3 Offline Evaluation

We do offline evaluation of our models under two settings: (1) on CQA test sets collected from the product pages at a popular e-commerce service, and (2) on user queries test set collected from query logs of the instant QA system on product pages of

[2]https://huggingface.co/bert-base-uncased
[3]https://huggingface.co/camembert-base
[4]https://huggingface.co/bert-base-multilingual-uncased
[5]https://pytorch.org

the same e-commerce service.Table 3 contains the test data statistics.

	CQA Test Set	User Queries Test Set	
	#Questions	#Queries	#Query-Response Pairs
CA	2722	1485	5992
DE	2871	1351	5591
FR	2547	1762	5127
IN	2773	1459	4225

Table 3: Test data statistics

		M0	M1	M2	M3
P@1	CA	39.02	52.87	**74.10**	73.55
	DE	40.82	38.66	**73.04**	71.65
	FR	37.42	42.25	73.85	**75.34**
	IN	26.51	35.67	53.05	**53.62**
mAP	CA	41.02	51.23	**73.37**	72.46
	DE	45.24	43.35	**74.80**	73.04
	FR	41.24	44.46	74.27	**75.38**
	IN	43.93	51.12	**72.17**	71.89
MRR	CA	31.21	42.41	**65.17**	64.17
	DE	37.05	35.48	**68.30**	66.20
	FR	32.26	35.03	66.78	**67.37**
	IN	34.70	41.23	58.44	**58.46**

Table 4: Evaluation on CQA pairs. M0: BM25, M1: sentence-transformers-STS-NLI, M2: SemQA-CQA, M3: SemQA-CQA-DS.

Evaluation on CQA Dataset: The goal of this section is to evaluate the relevance between community questions and answers learned by different approaches. For all locales we randomly sample questions posted on product pages. The paired answers to those questions are considered to be relevant answers and all other answers (from other CQA pairs) of the product are assumed to be irrelevant answers. We only sampled products that at least have 5 CQA pairs posted. For each question, the task is to rank all the candidate answers according to relevance. We report precision@1 (P@1), mean average precision (mAP) and mean reciprocal rank (MRR) in table 4. Since there may be multiple paired answers to a community posted question, the rank (for MRR) of a relevant answer is the number of irrelevant answers ranked above it plus one. We observe that both SemQA-CQA and SemQA-CQA-DS are able to significantly outperform other methods. This is expected since both of these methods were trained using CQA data and hence is able to learn a good QA relevance function, whereas the sentence-transformers-STS-NLI were trained using STS and NLI tasks and they failed to generalize. However, CQA pairs are significantly different from the language of user queries and in the next section, we will evaluate on those queries (the main goal of this paper).

Evaluation on User Queries: To evaluate on user queries, we sample user queries (and their corresponding top responses) uniformly at random from the query logs of the instant QA system. We also retrieve the top responses generated by the different models we trained. These query, response pairs are labeled as relevant or irrelevant by a team of human annotators. We use the area under the precision recall curve (PR-AUC) as our quality metric. We report the absolute percentage points change in PR-AUC with respect to the E-commerce Baseline in table 6 (+ve sign implies PR-AUC has improved and -ve sign implies PR-AUC has decreased). We make the following observations: (1) the vanilla BM25 baseline performs the worst which is expected as it relies solely on syntactic matches and fails to capture semantic intent; (2) both the sentence-transformers-STS-NLI and our SemQA-CQA models fail to generalize validating our hypothesis that learning a general semantic matching model or a QA relevance model is not sufficient to learn the nuances of user querying language; (3) the SemQA-CQA-DS models significantly outperform all other models.There are two underlying reasons for these improvements. Firstly, SemQA-CQA-DS is able to leverage the semantic understanding capabilities (that Pretrained-Transformers and SemQA-CQA are also able to do), and secondly, SemQA-CQA-DS is also able to learn the nuances of the task specific query language leading to a better relevance model between user queries and CQA pairs (that are potential candidate responses).

