AliMe Assist: An Intelligent Assistant for Creating an Innovative E-commerce Experience

Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, Wei Chu
Alibaba Group
fenglin.lfl@alibaba-inc.com

ABSTRACT

We present *AliMe Assist*, an intelligent assistant designed for creating an innovative online shopping experience in E-commerce. Based on *question answering* (QA), *AliMe Assist* offers assistance service, customer service, and chatting service. It is able to take voice and text input, incorporate context to QA, and support multi-round interaction. Currently, it serves millions of customer questions per day and is able to address 85% of them. In this paper, we demonstrate the system, present the underlying techniques, and share our experience in dealing with real-world QA in the E-commerce field.

KEYWORDS

Question Answering, Convolutional Neural Network, Knowledge Graph, Semantic Normalization, Sequence-to-Sequence, Rerank

1 INTRODUCTION

During the last few years, *question answering* (QA) based intelligent assistants have been very popular – partly due to the progresses achieved in deep learning and big data techniques, and partly due to the growing requirements in the real-world.

Customer service is one of the promising fields that intelligent assistants can play a key role in. On one hand, there is a strong demand for customer service staff in the E-commerce (EC) filed along with the fast-growing EC market. For example, according to the 2016 (first half) Annual Data Monitoring Report, the GMV of China EC market was 10.5 trillion with an increase of 37.6% year-on-year. On the other hand, with more and more people paying attention to shopping experience and service quality nowadays, the issues of traditional customer service are becoming obvious, e.g., limited availability (only office hours), inefficiency (customers often have to wait minutes), and long turn-on time, especially during peak period (e.g., the "Double 11" day).

This raises an important question: is it possible to design an intelligent assistant that can relieve customer service staff from answering simple, common and repetitive questions, and let them focus on the cases that really need human participation? If so, we can largely facilitate productivity, improve user experience and reduce cost.

We have been working on this topic for years. Our intelligent assistant, *AliMe Assist*, offers three kinds of services: assistance service, customer service and chatting service. Moreover, it is able to take voice and text input, incorporate context to QA, and support multi-round interaction. Currently, it serves millions of customer questions per day (mainly Chinese, also some English) and is able to address 85% of them. In this paper, we present the system and the underlying techniques, and share our experience in dealing with real-world QA scenarios in the E-commerce industry.

This work makes the following contributions:

- Designs and develops a real-world industrial intelligent assistant that offers customers with assistance service, customer service and chatting service in E-commerce.
- Presents a *Convolutional Neural Network* (CNN) model that incorporates the context of customer questions for intention identification.
- Proposes a *semantic normalization* and *knowledge graph* based approach for knowledge-oriented customer service question answering.
- Proposes a hybrid approach that uses an attentive *Sequence-to-Sequence* model to optimize the joint results of information retrieval and generation model for chatting.

The rest of the paper is structured as follows: Section 2 presents an overview of the system; Section 3 discusses the system features and the underlying techniques; Section 4 gives a demonstration of the system; Section 5 reviews related work, and Section 6 concludes the paper and sketches directions for future work.

2 SYSTEM OVERVIEW

We show the architecture of *AliMe Assist* in Fig. 1. The first layer is the input layer, which supports voice and text input from multi-end (e.g., mobile phone, pad, PC); the second layer sketches the intention layer, which determines the routing of each question (e.g., assistance service or customer service?); the third layer illustrates the components used for processing questions; and, finally, the forth layer stands for the knowledge source (QA pairs and knowledge graph), from which answers are retrieved.

We show the processing flow of customer questions in *AliMe Assist* in Fig. 2. Given an input question q, it first passes through the *business rule parser*, a *trie*-based pattern matcher. If q matches certain pattern(s), it will be judged as follows: if it requests for task-oriented assistance service, e.g., "我想订机票 (I want to book a flight ticket)", it will be handled by a *slot filling engine*; if it asks about promotional activities, e.g., "红包入口在哪里? (what is

CIKM'17, November 6–10, 2017, Singapore.

ISBN 978-1-4503-4918-5/17/11...$15.00
DOI: http://dx.doi.org/10.1145/3132847.3133169

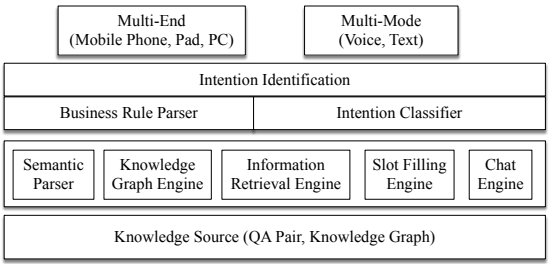

Figure 1: The overall architecture of *AliMe Assist*.

the entry of grabbing red envelopes?)", a pre-configured answer will be returned; if it is for online service staff, e.g., "呼唤真人 (real person, please)", *AliMe Assist* will ask the user to provide a description of her/his problem. If q does not match any pattern, it will be sent to an *intention classifier* for classification. That is, to be labelled with intention scenarios such as "return of sales" and "refund", depend on which q will be directed to different service staff group if human participation is needed.

