Generating Helpful Responses for Intelligent Tech Support

Wenhao Yu[1], Lingfei Wu[2], Shu Tao[2], Yu Deng[2], Qingkai Zeng[1], Meng Jiang[1]
[1]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
[2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
wyu1@nd.edu, {wuli, shutao, dengy}@us.ibm.com, {qzeng, mjiang2}@nd.edu

Abstract

A new kind of Complex QA task is described along with its datasets and new challenges. The task is to generate helpful responses for intelligent technical support. Tech support forums (e.g., IBM DeveloperWorks, StackOverflow and AskUbuntu) are widely available on the Internet, where users can report their technical issues and seek for solutions. However, it is challenging to obtain timely and helpful responses. Compared with traditional answer generation tasks (e.g., SQuAD, MS-MARCO), response generation in the tech support domain has three unique challenges to AI and NLP technologies. First, questions and responses are often long-text with a mixture of both symptoms and irrelevant information. Second, responses are in different semantic spaces from the questions and need domain knowledge (e.g., from user guide and manuals) to generate. Third, a thread of discussion starts from a question post and follows with a series of responses (and helpfulness signals) and follow-up questions, which is more complex than a question-answer pair. New technologies are needed to address the challenges.

Introduction

Tech companies and communities have developed and actively maintained tech support discussion forums (e.g., IBM DeveloperWorks, StackOverflow[1] and AskUbuntu[2]) where users can post technical issues and seek for advice and solutions from peers and experts. Tech companies often train employees to have professional communication skills and technical knowledge that are needed to respond to users' questions. However, it is often expensive to provide *timely* and *helpful* responses on these forums. There are two main reasons for this. First, the users (peers) often have limited domain-specific knowledge to describe their problems, so it would be difficult to figure out the core problem and give helpful advice. It is common that the users add additional information, forming an interactive discussion of a long series of posts, which is very time consuming. Second, unlike the traditional reading comprehension tasks, helpful responses do not come from a given piece of contextual information, but rather require the respondent to have in-depth background knowledge or have solved similar problems. Therefore, it is highly desirable to generate correct responses from tech support forums. If developed, such technologies can be used to develop intelligent applications, such as real-time tech support dialogue systems.

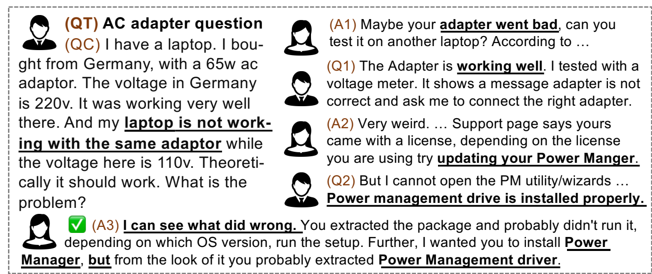

Figure 1: Discussion thread on a Tech Forum: An example

Figure 1 gives an example from a tech forum. Questions on the forum has two parts: question title (QT) and question content (QC). The question title summarizes the main content of the problem (e.g, battery problem, adapter problem). The question content describes the detailed information of the problem. Additionally, users usually give information of their products (e.g, device model, configurations, purchase year). Due to the limited background information, it is difficult for peer users to generate a response (A) of high quality as a good solution. Therefore, a series of follow-up discussions (A1, Q1, A2, Q2, etc.) enrich the context and eventually come to a solution-level response (A3).

Difference from existing tasks/models: The traditional QA task is to read a passage (or a set of passages) and generate the answer given a question. The open access of the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al. 2016) and the Microsoft MAchine Reading COmprehension Dataset (MS-MARCO) (Nguyen et al. 2016) provides large-scale manually created datasets for model training and testing for this task. There are some differences between SQuAD and MS-MARCO. First, SQuAD constrains an answer to be an exact sub-span in the passage, while answer words are not necessary in the passages of MS-MARCO. Second, SQuAD only has one passage for a question, while MS-MARCO has multiple passages. From the perspective of models, sequence-to-sequence models (Sutskever, Vinyals,

[1]https://stackoverflow.com/

[2]https://askubuntu.com

Figure 2: A summarization-expansion framework may address the *length* challenge (long-text question, response).