Next, we do a qualitative analysis on the cases where SemQA-CQA-DS is able to improve on the E-commerce Baseline. We identify two main areas of improvement: (1) improving relevance in cases where the baseline fails to capture the semantic intent, and (2) improving coverage in cases where the baseline fails to retrieves any response. We present examples of both cases in table 5. The examples include cases where the language is ill-formed and incoherent and our distantly supervised

User query	Top CQA pair retrieved by SemQA-CQA-DS	Top CQA pair retrieved by E-commerce Baseline
Improving semantic relevance		
Do you have size variation??? Like i need this in bigger wood..	**Q:** Is this available in still large size **A:** yes .. available size is 7*5,8*6,9*7,12*9 in inches	**Q:** Is the wood and print waterproof ? **A:** YES
It is compatible in gaming	**Q:** Does it run gta v **A:** Yesss... Very fine	**Q:** Is it compatible with Amd A6 processor ? **A:** Yes it's compatible DDR4
Total weight	**Q:** Each 1 how to kgs **A:** 10 kgs	**Q:** Total diameter of the plates? **A:** Plate Dia is 9.5 inches Hole Dia is 30 mm
Improving coverage		
What is fabric	**Q:** Which material is the scarf made up of **A:** It is like soft satin silk	No response
The dress with hands or seelveless	**Q:** Is it sleeveless **A:** we give a extra sleeves so u can attach or not..as ur wish	No response
Bettry perfon	**Q:** Batrrey capictiy **A:** This Phone has a Wonderful 4000 Mah Battery with Battery Saver Options & Can Watch videos continuously for 18 Hours!!!	No response

Table 5: Qualitative examples.

	M0	M1	M2	M3
CA	-19.75	-1.11	+1.53	**+9.25**
DE	-13.03	-11.90	+5.46	**+12.15**
FR	-11.66	-4.54	+4.93	**+7.68**
IN	-16.66	-0.26	+0.39	**+4.37**

Table 6: PR-AUC on user queries evaluation set. M0: BM25, M1: sentence-transformers-STS-NLI, M2: SemQA-CQA, M3: SemQA-CQA-DS. Numbers denote the absolute percentage points change with respect to the E-commerce Baseline.

model still captures the intent and retrieve relevant responses.

4.4 Online Evaluation

We also ran a large scale online A/B experiment with 50% of the user traffic. All locales were experimented at least for two weeks to ensure diversity in periodic patterns and have enough queries to achieve statistically significant conclusions (p-values < 0.01 in Chi-Square tests) about the improvement in metrics. Here the SemQA-CQA-DS model is used to complement the existing E-commerce Baseline [6] to improve the coverage of the system. There are two metrics of interest: (1) the coverage (percentage of queries answered by the system), and (2) the new question asking rate (percentage of queries for which even after seeing the response, a user asks a question in the CQA forum; if the relevance of the answers improves, the question asking rate should decrease). We report the change in absolute percentage points with respect to the E-commerce Baseline (for coverage +ve is better, and for question asking rate -ve is better). The results are present in table 7. SemQA-CQA-DS was able to improve coverage while reducing the rate of new questions posted by users in all locales thereby showing the efficacy of our approach at scale.

	Coverage	Question Asking Rate
CA	+2.96	-0.69
DE	+3.12	-0.44
FR	+4.56	-1.60
IN	+6.92	-0.97

Table 7: A/B test evaluation. Numbers denote the absolute percentage points change of Treatment with respect to Control.

[6] Details can't be disclosed due to proprietary information

5 Conclusions & Future Works

In this paper we presented 'SemQA', a practical transformer-based framework to provide instant QA efficiently on the product pages of e-commerce services. Given a user query, our framework directly retrieves the relevant CQA pairs from the product page, where user queries and CQA pairs have significantly different language characteristics. Our model is able to learn a robust relevance function between user queries and CQA pairs by learning representations that leverage the strengths of both syntactic and semantic features, without the need for any explicit human labeled data. Our model is able to scale to large scale real-time e-commerce systems and at inference time only requires model encoding of user queries for by index lookups, and candidate responses are encoded offline into the index in a space efficient manner. Extensive offline evaluation shows our approach generalizes to multiple locales spanning different languages with a PR-AUC gain by upto 12.15% over the existing system at a popular e-commerce service. We also ran a large scale online A/B experiment with 50% of the user traffic and our framework was able to improve coverage by upto 6.92% by complementing the existing system.

