Lingke: A Fine-grained Multi-turn Chatbot for Customer Service

Pengfei Zhu[1,2,4], **Zhuosheng Zhang**[1,2], **Jiangtong Li**[1,2,3], **Yafang Huang**[1,2], **Hai Zhao**[1,2,†]
[1]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[3]College of Zhiyuan, Shanghai Jiao Tong University, China
[4]School of Computer Science and Software Engineering, East China Normal University, China
101525101900@stu.ecnu.edu.cn, {zhangzs, keep_moving-lee}@sjtu.edu.cn,
huangyafang@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Traditional chatbots usually need a mass of human dialogue data, especially when using supervised machine learning method. Though they can easily deal with single-turn question answering, for multi-turn the performance is usually unsatisfactory. In this paper, we present Lingke, an information retrieval augmented chatbot which is able to answer questions based on given product introduction document and deal with multi-turn conversations. We will introduce a fine-grained pipeline processing to distill responses based on unstructured documents, and attentive sequential context-response matching for multi-turn conversations.

1 Introduction

Recently, dialogue and interactive systems have been emerging with huge commercial values (Qiu et al., 2017; Yan et al., 2016a; Zhang et al., 2017; Huang et al., 2018; Zhang et al., 2018b; Zhang et al., 2018a), especially in the e-commerce field (Cui et al., 2017; Yan et al., 2016b). Building a chatbot mainly faces two challenges, the lack of dialogue data and poor performance for multi-turn conversations. This paper describes a fine-grained information retrieval (IR) augmented multi-turn chatbot - Lingke. It can learn knowledge without human supervision from conversation records or given product introduction documents and generate proper response, which alleviates the problem of lacking dialogue corpus to train a chatbot. First, by using *Apache Lucene*[1] to select top 2 sentences most relevant to the question and extracting subject-verb-object (SVO) triples from them, a set of candidate responses is generated. With regard to multi-turn conversations, we adopt a dialogue manager, including self-attention strategy to distill significant signal of utterances, and sequential utterance-response matching to connect responses with conversation utterances, which outperforms all other models in multi-turn response selection. An online demo is available via accessing `http://47.96.2.5:8080/ServiceBot/demo/`.

2 Architecture

This section presents the architecture of Lingke, which is overall shown in Figure 1.

The technical components include 1) coreference resolution and document separation, 2) target sentences retrieval, 3) candidate responses generation, followed by a dialouge manager including 4) self-matching attention, 5) response selection and 6) chit-chat response generation.

The first three steps aim at selecting candidate responses, and in the remaining steps, we utilize sentences from previous conversations to select the most proper response. For multi-turn conversation modeling, we develop a dialogue manager which employs self-matching attention strategy and sequential utterance-response matching to distill pivotal information from the redundant context and determine the most proper response from the candidates.

† Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

[1]`http://lucene.apache.org`

Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 108–112
Santa Fe, New Mexico, USA, August 20-26, 2018.

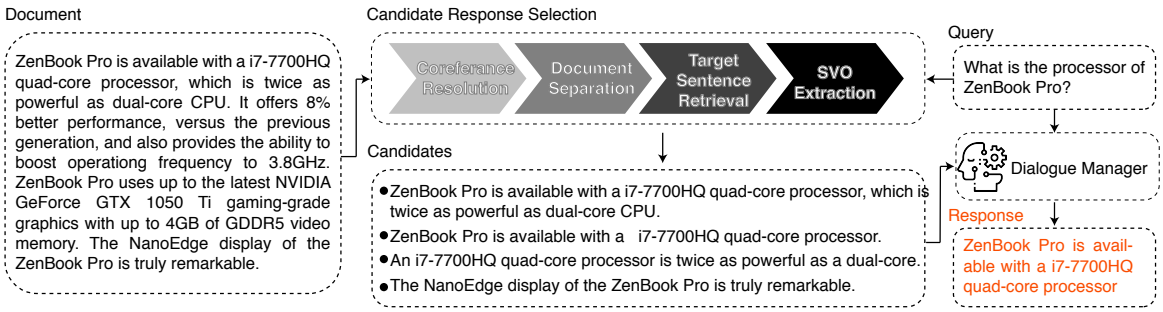

Figure 1: Architecture of Lingke.