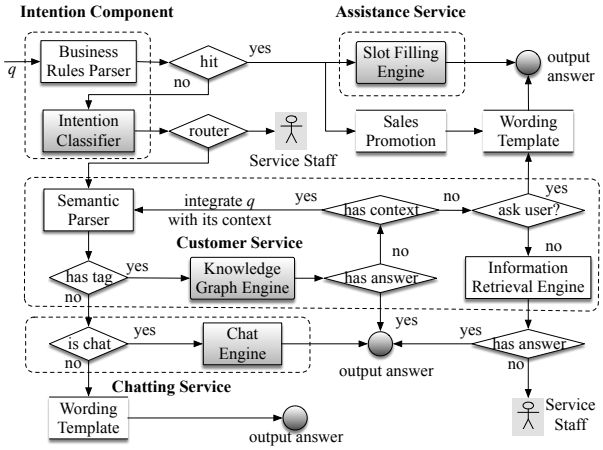

Figure 2: The overall processing flow of *AliMe Assist*

Next, q will be fed into a *trie*-based *semantic parser*. If any semantic tag (entity in knowledge graph, e.g., "customer account") is identified, q will be treated as business related (knowledge-oriented), and the identified tags will be used to retrieve answer through a *knowledge graph engine*. If no answer is retrieved, *AliMe Assist* will enrich q with its context (the previous question) and sent the concatenation back to the *semantic parser* again.

There are two points to be noted. First, the question q may not have a context (i.e., q is the first question of a session). Second, it could be the case that there is still no answer even with the concatenation. In both cases, *AliMe Assist* will move to the next step (i.e., it uses the context of q only once, if exists): asking customers for more information if the identified tags include only one entity or action, otherwise passing on q to the *retrieval engine*. In case the retrieval model still fails to give an answer, q will be transferred to service staff according to the labelled intention scenario (our *recommendation engine* is omitted here due to space limitation).

On the other hand, if q is irrelevant to business and is recognized as a chat, a *chat engine* will be employed to provide a response. If q is not a chat, a pre-configured answer will be outputted.

3 SYSTEM FEATURES

3.1 Intention Identification

User intention in *AliMe Assist* is classified into three categories: (1) requesting for assistance, e.g., "I want to book a flight ticket"; (2) asking for information or solution, e.g., "how to find back my password?"; (3) chatting (e.g., "I am unhappy"). Each category is further specialized according to supported biz scenarios, e.g., assistance service supports flight booking, mobile recharge, etc.

The two components, *business rule parser* and *intention classifier*, work together in identifying the intention of each customer question. Our business rule parser is built on hundreds of thousands of patterns obtained from frequent itemset mining and a *trie*-based pattern matcher. Our intention classifier is built on CNN [5], as shown in Fig. 3.

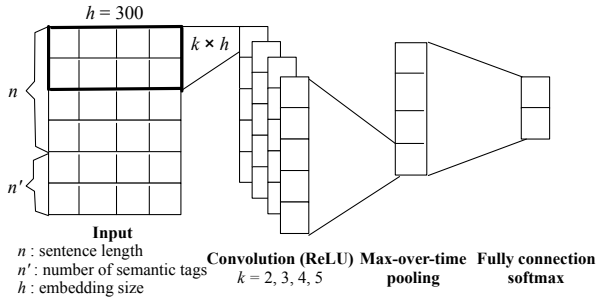

Figure 3: Intention classification with CNN

The input of our convolutional intention model is the embedding of each word of a given question q, followed by that of the semantic tags identified from q and its previous question (i.e., its context). The embeddings are pre-trained using FastText [2] and further fine tuned in the CNN model. Our experiment data includes 40 intention scenarios and 67,797 instances (47,455 for training and 20,342 for testing). With semantic tags, our CNN model is able to achieve a precision of 89.91%, which is 0.91 percent higher than that without them (89%). Both models are much better than our baseline one (an ensemble of SVM and MaxEnt, 82.71%).

We choose one layer convolution-pooling CNN instead of Recurrent NN (e.g., LSTM) because: (1) CNN can also capture the context information of words in text (before and after) and is able to achieve a "good enough" result in our case; (2) it is efficient and is able to support a QPS (query per second) around 200 in our industrial setting. More convolution-pooling layers or RNN is able to achieve better results, but has a poorer scalability.

