Too much of product information : Don't worry, let's look for evidence!

Aryan Jain[*]
Amazon
aryan110801@gmail.com

Jitenkumar Rana
Amazon
jitenkra@amazon.com

Chetan Aggarwal
Amazon
caggar@amazon.com

Abstract

Product question answering (PQA) aims to provide instant response to customer questions posted on shopping message boards, social media, brand websites and retail stores. In this paper, we propose a distantly supervised solution to answer customer questions by using product information. Auto-answering questions using product information poses two main challenges :(i) labelled data is not readily available (ii) lengthy product information requires attending to various parts of the text to answer the question. To this end, we first propose a novel distant supervision based NLI model to prepare training data without any manual efforts. To deal with lengthy context, we factorize answer generation into two sub-problems. First, given product information, model extracts evidence spans relevant to question. Then, model leverages evidence spans to generate answer. Further, we propose two novelties in fine-tuning approach: (i) First, we jointly fine-tune model for both the tasks in end-to-end manner and showcase that it outperforms standard multi-task fine-tuning. (ii) Next, we introduce an auxiliary contrastive loss for evidence extraction. We show that combination of these two ideas achieves an absolute improvement of 6% in accuracy (human evaluation) over baselines.

1 Introduction

Around the world, customers post millions of questions across digital mediums to obtain important information before completing their purchase journey for a given product. Plethora of content on product pages makes it very difficult for customers to discover relevant information which leads to questions. Answering customer questions instantly is very crucial for organizations to ensure a seamless buying experience, thereby increasing customer engagement and reducing purchase abandonment possibly due to lack of information. In this paper, we aim to build a scalable solution to auto-answer customer questions using product pages.

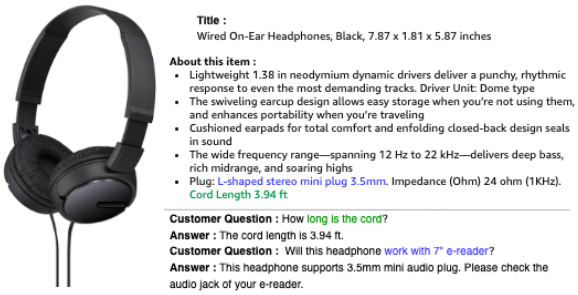

Figure 1: Example of typical product listings, questions and answers

Auto-answering product questions using product page content poses two main challenges. Firstly, labelled data for the task is not available. The existing (question, answer) pairs openly available on product pages are not sufficient since we need to ensure that answers posted are verified and question is answerable using the product content. Secondly, description information for products is very lengthy. Details of many top selling products span over six to eight thousand words which is equivalent to 15 A4 sheets (Mittal et al., 2021). Answering questions using lengthy contexts is a difficult task.

To tackle the challenges mentioned above, we first propose a distant supervision based natural language inference (NLI) model to prepare training data. We leverage NLI model to compute relevance between question, answer pair and question and sentences of product content. If product content contains high relevance sentences, we treat the question as answerable and unanswerable otherwise.

To deal with the lengthy context, we factorize PQA task into two sub-tasks. Task 1 (generatively) extracts evidence span from the context. Whereas, task 2 (answer generation) uses both evidence span and context to generate the answer.

[*]Work done during internship at Amazon

Next, we also propose several novelties in the training procedure. To capture dependency of answer generation on evidence span explicitly during training, we propose a joint end-to-end training of both the tasks. This is in contrast to standard multi-task training where every task is treated independently. Further, to improve performance of evidence selection task, we also introduce auxiliary contrastive loss (Caciularu et al., 2021) which helps model distinguish between supporting evidence and irrelevant sentences. We showcase that the combination of end-to-end training along with contrastive objective outperforms other baselines. To the best of our knowledge, this is the first work that deals with long context PQA with distantly supervised data creation.

Rest of the paper is organized as follows. We review related work in section 2. In section 3, we discuss distant supervision based training data creation approach. Section 4 explains the details of evidence extraction and answer generation task. We discuss experiments in section 5 and results in section 6. Finally, we conclude paper with a discussion on industry impact in section 7 and conclusion as well as future directions in section 8.

