Automatic Question Answering System for Consumer Products

Seunghyun Yoon[1,3](✉), Mohan Sundar[2], Abhishek Gupta[2], and Kyomin Jung[1,4]

[1] Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
{mysmilesh,kjung}@snu.ac.kr

[2] Samsung R&D Institute India, Bangalore, India
{sundar,abhishek.gu}@samsung.com

[3] Samsung Software R&D Center, Seoul, Korea

[4] Automation and Systems Research Institute, Seoul National University, Seoul, Korea

Abstract. In this paper, we present a novel automatic question answering system being the first research to deal with various types of user questions in consumer electronics domain. We gathered more than 185,000 actual data pairs and 50,000 data triples from Samsungs official web site and a crowd question answering web site in order to provide proper answers to user queries. With this system, users can resolve their problems in using consumer electronic gadgets at any time with a guaranteed response time and answer quality by simply asking questions to the system in a natural language form. The system architecture and test results are addressed in the paper.

Keywords: Question answering · Natural language processing · Information retrieval

1 Introduction

Recently, many consumer products improve their adaptability via connections to the web, and launch new functionalities for providing a better service to users. However, sometimes a user does not understand the features in his device and needs somebody to help him. When customers face trouble in using their products, they usually connect to the customer support center for resolving the problem. To support those customers, a company needs well trained human agents to provide a prompt and accurate response [1], which is not easy to maintain for several reasons. When a company releases new products or services, all agents need to be trained in the newly updated information quickly. Moreover, a company usually keeps a limited number of agents in order to save on operation costs. A sudden increase in user queries may cause customers to wait a long time for a response. To solve these problems, we propose an IT meditated automatic

Y. Bi et al. (eds.), *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Lecture Notes in Networks and Systems 16, DOI 10.1007/978-3-319-56991-8_78

question answering system for supporting user needs in using their own products. The system can support a sudden increase in queries by scaling out service servers using cloud service such as amazon, and can provide a high level of answers to user questions all the time, whereas the quality of answers solely relies on the level of personal expertise in the legacy human agent system.

The question answering (QA) system uses structured and unstructured data to provide answers to user questions. Intelligent voice agents such as S-Voice of Samsung Electronics and Siri of Apple use pre-defined structural data. When these services receive a query from the user, they try to match it with pre-defined actions and execute that action. Another type of research analyzes user questions and retrieves relevant information from indexed data to provide results to the user. The AnswerBus system implements a QA system based on sentence level Web information retrieval by using several search engines and achieves 70.5% accuracy for factoid type of question [2].

One of the most intelligent approaches to the QA system is using both structured and unstructured data after analyzing user queries. The IBM Watson system extracts structured information from unstructured data and builds a knowledge base [3]. It also indexes unstructured data using search clusters [4]. However, no previous research has been applied to a QA system in the consumer electronics domain, as they have mainly focused on the factoid type of questions, whereas users have a more diverse type of query, such as procedural questions, how to questions, and troubleshooting issues.

In this paper, we present a novel automatic QA system for the consumer products. With this system, user's questions can be solved in a natural language form at any time.

2 System Architecture

2.1 Offline Data Processing

All the data used in the QA system are generated and updated in an offline system for reducing service response time. To create a proper answer to user questions, all the related data needs to be gathered. We gathered data from Samsung's official web site and a crowd question answering web site. The gathered data was pre-processed and categorized into QA type, document type, and triple type. The QA type consists of user questions and matched answers, such as frequently asked questions and troubleshooting data. The document type consists of titles and related passages such as passages in user manuals. The triple type consists of subject, predicate, and objects such as product specification data. Additionally, automatic question generator was developed by organizing FAQ, howto, user manual and troubleshoots data into small passages (with heading hierarchy) and then applying Named Entity Recognition & Part-of-speech tagging to generate question templates for various classes of questions: what, how, why and where ("what-definition", "what-options, "what-features", "how-crud", "how-operation", "how-communication", "why-reason", "where-location"). Those passages which could not be classified into the defined

Table 1. Offline data statistics

Data engine	Category	Total number
Search index	QA Type	166,309
	Doc Type	19,165
Knowledge graph	Triple Type	50,783

categories using templates were given to an SVM-classifier to classify into respective categories and then apply templates to generate rules. After manual evaluation of about 45 K generated questions, it is observed that 49% were correct questions, 43% partially correct with grammatical errors or missing words and 8% were incorrect questions. The generated data were considered as QA Type data. To enhance the online system performance, named entities, product category, and triple information in the data were pre-processed and indexed. Table 1 shows statistics about offline data we generated for this research.