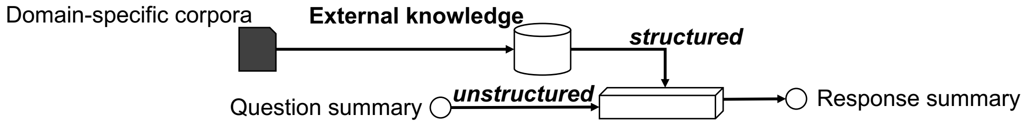

Figure 3: Responses are in a different space from questions and need external, specialized knowledge to be generated.

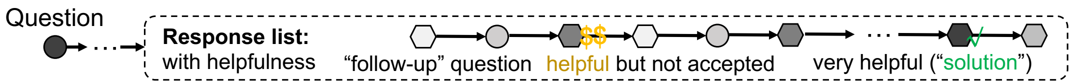

Figure 4: Beyond question-answer pairs: Modeling response sequences and helpfulness signals for generating the best solution.

and Le 2014) have demonstrated good performance in a wide range of natural language processing tasks including natural answer sentences generation. Besides, the application of attention mechanism (Vaswani et al. 2017) enabled attempts to learn sentence varieties from the memory and training data. For tech support domain, different from SQuAD and MS-MARCO, we are facing three unique challenges: (1) the text of questions and responses are of great length, (2) responses are in very different semantic spaces from questions and need domain knowledge to fill in the gap between questions and responses, and (3) existing methods cannot directly be applied to create long-text content with correct semantics.

Related Work

Long-text Generation. Although there are no related works on the generation of long-text questions and responses, the long-text generations of story, news and paper abstract have been extensively researched (Liang et al. 2017; Fan, Lewis, and Dauphin 2018; Wang et al. 2019). (Li, Luong, and Jurafsky 2015) proposed a hierarchical neural encoder-decoder for paragraph generation, which arranges tokens, sentences and paragraphs in a hierarchical structure, with different levels of encoder-decoder to capture their compositionality. In contrast, by dividing generation into two steps, (Wang et al. 2018) designed a novel writing-editing network that can attend to both the title and the previously generated abstract drafts and then iteratively revise and polish the abstract.

Knowledge Driven Generation. Modeling knowledge as a graph to improve generation results has been proved successfully on several tasks such as amr-to-text (Song et al. 2018), machine translation (Beck, Haffari, and Cohn 2018) and question generation (Chen, Wu, and Zaki 2020). Recently, (Koncel-Kedziorski et al. 2019) introduced a novel graph transforming encoder which can leverage the relational structure of knowledge graphs without imposing linearization or hierarchical constraints.

Datasets

Tech support forums are widely available on the Internet. After a user asks a question, peers and experts could make responses. Here are two example datasets which can be used for the question-answering task. For each of the datasets, user guide or developer documentation can serve as external knowledge sources (though unstructured). We are organizing the datasets (and collecting more datasets) and making documentation of them.

Dataset 1 based on a hardware support forum. This dataset is consisted of 86,394 question posts (threads) and 334,387 response posts. 17,952 (20.8%) threads have one response post labeled as "solution" by the questioner. The average length (i.e., #words) of questions and responses are 103 and 52, respectively.

Dataset 2 based on AskUbuntu. It has more than 332,914 question posts. Over 219,756 (66.0%) questions have a "solution" response. The average length of questions and responses are 51 and 37, respectively. They are relatively shorter compared to Dataset 1.

Challenges and Potential Ideas

Generating long-text questions and responses

Questions and responses (including solutions) on the tech support forums are often long-text with a mixture of symptoms and irrelevant information. For example, the question posts often have greeting words to general audience, appreciation words to thank for replies in advance, and some detailed information of the device model or configurations. The response posts may have greeting words to the questioner, redundant content (like direct quotes) from the question posts, or describe similar experience (which could be a long story). The solution posts may have itemized, suggested actions in steps. The irrelevant information could distort the representation of posts and hurt the quality of generated responses. A basic idea to address it is to add question summary and response summary as bridges to connect the long-text question post and response post (see Figure 2). Then the main task becomes generating response summary from question summary with a reasonable length. Thus a summarization module and an expansion module must be added to the framework.