2 Related work

Product question answering has gained a lot of attention as a research problem. (Deng et al., 2023) provides a comprehensive survey of research work done so far in this space. We can divide the current approaches into two main categories : a) extractive answering b) generative answering. (Xu et al., 2019) proposed extracting answers as spans of text from reviews by post training BERT on review data. (Zhang et al., 2020b) leverages multiple heterogeneous sources such as reviews and structured attributes to filter snippets of text for answering a question. (Mittal et al., 2021) proposed distantly supervised extractive approach for PQA. To generate customer friendly responses, (Shen et al., 2022; Roy et al., 2022; Zhang et al., 2020a) explored generative approaches. They leverage LLMs such as T5 (Roberts et al., 2019), Flan-T5 (Chung et al., 2022) and Unified-QA (Khashabi et al., 2020) to generate natural language answers to questions. However, most of the existing models have short context window (<2k max token length) which limits their performance in long context scenario. In this work, we aim to combine the power of extractive and generative approaches for PQA for very lengthy product content.

3 Distant supervision for automated training data creation

In this section, we capture the data requirement, challenges, and distant supervision approach to automatically create training data. As stated earlier, the primary focus of this work is to answer questions using only product page content. Obtaining training data manually for thousands of product categories is challenging. Given the scope, we are faced with following three primary challenges: a) *Answer-ability*: We need to ensure that question in the training data is answerable using product content. b) *Unavailability of evidence*: The first sub-task requires ground truth evidence for training. There is no such dataset available as of today. c) *Truthfulness*: Answers posted can be incorrect since they are not moderated. We need to remove untrustworthy answers from dataset for better quality of training data. In subsection 3.2, we describe detailed approach to deal with challenges mentioned above.

3.1 Problem statement

Given a question q, list of answers $A = [a_1, a_2, \ldots, a_k]$ and product content P, goal is to create (q, a, S, P). Here, a is the correct answer for q, S is the list of supporting evidence sentences from P.

3.2 NLI model for training data creation

Figure 2 describes the details of process to obtain training tuple (q, a, S, P).

We start with AmazonPQA (Rozen et al., 2021), a publicly available dataset that contains product content including all the question-answers posted by customers and other product metadata from amazon.com. There can be multiple questions for a product and multiple answers for a question.

To obtain *answerable questions* with *correct answer* along with *supporting evidence* from AmazonPQA, we train and NLI model. First of all, to obtain the *correct answer* a, we select the answer provided by the highest rated sellers as it has the higher correctness compared to the other answers. Given input q and a sentence s, the NLI model outputs 1 if sentence is relevant to q and 0 otherwise. We need to obtain positive and negative (q, s) pairs to train NLI model. **Positive pairs** are mainly obtained by pairing existing (q, a) pairs. Since we

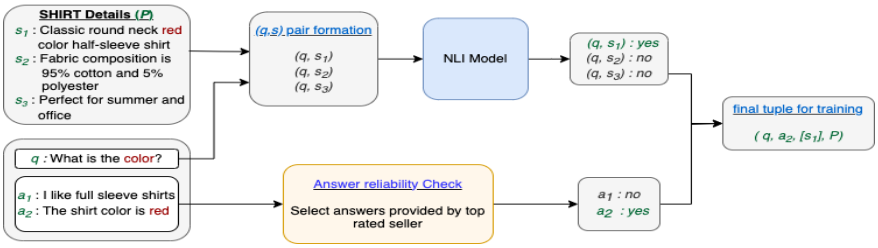

Figure 2: Distantly supervised training data creation for evidence extraction and answer generation task

aim to use the NLI model to identify evidence S from product content - where semantics are very different from actual answers, we also add artificial positive (q, s) pairs to the training data. To create artificial positive pair (q, s), we create questions q (using basic templates) for attributes (color, brand, model name) commonly present in product names and pair it with product name. For example, for a headphone product, we create question as "what is the color of the headphone?" and select name of the product as relevant sentence s.