2.2 Online System

Figure 1 briefly shows the online system architecture. When the system receives a user query, the domain detector checks the question and filters out the domain un-matching cases. A question analysis module runs a natural language process and classifies question types and answer types. It also tokenizes the question and finds named entity information from each token. We built a named entity dictionary by analyzing an n-gram using a domain corpus in an offline system. The reasoning module expands information of the tokens by using a synonym and hypernym dictionary and FrameNet [5] for later use. An information retrieval (IR) module generates queries by using passed-over information from the previous modules and executes it to a target database such as a knowledge graph, a document search engine, and a passage search engine. To increase the recall rate, we use multiple queries expanded by using tokens, sentences, part of speech information and predicate information. An answer extraction module extracts sentences from the data retrieved from the IR module, and generates candidate answers. All the prepared candidate answers are evaluated and the final answer is selected in the Re-ranking module.

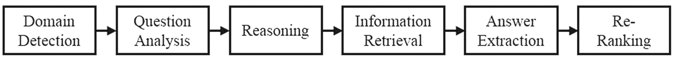

Fig. 1. Online QA system architecture

3 Experiments

A total of 230 questions were used for evaluation (100/65/65, real user question, crowd question answering site, official site, respectively). Figure 2 shows the evaluation results. The Y-axis shows the accuracy of the answer for the user question,

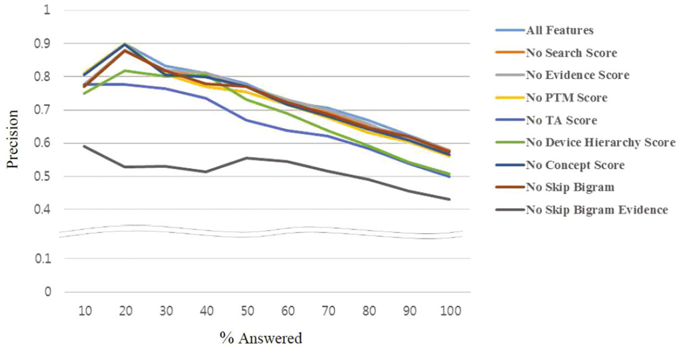

Fig. 2. Impact of scoring features. The Y-axis shows the accuracy of the answer for the user question, and the X-axis represents the percentage of user questions answered.

and the X-axis represents the percentage of user questions answered. The highest accuracy is 0.891 at 20% answered point, and the highest F1-score is 0.732 according to the graph. It also shows the impact of the scoring feature in the re-ranking module. Each feature measures the similarity between user questions and candidate answers, whereas the evidence score feature measures the similarity between user questions and the title of the retrieved data. The search score feature is calculated by using the term frequency inverse document frequency (TF-IDF) value between query and answer. The device hierarchy score feature calculates the product hierarchal distance among the products in the user's question and answer. The concept score feature measures the similarity by using Latent Dirichlet Allocation (LDA) [6]. We used Jensen-Shannon based similarity measure between concepts of the query and that of the retrieved documents. Other scores' features are measured by using passage term matches (PTM), textual alignment (TA) and skip bigram score [7]. The three major contributors are skip bigram evidence, TA score and device hierarch score.

4 Conclusion

We present a novel automatic question answer system for the consumer electronics domain. With this system, users can resolve problems and questions related to their device at any time in a natural language form. The system evaluation results show an F1-score of 0.732. In the future work, we aim to achieve better accuracy of the system and apply them to real world commercial QA systems.

References

1. Kazmer, M.M., Burnett, G., Dickey, M.H.: Identity in customer service chat interaction: implications for virtual reference. Libr. Inform. Sci. Res. **29**(1), 5–29 (2007)
2. Zheng, Z.: Answerbus question answering system. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 399–404. Morgan Kaufmann Publishers Inc. (2002)
3. Fan, J., Kalyanpur, A., Gondek, D., Ferrucci, D.A.: Automatic knowledge extraction from documents. IBM J. Res. Dev. **56**(3/4), 1–5 (2012)
4. Chu-Carroll, J., Fan, J., Boguraev, B., Carmel, D., Sheinwald, D., Welty, C.: Finding needles in the haystack: search and candidate generation. IBM J. Res. Dev. **56**(3.4), 1–6 (2012)
5. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of the 17th International Conference on Computational Linguistics, vol. 1, pp. 86–90. Association for Computational Linguistics (1998)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
7. Murdock, J.W., Fan, J., Lally, A., Shima, H., Boguraev, B.: Textual evidence gathering and analysis. IBM J. Res. Dev. **56**(3/4), 8:1–8:14 (2012)