Generating responses with external knowledge

Unfortunately, there is no "passage" (like in SQuAD) in the tech support data that contains both the question context and answer. Responses are in different semantic spaces from the questions and need domain knowledge to generate. The idea is to utilize the massive *unstructured* domain-specific corpora including user guide or developer documentations (see Figure 3). Knowledge bases and/or graphs can be constructed using information extraction technologies (Zeng et al. 2019; Yu et al. 2020a) and then be used for generating responses with rich and relevant semantics.

Learning helpfulness from response series

Problem solving on technical support forums is an interactive process. A thread of discussion starts from a question post and follows with a series of responses (and helpfulness signals). During the process, the questioner can describe the problem with additional information ("follow-up questions"). The forums often have mechanisms to motivate helpful responses. For example, on StackOverflow, users can give *up votes* to the responses they like. This is more complex than traditional inputs - question-answer pairs only. To generate helpful responses, the basic idea is to learn helpfulness signals by modeling the series of response sentences (Ouyang et al. 2019; Yu et al. 2020b), which could improve the quality of response generation process (see Figure 4).

References

[Beck, Haffari, and Cohn 2018] Beck, D.; Haffari, G.; and Cohn, T. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 273–283.

[Chen, Wu, and Zaki 2020] Chen, Y.; Wu, L.; and Zaki, M. J. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. *In The International Conference on Learning Representations*.

[Fan, Lewis, and Dauphin 2018] Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898.

[Koncel-Kedziorski et al. 2019] Koncel-Kedziorski, R.; Bekal, D.; Luan, Y.; Lapata, M.; and Hajishirzi, H. 2019. Text generation from knowledge graphs with graph transformers. 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)*, 2284–2293.

[Li, Luong, and Jurafsky 2015] Li, J.; Luong, T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1106–1115.

[Liang et al. 2017] Liang, X.; Hu, Z.; Zhang, H.; Gan, C.; and Xing, E. P. 2017. Recurrent topic-transition gan for visual paragraph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3362–3371.

[Nguyen et al. 2016] Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. Ms marco: A human-generated machine reading comprehension dataset.

[Ouyang et al. 2019] Ouyang, W.; Zhang, X.; Li, L.; Zou, H.; Xing, X.; Liu, Z.; and Du, Y. 2019. Deep spatio-temporal neural networks for click-through rate prediction. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[Rajpurkar et al. 2016] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

[Song et al. 2018] Song, L.; Zhang, Y.; Wang, Z.; and Gildea, D. 2018. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1616–1626.

[Sutskever, Vinyals, and Le 2014] Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

[Wang et al. 2018] Wang, Q.; Zhou, Z.; Huang, L.; Whitehead, S.; Zhang, B.; Ji, H.; and Knight, K. 2018. Paper abstract writing through editing mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 260–265.

[Wang et al. 2019] Wang, Q.; Huang, L.; Jiang, Z.; Knight, K.; Ji, H.; Bansal, M.; and Luan, Y. 2019. PaperRobot: Incremental draft generation of scientific ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1980–1991. Florence, Italy: Association for Computational Linguistics.

[Yu et al. 2020a] Yu, W.; Peng, W.; Shu, Y.; Zeng, Q.; and Jiang, M. 2020a. Experimental evidence extraction in data science with hybrid table features and ensemble learning. In *The World Wide Web Conference*. ACM.

[Yu et al. 2020b] Yu, W.; Yu, M.; Zhao, T.; and Jiang, M. 2020b. Identifying referential intention with heterogeneous contexts. In *The World Wide Web Conference*. ACM.

[Zeng et al. 2019] Zeng, Q.; Yu, M.; Yu, W.; Xiong, J.; Shi, Y.; and Jiang, M. 2019. Faceted hierarchy: A new graph type to organize scientific concepts and a construction method. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*.