Negative pair creation is straightforward. To do so, we pair q with any randomly selected sentence from product content or answer to other randomly selected question. We believe that (q, s) pair obtained by pairing q with randomly sampled sentence s from the same product content sentences serve as hard negative. In future, we also plan to leverage more advanced hard negative mining techniques.

Using the data (q, s) pairs generated above, we fine-tune FlanT5-base (Chung et al., 2022) for 2 epochs with learning rate of $2e^{-5}$. We observe that NLI model achieves 89% precision and 97% recall based on human evaluation when tested on a 20 product dataset that contains a total of 1237 (q, s) pairs.

Using the NLI model fine-tuned above and AmazonPQA, training tuple (q, a, S, P) for evidence extraction and answer generation task can be obtained using the steps mentioned below:

- Given multiple answers for a question q, select the answer provided by the highest rated seller.
- Use NLI model on (q, a) pair. If model output is "no", drop the q from the training set. This step ensures that answer is relevant to question and filter questions with junk answers.
- Split P to obtain list of sentences $C = [c_1, c_2, \ldots, c_n]$ using sentence tokenizer.
- Use NLI model for each $(q, c_i), i \in [1, 2, \ldots, n]$ to obtain the S (subset of C), the list of evidence sentences.
- If S is empty, it implies question is not answerable using P. In such case, we set a as "We can not answer the question based on product content information".
- If S is not empty, (q, a, S, P) is the desired training tuple.

4 Answer generation approach

4.1 Problem statement

Given a question q and product content P, the goal is to generate answer a using only the information provided by P.

4.2 Proposed approach

Figure 3 captures the details of the proposed approach. Formally, we motivate our approach based on following factorization of conditional probability of answer given question and product content:

$$p(a|q, P) = p(a|S, q, P) * p(S|q, P) \quad (1)$$

Here, S is the list of evidence spans relevant to question. This factorization corresponds to two stage approach: evidence extraction followed by answer generation. Specifically, in step 1, we propose to extract relevant spans $S = [s_1, s_2, \ldots, s_k]$ from P. In step 2, we propose to use q, S and P to generate answer a using the same model. Note that, we also use P along with S as input for answer generation task. Mathematically speaking, we don't assume that a and S are independent when conditioned on P. We will empirically show the merit of two stage approach in long context product question answering in section 6.

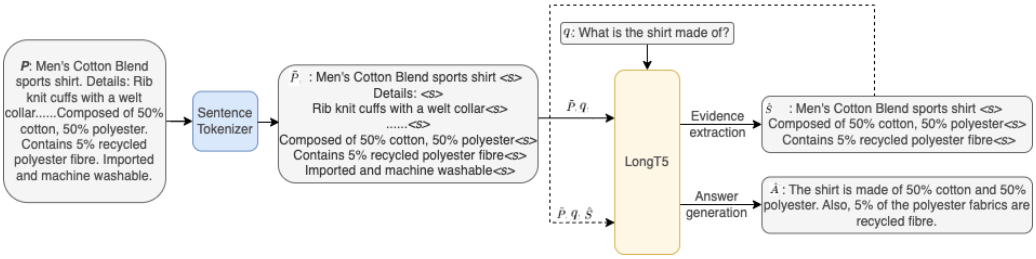

Figure 3: Proposed pipeline for question-answering

Table 1: Input format for evidence extraction and answer generation tasks

Task	Input
Evidence extraction	Select the sentences from the product content that can be used to answer the following question. Question: {q}<s>; product content: {$\tilde{P}$}
Answer generation	Given the product content and relevant sentences from product content, answer the following question. Question: {q}<s>; relevant sentences: {$\hat{S}^*$} product content: {$\tilde{P}$}.

* $\hat{S}$ is span extracted from evidence extraction task

4.3 Model input

Table 1 provides details of inputs for evidence extraction and answer generation tasks. Given question q and product content P, we first split P using sentence tokenizer to obtain $C = [c_1, c_2, \ldots, c_n]$. Then, we concatenate sentences in C using special token <s> to obtain $\tilde{P} = c_1$<s>c_2<s>$\ldots$<s>c_n<s>. Note that, we add special token <s> between every c_i. Encoder representations of <s> can be thought of as representation of sentence preceding <s>. We will show it later how encoding of <s> is leveraged to compute auxiliary contrastive loss for evidence extraction task.

