Information Processing and Management 48 (2012) 83–93

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Effects of answer weight boosting in strategy-driven question answering

Hyo-Jung Oh [a,*], Sung Hyon Myaeng [b], Myung-Gil Jang [a]

[a] Electronics and Telecommunications Research Institute (ETRI), 161 Gajeong-dong, Yuseong-gu, Daejeon 305-700, Republic of Korea
[b] Korea Advanced Institute of Science and Technology (KAIST), 119 Munjiro, Yuseong-gu, Daejeon 305-732, Republic of Korea

ARTICLE INFO

Article history:
Received 25 June 2009
Received in revised form 22 July 2010
Accepted 30 January 2011
Available online 23 February 2011

Keywords:
Question answering
Strategy-driven QA
Answer verification and weight boosting

ABSTRACT

With the advances in natural language processing (NLP) techniques and the need to deliver more fine-grained information or answers than a set of documents, various QA techniques have been developed corresponding to different question and answer types. A comprehensive QA system must be able to incorporate individual QA techniques as they are developed and integrate their functionality to maximize the system's overall capability in handling increasingly diverse types of questions. To this end, a new QA method was developed to learn strategies for determining module invocation sequences and boosting answer weights for different types of questions. In this article, we examine the roles and effects of the answer verification and weight boosting method, which is the main core of the automatically generated strategy-driven QA framework, in comparison with a strategy-less, straightforward answer-merging approach and a strategy-driven but with manually constructed strategies.

1. Introduction

Question answering (QA) is considered a key technology for handling information overload caused by the ever increasing amount of information. With the advances in natural language processing (NLP) techniques and the need to deliver more fine-grained information or answers than a set of documents returned by a search engine, various QA techniques have been developed corresponding to different question and answer types. It is obvious that future QA systems will be equipped with more and more capabilities to handle different types of advanced questions and with more effective technologies for the existing capabilities. To make best use of additional QA technologies, a QA framework was proposed where the invocation sequence of multiple QA modules was determined by a learned strategy for the particular type of question at hand (Oh, Myaeng, & Jang, 2009). This way, individual QA techniques can be integrated to maximize the system's overall capability in handling diverse types of questions.

The main core of the strategy-driven QA method is to invoke individual QA modules in the sequence based on the learned strategy and boost the weight of returned answers. A set of strategies are learned automatically from the past history of QA operations involving individual QA modules, which reflect how well different QA techniques provided correct answers for different types of questions. Strategies prescribe not only the sequence of invocation but also ways to verify an answer returned from a QA module and boost its weight if necessary.

The main benefit of the strategy-driven QA method lies in finding answers more accurately with the reduced number of transactions when compared against methods without a strategy and a method with manually constructed strategies (Oh et al., 2009). More specifically, such a strategy-driven QA system with multiple QA modules can:

* Corresponding author. Tel.: +82 42 860 5405; fax: +82 42 860 4889.
E-mail addresses: ohj@etri.re.kr (H.-J. Oh), myaeng@kaist.ac.kr (S.H. Myaeng), mgjang@etri.re.kr (M.-G. Jang).

0306-4573/$ - see front matter
doi:10.1016/j.ipm.2011.01.010

– provide an answer with an increased level of confidence because an answer returned by a QA module can be verified by another,
– improve efficiency, without sacrificing the quality (i.e. effectiveness), compared to the straightforward approach of blindly routing the question to all the available QA modules and combining the results, and
– save efforts in manually constructing strategies and therefore make it easier to incorporate newly available QA techniques

Building on the previous work that proposed the method and showed its overall values, this article investigates the effects and the roles of the answer verification and weight boosting method, which is the main core of the strategy-driven QA framework, in comparison with a strategy-less, straightforward answer-merging approach and a strategy-driven but with manually constructed strategies. The main contribution of this paper is detailed analyses of the strategy-driven QA framework through a series of experiments, from which we obtain new insights on its internal workings and its sensitivity to key elements such as question types, the number of QA modules, and types/amounts of errors made by QA modules.

2. Related work

Some recent research attempts to break away from the strict single pipeline architecture of traditional question answering systems with the aim of improving the performance. They exploited a "multi-strategy"[1] or multi-source approach with several answering modules for different question types and showed that it gave a better performance than a single pipeline approach.

One way to utilize multiple QA modules is to route a question to all the modules as in a "meta-search" engine. A processed question is simply distributed to the individual QA modules in parallel, and then the results are merged with re-ranking. While most multi-technique based systems (Chu-Carroll, Pager, Welty, Czuba, & Ferrucci, 2002; Katz et al., 2004; Lin & Katz, 2003) adopt this straightforward strategy, obvious weaknesses of this approach are the inefficient use of resources, especially with a large number of QA agents, and the one-for-all result combination strategy. One can also argue that getting the same answer from multiple sources would increase its confidence level (Clarke, Cormack, & Lyman, 2001; Nyberg et al., 2004). For example, Clarke et al. (2001) obtained 18% improvement (23–41%) using the redundancy. However, this type of redundancy may not be necessary all the time.

Another direction is to "hard-code" answering strategies manually. When implementing a QA system, the developer has to understand the capability of individual QA agents and carefully craft the answer selection strategy for the given agents. This method seeks to approximate domain-specific expert knowledge by employing human experts to develop combining strategies (Hickl, Williams, & Harabagiu, 2004; Oh, Lee, et al., 2006). PowerAnswer3 and CHAUCER (Hickl, Williams, Bensley, Roberts, & Rink, 2007; Moldovan, Bowden, & Tatu, 2007) are a fully modular, distributable, "strategy"-based QA system that is configurable. Each system consists of a set of "strategies" (modules in our terminology), one or two of which are selected based on the question processing result, to solve different classes of questions either independently or together. Aside from the fact that it is more like a one-best approach, there is no explicit mechanism for answer verification and weight boosting.

QA systems described above assign a suitable "strategy" (a QA module in our terminology) for a user question based on some built-in rules or scenarios. The process of manually determining the strategies and setting some parameters for combining multiple results requires a significant amount of human efforts in understanding the capabilities of and the interplay among different techniques. The problem is not only the initial cost but also the lack of extensibility and flexibility for additional QA modules to be added at a later stage. When a new technique is added, the entire set of strategies must be re-configured manually.

The work in (Oh et al., 2009) is unique in that the question analysis process determines a strategy for a module invocation sequence and result integration, not just a single module or all modules to which the question is sent. The answers from the selected modules are verified and combined based on the strategy that was learned automatically. While the paper introduces the new method and shows its efficacy in comparison with the other approaches in an experiment, it treats the method as a black box in the experimental analysis. This paper investigates the roles and effects of the answer verification and weight boosting method and attempts to provide insights on its internal workings and its sensitivity to key elements such as question types, the number of QA modules, and types/amounts of errors made by QA modules.

3. QA driven by automatically learned strategies

To provide a context in which we can analyze the roles and effects of the strategy-driven QA and the automatically learned strategies for weight boosting, we give a brief explanation of the underlying QA system equipped with a strategy learning and execution method utilizing multiple QA. The right hand side in Fig. 1 corresponds to the offline process of text analysis for answer generations and of strategy learning. The other half on the left roughly corresponds to the process of QA with a user question that needs to go through the *Question Analysis* module. The result from the module is passed to the

[1] The term "strategy" sometimes refers to a particular QA source or technique in previous work, making "multi-strategy" mean more than one QA module in a system. To avoid confusion, we use "module" for a QA source and "strategy" for a sequence of QA module invocations.