3.2 Task-Oriented Assistance Service

Assistance service scenarios often require customers to provide information for certain attributes/slots in order to finish a specific task. For example, to book a flight ticket, one needs to provide the departure, the destination and the date. Our solution to such task-oriented QA is to first define a schema that specifies the mandatory and optional slots for a task, and then use *slot-filling* to extract from customer inputs and fill in values for the predefined slots. Our *slot-filling* engine is mainly built on dictionaries and patterns, and is able to identify fifteen kinds of attributes, e.g., product, location and date. *AliMe Assist* will ask customers for mandatory information, and request third-party services to complete the task.

3.3 Knowledge-Oriented Customer Service

In *AliMe Assist*, customer questions looking for information/solutions need to be addressed as precise as possible. We employ *knowledge graph* to tackle such knowledge-oriented questions.

The building blocks of our knowledge graph are *entities* and *relations*. We first extract candidate nouns and verbs from natural language knowledge items through word segmentation, part-of-speech (POS) tagging and tf-idf filtering, and use them to construct high-order (a combination of several, usually two) entities based on *mutual information*. We then have business analysts review the entities, and design relations to build a hierarchical structure. Finally, we adopt Neo4j [1] as our query engine. Our knowledge graph includes several thousands of entities and a fixed set of relations, and supports short (one- or two-) hop reasoning. For example, as shown in Fig. 4, if one asks about "how to find my lost login password" which does not have an associated knowledge item, we can provide a generalized but also useful answer by returning the knowledge item associated with its parent node "find lost password".

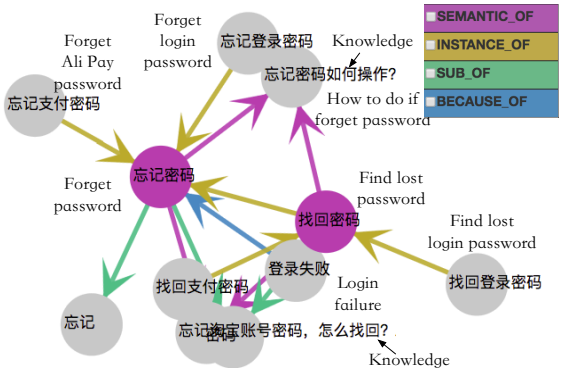

Figure 4: An excerpt of a knowledge graph

In practice, *semantic normalization* – mapping different kinds of utterances to a semantically equivalent entity in the knowledge graph – is of key importance as customer questions are often highly diversified. We tackle this problem through utterance diversification and pattern mining. In the first stage, given a set of knowledge items K, we identify its diversified utterances through finding similar answers K' from historical chat logs (between customers and service staff) and taking the corresponding questions for K'. The similarity calculation algorithm is designed based on sentence embedding [6], which helps to capture semantic similarity, and implemented using Map-Reduce [3]. Once we have the mapping between a knowledge item and a set of diversified utterances, we are able to extract wording patterns for entities using frequent itemsets mining. These patterns will be used in the trie-based semantic parser to identify semantic tags (again, entities in knowledge graph) for each customer question.

Compared with traditional information retrieval (IR) model, our knowledge graph approach increases accuracy by 10%. Since our knowledge graph covers only frequently asked (i.e., a small subset of) knowledge items, retrieval model is used as a complement to improve recall.

[1] https://neo4j.com/

3.4 Chatting Service

In *AliMe Assist*, the majority of customer questions is business-related, but also around 5% of them is chat-oriented (several hundreds of thousands in number). To offer better user experience, we build an open-domain chatbot engine, where we propose a hybrid approach that uses an attentive *Sequence-to-Sequence* (Seq2Seq) model to optimize the joint results of IR models [15] and Seq2Seq generation models [10, 11].

The overview of our approach is shown in Fig. 5. First, a retrieval model is used to collect candidate answers. Second, an attentive Seq2Seq model is employed to rerank the candidates by giving each candidate a confidence score. Top candidate with a confidence score that exceeds a given threshold can be used as an answer. If it is lower than the threshold, the generated answer by Seq2Seq model will be taken as the output. Interested readers can refer to [7] for details.