4.4 Model training

In this paper, we introduce a novel combination of two ideas for model training: a) joint end-to-end fine-tuning b) contrastive loss for evidence extraction task. We will show in section 6 that combination of these two ideas improves performance of final answer generation task.

4.5 Joint end-to-end fine-tuning

Equation 2, 3 captures the details of joint fine-tuning. To capture the dependence of answer generation on evidence extraction, we use $\hat{S}$ as input to the answer generation task during training. Difference between standard multi-task training and end-to-end fine-tuning is that the standard multi-task training treats tasks independently and uses ground truth evidence S whereas the latter uses predicted evidence $\hat{S}$ as input for answer generation task during training phase. Conditioning on $\hat{S}$ for answer generation during training helps model capture the dependencies between evidence extraction and answer generation tasks.

$$\hat{S} = M_\theta(q, P), \hat{A} := M_\theta(q, \hat{S}, P) \tag{2}$$

$$\theta := optimizer(\theta, \nabla_\theta J(S, \hat{S}, A, \hat{A})) \tag{3}$$

Here, θ are the parameters of encoder-decoder model M_θ, $J(S, \hat{S}, A, \hat{A})$ is the total loss for end-to-end fine-tuning. Note that, the proposed training method is truly joint end-to-end fine-tuning since model uses $\hat{S}$ (instead of S) for answer generation during training.

4.6 Loss function

The loss function J consists of mainly three components : a) L_s: cross-entropy loss for evidence extraction, b) L_a: cross-entropy loss for answer generation task and, c) L_c: auxiliary contrastive loss for evidence extraction task (equation 5). Final loss is given in equation 4 below:

$$J = \underbrace{(1-\lambda)L_S + \lambda L_C}_{\textit{evidence extraction loss}} + \underbrace{L_A}_{\textit{answer generation loss}} \tag{4}$$

where, $\lambda \in [0, 1]$ is the hyper-parameter to adjust the weights of contrastive and cross-entropy loss in the overall evidence extraction loss.

Contrastive loss for evidence extraction is given below:

$$L_c = -log \sum_{s \in S} \frac{e^{sim(e_s, e_q)/\tau}}{\sum_{p \in P} e^{sim(e_p, e_q)/\tau}} \tag{5}$$

$$sim(s,q) = \frac{e_s^T W_c W_q e_q}{||e_s^T W_c|| . ||e_q^T W_q||} \tag{6}$$

Here, S is the list of ground truth evidence from P relevant to input question q, e_t is the encoder representation of token <s> which follows sentence t. τ is the temperature coefficient that can be tuned. W_c, W_q are the learnable projection matrices.

Given question q, product content P, and list of sentences relevant to question S, L_c tries to maximise similarity between q and $s \in S$ in a linearly projected space.

4.7 Inference

Figure 3 describes answer generation process. Given, question q, product content P, we first obtain $\tilde{P}$ which contains special token <s> after each sentence. Then using q and $\tilde{P}$, model first generates $\hat{S}$, the concatenation of all the relevant spans relevant to q from P. Next, model uses q, $\hat{S}$ and $\tilde{P}$ to generate the answer $\hat{A}$. Note that, we also use product content $\tilde{P}$ along with $\hat{S}$ for answer generation task. We will show in section 6 that including product content along with $\hat{S}$ reduces impact of evidence extraction error on answer generation.

5 Experiments

We conduct various experiments to evaluate the proposed approach proposed on following aspects: a) comparison with QA baselines, b) effectiveness of end-to-end fine-tuning and, c) effectiveness of proposed approach in lengthy context.