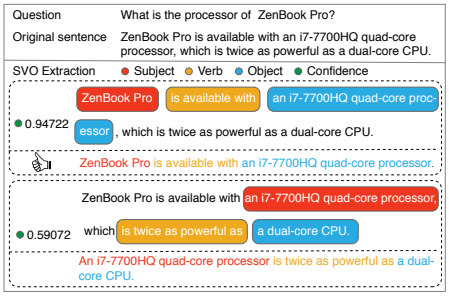

Figure 2: Example of SVO Extraction.

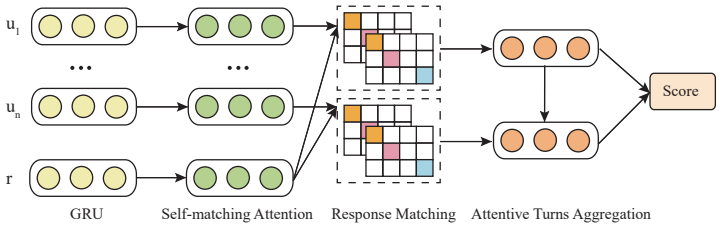

Figure 3: Structure overview of the dialogue manager.

Coreference Resolution and Document Separation Since the response is usually supposed to be concise and coherent, we first separate a given document into sentences. However, long documents commonly involve complex reference relations. A direct segmentation might result in severe information loss. So before the separation, we used *Stanford CoreNLP* (Manning et al., 2014) [2] to accomplish the coreference resolution. After the resolution, we cut the document into sentences $A = \{A_1, A_2, ..., A_n\}$.

Target Sentences Retrieval There is abundant information in the whole document, but what current message cares about just exists in some paragraphs or even sentences. So before precise processing, we need to roughly select sentences which are relevant with the current message. We used *Apache Lucene* to accomplish the retrieval. Given sentence collection A from step 1, we retrieve k relevant sentences $E = \{E_1, E_2, ..., E_k\}$. In our system, the value of k is 2.

Candidate Responses Generation Generally, the response to a conversation can be expressed as a simple sentence, even a few of words. However, sentences from a product introduction document are usually complicated with much information. To extract SVO, we used an open information extraction framework ReVerb (Fader et al., 2011), which is able to recognize more than one group of SVO triples (including triples from the clauses). Figure 2 shows an example. Based on an utterance E_i from E, we extract its SVO triples $E_s = \{E_{s1}, E_{s2}, ..., E_{sn}\}$, $E_v = \{E_{v1}, E_{v2}, ..., E_{vn}\}$, $E_o = \{E_{o1}, E_{o2}, ..., E_{on}\}$, and by concatenating each triple, we obtain multiple of simple sentences $T = \{T_1, T_2, ..., T_n\}$.

The above first three steps generate all sentences and phrases as candidate responses, which are denoted as $R = E \cup T$. What we need to do next is to rerank the candidates for the most proper response.

Dialogue Manager We combined self-matching attention strategy and sequential utterance-response matching to develop a multi-turn dialogue manager. Figure 3 shows the structure.

(1) Self-matching Attention Since not all of the information is useful, it is a natural thought that adopts self-matching attention strategy to filter redundant information. Before that, we transform raw dialogue data into word embedding (Mikolov et al., 2013) firstly. Each conversation utterance or candidate

[2] https://stanfordnlp.github.io/CoreNLP/index.html

response is fed to the gated recurrent units (GRUs) (Cho et al., 2014). Then, we adopt a self-matching attention strategy (Wang et al., 2017) to directly match each utterance or response against itself to distill the pivotal information and filter irrelevant pieces.

(2) Response Selection Following Sequential Matching Network (SMN) (Wu et al., 2017), we employ sequential matching for multi-turn response selection. Given the candidate response set, it matches each response with the conversation utterances in chronological order and obtains accumulated matching score of the utterance-response pairs to capture significant information and relations among utterances and each candidate response. The one with highest matching score is selected as final response.

(3) Chit-chat Response Generation When given a question irrelevant to current introduction document, *Target Sentences Retrieval* may fail, so we adopt a chit-chat engine to give response when the matching scores of all the candidate responses are below the threshold which is empirically set to 0.3. The chit-chat model is an attention-based seq2seq model (Sutskever et al., 2014) achieved by a generic deep learning framework *OpenNMT*[3]. The model is trained on twitter conversation data, which has 370K query-reply pairs, and 300K non-duplicate pairs are selected for training.

3 Experiment

Dataset We evaluate Lingke on a dataset from our *Taobao*[4] partners, which is a collection of conversation records between customers and customer service staffs. It contains over five kinds of conversations, including chit-chat, product and discount consultation, querying delivery progress and after-sales feedback. We converted it into the structured multi-turn format as in (Lowe et al., 2015; Wu et al., 2017). The training set has 1 million multi-turn dialogues totally, and 10K respectively in validation and test set.