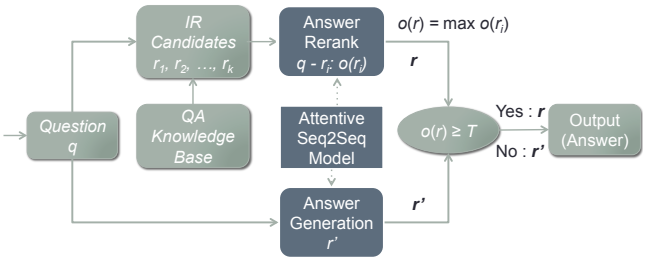

Figure 5: An overview of our hybrid chat approach

Extensive experiments show that our approach significantly outperforms IR and generation models: in a test with 600 questions, the P_{top_1} (whether the top-1 candidate is acceptable) of our approach, IR and Seq2Seq generation are 60.01%, 47.11% and 52.02%; an online A/B test (2136 questions, 1047 by the hybrid approach, 1089 by IR) also confirms that our hybrid approach performs much better than IR (P_{top_1}: 60.36% vs. 40.86%).

4 DEMONSTRATION

We launch *AliMe Assist* for a real-world industrial intelligent assistant [2], which currently serves millions of customer questions per day and is able to address 85% of them.

We demonstrate the key features, namely assistance service, customer service and chatting service, through three realistic scenarios in Fig 6. Fig 6 (a) illustrates how *AliMe Assist* helps users to book a flight ticket through asking for needed information step by step, and then calling Ali Trip, a travel agency of Alibaba Group, to complete the task. Fig 6 (b) demonstrates how *AliMe Assist* helps to address customer service questions by combining the previous question "我想查看 (I want to check)" and the current one "淘宝账户 (Taobao account)", and then employing the knowledge graph engine to provide an answer. Fig 6 (c) shows a chat conversation between *AliMe Assist* and customers. These three scenarios together demonstrate the intention identification feature: given a question, *AliMe Assist* is able to identify its intention, and distribute it to the corresponding service (e.g., assistance service) and scenario (e.g., booking a flight).

[2] Interested readers can access *AliMe Assist* through the Taobao App by following "我的淘宝 (My Taobao) → 我的小蜜 (My AliMe)", or the web version via https://consumerservice.taobao.com/online-help (text only, assistance service not enabled).

(a) Assistance Service (b) Customer Service (c) Chatting Service

Figure 6: A demonstration of *AliMe Assist*

5 RELATED WORK

In this section, we review related work on task-, knowledge-, and chat-oriented QA, which are closely related to the techniques employed in *AliMe Assist*.

Task-oriented QA. This belongs to closed domain QA. Typically people use rule- or template- based methods [12], and dialog state tracking [4]. Our slot-filling method is similar to the template-based method.

Knowledge-oriented QA. This technique can be used in both closed and open domain. Usually, data-driven IR model is used here. The idea is to find the nearest question(s) from a QA knowledge base for each input question, e.g., [15]. A recent work [14] has tried a neural network based method for matching. Being different from these methods, we uses both knowledge graph and IR: the former is designed for high precision QA in E-commerce and the latter is used as a complement.

Chat-oriented QA. Commonly used methods for this open domain QA include IR [15] and generation models [1, 10]. Our approach alleviates the flaws of both IR and Seq2Seq generation models: the former can only handle questions that are close to those in a QA knowledge base, while the latter may generate inconsistent or meaningless answers [8]. Another recent combinational approach [9] uses an IR model to rerank the union of retrieved and generated answers. Our work differs from it in that we use an attentive Seq2Seq rerank approach to optimize the joints results of IR and generation models.

6 CONCLUSION

In this paper, we present *AliMe Assist*, an intelligent assistant that is designed for creating an innovative E-commerce experience. We launch *AliMe Assist* for a real-world application that offers customers with assistance service, customer service and chatting service, and currently handles millions of customer questions per day.

As for future work, several points will be further explored to improve our assistant. For example, strengthening context based multi-round interaction [13], offering shopping guidance based on Reinforcement Learning (RL), empowering *AliMe Assist* with the capability to "read" images through image recognition, etc.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR'15*.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint* (2016).

[3] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[4] Matthew Henderson. 2015. Machine Learning for Dialog State Tracking: A Review. In *MLSLP'15*.

[5] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint* (2014).

[6] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[7] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *ACL'17*.

[8] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI'16*. 3776–3784.

[9] Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. *Two are Better than One: An Ensemble of Retrieval- and Generation-Based Dialog Systems.* arxiv preprint.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS'14*. 3104–3112.

[11] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. In *ICML DL Workshop'15*.

[12] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. *A network-based end-to-end trainable task-oriented dialogue system.* arXiv preprint.

[13] Yu Wu, Wei Wu, Ming Zhou, and Zhoujun Li. 2016. Sequential Match Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots. *arXiv preprint* (2016).

[14] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR'16*. 55–64.

[15] Zhao Yan, Nan Duan, Jun-Wei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016. DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. In *ACL'16*.