5.1 Baselines

LongT5 (Guo et al., 2022) is a scalable T5 architecture specifically trained to deal with long context window (upto 16k tokens). We use LongT5-Large as the base model for our experiments. We also compare performance of LongT5 with Flan-T5-Large which is a model with short context window of length 2k tokens. Models fine-tuned on only answer generation task are denoted as "model-name-A". Whereas, models fine-tuned on both the tasks in multi-task and joint end-to-end manner are denoted with "model-name-MT" and "model-name-E2E", respectively. Note that, MT and E2E approaches differ in only training. E2E approach uses predicted evidence whereas MT approach uses ground truth evidence during training. Inference procedure remains same for both the approaches. Further, we also compare performance of our model with GPT-3.5. Note that, the proposed architecture in this paper is "LongT5-E2E".

5.2 Ablation study

There are several decisions made in the training and inference approach: a) joint end-to-end fine-tuning (using predicted vs ground truth span for answer generation task during training), b) auxiliary loss for evidence extraction and, c) using product information along with evidence as opposed to only using evidence as input for answer generation during inference. We conduct systematic experiments to study impact of each of the design decisions.

5.3 Dataset

We use AmazonPQA and prepare training data using the distant supervision method mentioned in section 3. We use two test sets: Test-SC and Test-LC. Test-SC is a dataset with short product content (<2k tokens) whereas Test-LC is a long context dataset (>2k tokens). Both sets contain 2000 manually curated samples. Please refer to Table 2 for detailed data statistics.

Table 2: Data statistics

	Train	Test-SC	Test-LC
Product categories	11	11	11
Products	184,754	2,000	2,000
Questions	1,082,652	2,000	2,000
Answers	2,096,872	2,000	2,000
(q, a, S, P) tuples used	353,267	2,000	2,000
Mean token length of product content	5,475	1,092	5,523

5.4 Evaluation metrics

We use BLEU as automated metric for answer generation and tuning hyper-parameters. BLEU is the approximate indicator of the model's performance. Hence, we also report accuracy based on human evaluation for answer generation and report F1 based on ground truth and predicted evidence for evidence extraction.

We fine-tune all models for 2 epochs using Adam optimizer (Kingma and Ba, 2015), learning rate of $2e^{-5}$, batch size of 32.

6 Results

In this section, we discuss the observations made based on experiment results.

Joint end-to-end fine-tuning improves performance. From Table 3, we observe that LongT5-E2E achieves absolute improvement of $\sim 2\%$ and $\sim 6\%$ over LongT5-MT and LongT5-A, respectively. The only difference between LongT5-E2E and LongT5-MT is that the former is trained in truly end-to-end manner whereas former is not.

Table 3: Metrics on Test-LC

	Answer generation		Evidence extraction		
Model	Accuracy	BLEU	P	R	F1
LongT5-A	0.83	0.29	-	-	-
LongT5-MT	0.87	0.34	0.91	0.93	0.91
LongT5-E2E (ours)	0.89	**0.36**	**0.95**	**0.96**	**0.95**
GPT-3.5-turbo	**0.92**	0.20	-	-	-

Two-stage formulation achieves highest performance improvement. Table 3 suggests that two stage answer generation model LongT5-MT achieves absolute improvement of 4% in accuracy as compared to direct answer generation model LongT5-A particularly when input context is long.

Table 4: Answer generation accuracy for long and short context

	FlanT5-A	FlanT5-E2E	LongT5-A	LongT5-E2E
Test-SC (<2k context tokens)	0.89	0.90	0.89	0.89
Test-LC (>2k context tokens)	0.75	0.77	0.83	0.89

Two-stage formulation helps particularly in long context. Table 4 suggests that when context length is short, direct answer generation performs at par with two-stage approach. Further, FlanT5 also performs at par with LongT5 in short context scenario. However, performance gap between the two-stage and single stage approaches widens only in the high context length scenario.

Table 5: Ablation studies

		Answer generation		Evidence extraction		
Phase	Variation	Accuracy	BLEU	P	R	F1
Training	LongT5-E2E (only cross-entropy loss for evidence extraction)	0.88	0.34	0.92	0.91	0.91
	+contrastive loss for evidence extraction	0.89	0.36	0.95	0.96	0.95
Inference	LongT5-E2E (only $\hat{S}$ as input for answer generation)	0.87	0.31	0.93	0.94	0.94
	$\hat{S}$ and P as input for answer generation*	0.89	0.36	0.95	0.96	0.95

* P and $\hat{S}$ are product content and predicted evidence, respectively.