	TF-IDF	RNN	CNN	LSTM	BiLSTM	Multi-View	SMN	**Our model**
$R_{10}@1$	0.159	0.325	0.328	0.365	0.355	0.421	0.453	**0.476**
$R_{10}@2$	0.256	0.463	0.515	0.536	0.525	0.601	0.654	**0.672**
$R_{10}@5$	0.477	0.775	0.792	0.828	0.825	0.861	0.886	**0.893**

Table 1: Comparison of different models.

Evaluation Our model is compared with recent single-turn and multi-turn models, of which the former are in (Kadlec et al., 2015; Lowe et al., 2015) including TF-IDF, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), LSTM and biLSTM. These models concatenate the context utterances together to match a response. Multi-view model (Zhou et al., 2016) models utterance relationships from word sequence view and utterance sequence view, and Sequential Matching Network (Wu et al., 2017) matches a response with each utterance in the context. We implemented all the models following the same hyper-parameters from corresponding literatures (Wu et al., 2017; Lowe et al., 2015). Our evaluation is based on Recall at position k in n candidates ($Rn@k$). Results in Table 1 show that our model outperforms all other models, indicating filtering redundant information within utterances could improve the performance and relationships among utterances and response can not be neglected.

4 Usability and Analysis

In this section, we will discuss the usability of Lingke. In situation of lacking enough dialogue data such as when a new product is put on an online shop, Lingke only needs an introduction document to respond to customers. Because of the chit-chat response generation engine, Lingke can easily deal with any commodity-independent conversations. Thanks to our multi-turn model, Lingke will not get confused when customer gives incomplete questions which need to be understood based on context.

Figure 4-5 show two typical application scenarios of Lingke, namely, *conversation record based* and *document-based* ones, which vary based on the training corpus. Figure 4 shows Linke can effectively respond to the customer shopping consultations. The customer sends a product link and then Lingke

[3] `http://opennmt.net/OpenNMT`

[4] It's the largest e-commerce platform in China. `https://www.taobao.com`.

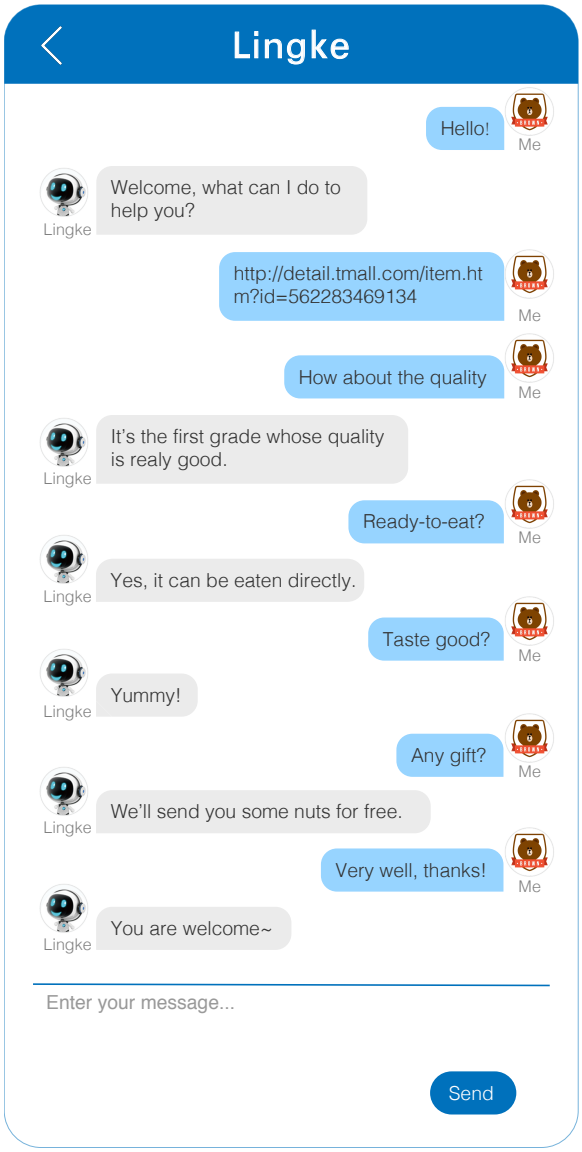

Figure 4: A conversation record based example.