As a future direction, we would like to expand our SemQA system to include responses from additional content on the product pages (reviews, descriptions etc.). We believe some of the existing approaches to leverage reviews (discussed in section 2) can be used to complement our system to expand our relevance model beyond CQA data. Another direction of research will be to include features such as accuracy, sentiment, freshness etc. within our proposed SemQA system's responses.

6 Acknowledgements

We thank all the anonymous reviewers for providing their valuable comments that helped us improve the quality of our paper. We also thank our colleagues in the science, product, and engineering teams at Amazon for their valuable inputs.

References

B. Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering. In EMNLP/IJCNLP.

Kurt D. Bollacker, C. J. Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In SIGMOD Conference.

R. Burke, K. Hammond, Vladimir A. Kulyukin, S. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. AI Mag., 18:57–66.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2009. Large scale online learning of image similarity through ranking. In IbPRIA.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In ACL.

L. Chen, Ziyu Guan, W. Zhao, Wanqing Zhao, Xiaopeng Wang, Zhou Zhao, and Huan Sun. 2019a. Answer identification from product reviews for user questions by multi-task attentive networks. In AAAI.

Shiqian Chen, Chenliang Li, Feng Ji, W. Zhou, and Haiqing Chen. 2019b. Review-driven answer generation for product-related questions in e-commerce. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.

Yang Deng, Wenxuan Zhanng, and Wai Lam. 2020. Opinion-aware answer generation for review-driven question answering in e-commerce.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

Shen Gao, Z. Ren, Yihong Zhao, Dongyan Zhao, D. Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary Chase Lipton. 2019. Amazonqa: A review-based question answering task. In IJCAI.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. ArXiv, abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation, 9:1735–1780.

Ashish Kulkarni, Kartik Mehta, Shweta Garg, Vidit Bansal, Nikhil Rasiwasia, and Srinivasan Sengamedu. 2019. Productqna: Answering user questions on e-commerce product pages. In Companion Proceedings of The 2019 World Wide Web Conference, pages 354–360.

Sawan Kumar, Shweta Garg, K. Mehta, and Nikhil Rasiwasia. 2019. Improving answer selection and answer triggering using hard negatives. In EMNLP/IJCNLP.

M. Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, A. Mohamed, Omer Levy, V. Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval. Cambridge university press.

Louis Martin, B. Muller, Pedro Javier Ortiz Suárez, Y. Dupont, L. Romary, 'Eric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. ArXiv, abs/1911.03894.

Julian McAuley and A. Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. ArXiv, abs/1512.06863.

Bhaskar Mitra and Nick Craswell. 2019. An updated duet model for passage re-ranking. ArXiv, abs/1903.07666.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In EMNLP.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In EMNLP/IJCNLP.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. ArXiv, abs/2004.09813.

S. Robertson, S. Walker, Susan Jones, M. Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In TREC.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. The 2015 IEEE conference on computer vision and pattern recognition.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NIPS.

E. Voorhees and Dawn M. Tice. 1999. The trec-8 question answering track evaluation. In TREC.

Jiang Wang, Yang song, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. The 2014 IEEE conference on computer vision and pattern recognition.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.

Yi Yang, Wen tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In EMNLP.

Shiwei Zhang, Jey Han Lau, Xiuzhen Zhang, Jeffrey Chan, and Cécile Paris. 2019. Discovering relevant reviews for answering product-related queries. 2019 IEEE International Conference on Data Mining (ICDM), pages 1468–1473.

Shiwei Zhang, Xiuzhen Zhang, Jey Han Lau, Jeffrey Chan, and C. Paris. 2020a. Less is more: Rejecting unreliable reviews for product question answering. ArXiv, abs/2007.04526.

Wenxuan Zhang, Yang Deng, and Wai Lam. 2020b. Answer ranking for product-related questions via multiple semantic relations modeling. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.