Contrastive loss improves performance of both tasks. Table 5 suggests that adding contrastive loss for evidence extraction task improves performance of both the tasks. This suggests that auxiliary loss helps model learn better alignment between question and evidence.

Using product content along with evidence as input for answer generation improves answer generation performance. We can see from Table 5 that performance improves by 2% accuracy points when product content is also used with evidence as input for answer generation. It suggests that answer and context conditioned on evidence are not independent. Qualitative analysis suggests that additional context helps model mitigate the impact of evidence extraction error on answer generation.

GPT-3.5-turbo outperforms LongT5-E2E as expected. As observed in Table 3, GPT-3.5-turbo performs low on BLEU score but achieves 3% absolute improvement on accuracy compared to the other models. Main reason for GPT-3.5-turbo's low BLEU score is that it generates lengthy output. Even though GPT-3.5-turbo's accuracy is higher, there are three major limitations that prevents us from using it in production system as of today: a) It hallucinates particularly in the case when question is not answerable using product content b) Cost is high due to paid API and, c) Inference latency is high for real-time application.

7 Industry impact

In this paper, we proposed a practical solution for auto-answering product queries using product information which helps customers make quicker purchase decisions. Applications of this work have the potential to auto-answer or reply in real-time to thousands of perennially unanswered questions leading to elimination of redundant work and resource savings.

8 Conclusion

In this paper, we proposed distant supervision based approach that combines the power of extractive and generative techniques for product question answering. There are two key contribution of the approach presented in this paper. First, we proposed a distant supervision and NLI based technique to create training data without any manual intervention. Next, proposed two-stage answer generation approach which achieves 6% point improvement in accuracy over only answer generation approach. Further, we also introduce a novel training mechanism which is a combination of two key ideas: a) Joint end-to-end fine-tuning b) contrastive loss for evidence extraction. We systematically studied the impact of each component and showed that the combination of ideas proposed above achieves highest performance. In future, we plan to extend this work to incorporate multi-modal input sources such as product reviews, images and videos. We can also leverage RLHF based techniques to achieve better alignment of the model output with human preferences.

References

Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2021. Utilizing evidence spans via sequence-level contrastive learning for long-context question answering. *CoRR*, abs/2112.08777.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Yang Deng, Wenxuan Zhang, Qian Yu, and Wai Lam. 2023. Product question answering in e-commerce: A survey.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences.

D. Khashabi, S. Min, T. Khot, A. Sabhwaral, O. Tafjord, P. Clark, and H. Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Happy Mittal, Aniket Chakrabarti, Belhassen Bayar, Animesh Anant Sharma, and Nikhil Rasiwasia. 2021. Distantly supervised transformers for E-commerce product QA. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4008–4017, Online. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google.

Kalyani Roy, Vineeth Balapanuru, Tapas Nayak, and Pawan Goyal. 2022. Investigating the generative approach for question answering in E-commerce. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 210–216, Dublin, Ireland. Association for Computational Linguistics.

Ohad Rozen, David Carmel, Avihai Mejer, Vitaly Mirkis, and Yftah Ziser. 2021. Answering product-questions by utilizing questions from other contextually similar products.

Xiaoyu Shen, Gianni Barlacchi, Marco Del Tredici, Weiwei Cheng, Bill Byrne, and Adrià Gispert. 2022. Product answer generation from heterogeneous sources: A new benchmark and best practices. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 99–110, Dublin, Ireland. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Jing Ma, and Wai Lam. 2020a. AnswerFact: Fact checking in product question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2407–2417, Online. Association for Computational Linguistics.

Wenxuan Zhang, Qian Yu, and Wai Lam. 2020b. Answering product-related questions with heterogeneous information. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 696–705, Suzhou, China. Association for Computational Linguistics.