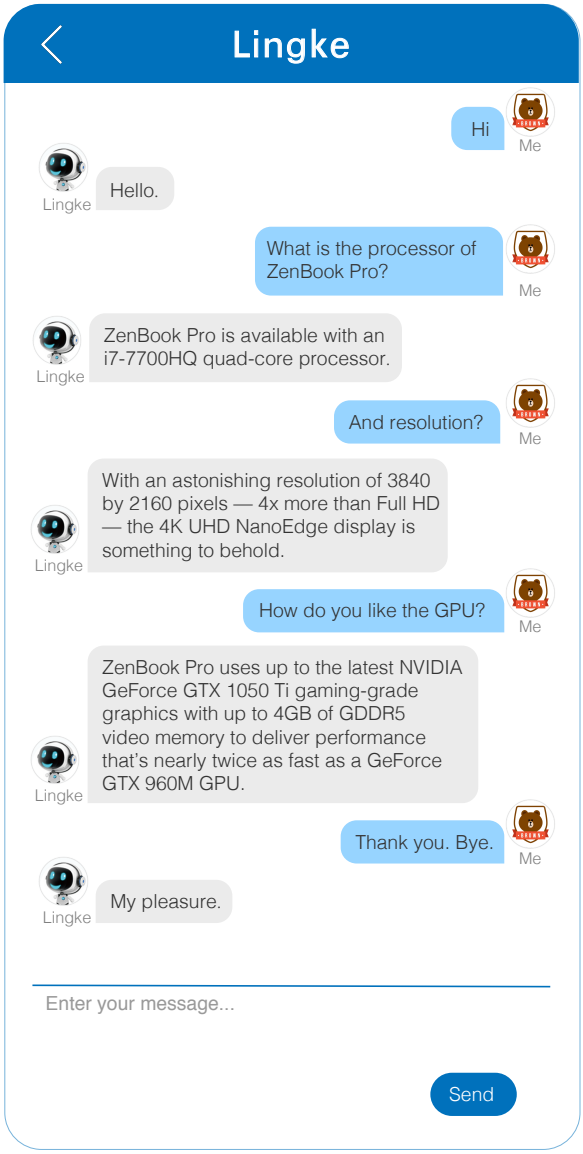

Figure 5: A document-based example.

recognizes it, and when the customer asks production specifications Lingke will give responses based on information from the context and the conversation record. Figure 5 shows a typical scenario when a customer consults Lingke about a new product. The customer starts with a greeting, which is answered by chit-chat engine. Then the customer asks certain features of a product. Note that the second response comes from a sentence which has a redundant clause, and main information the customer cares about has been extracted. In the third user utterance, words like "What" and "ZenBook Pro" are omitted, which can be deducted from the prior question. Such pivotal information from the context is distilled and utilized to determine proper response with the merit of self-matching attention and multi-turn modeling.

The user utterances of examples in this paper and our online demo are relatively simple and short, which usually aim at only one feature of the product. In some cases, when the customer utterance becomes more complex, for example, focusing on more than one feature of the product, Lingke may fail to give complete response. A possible solution is to concatenate two relevant candidate responses, but the key to the problem is to determine the intents of the customer.

5 Conclusion

We have presented a fine-grained information retrieval augmented chatbot for multi-turn conversations. In this paper, we took e-commerce product introduction as example, but our solution will not be limited to this domain. In our future work, we will add the mechanism of intent detection, and try to find solutions of how to deal with introduction document that contains more than one object.

References

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014*, pages 1724–1734.

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

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP 2011*, Edinburgh, Scotland, UK, July 27-31.

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: neural-based chinese pinyin aided input method with customizable association. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), System Demonstration*.

Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved deep learning baselines for Ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL 2015*, pages 285–294.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL 2014, Demo*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Minghui Qiu, Feng Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. Alime chat: A sequence to sequence and rerank based chatbot engine. In *ACL 2017*, pages 498–503.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *ACL 2017*, pages 189–198.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL 2017*, pages 496–505.

Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016a. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *ACL 2016*, pages 516–525.

Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2016b. Building task-oriented dialogue systems for online shopping. In *AAAI-17*, pages 516–525.

Wei Nan Zhang, Ting Liu, Bing Qin, Yu Zhang, Wanxiang Che, Yanyan Zhao, and Xiao Ding. 2017. Benben: A Chinese intelligent conversational robot. In *ACL 2017, Demo*, pages 13–18.

Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2018a. Subword-augmented embedding for cloze reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *EMNLP 2016*, pages 372–381.