Which is the best?: Re-ranking Answers Merged from Multiple Web Sources

Hyo-Jung Oh, Pum-Mo Ryu, Hyunki Kim
Knowledge Mining Research Team, Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, Korea (305-700)
{ohj, pmryu, hkk}@etri.re.kr

***Abstract*—The main motivation of this paper is to devise a way to select the best answers collected from multiple web sources. Depending on questions, we need to combine multiple QA modules. To this end, we analyze real-life questions for their characteristics and classify them into different domains and genres. In the proposed distributed QA framework, *local* optimal answers are selected by several specialized sub-QAs. For fining *global* optimal answers, merged candidates are re-ranked by adjusting confidence weights based on the question analysis. We adopt the idea of the margin separation of SVM classification algorithm to adjust confidence weights calculated by own ranking methods in sub-QAs. We also prove the effects of the proposed re-ranking algorithm based on a series of experiments.**

***Index Terms*—Re-ranking, Multiple Web sources, Question Answering**

I. Introduction

Depending on questions, various answering methods and answer sources can be used. To find the answer for general questions in a simple factoid-style, many systems of TREC (Text Retrieval Conference, [1]) adopted statistical answering methods [2]. For some questions that ask record information such as "Which is the longest river?", finding the answer in a specific corpus like the Guinness Book is more effective. Otherwise, knowledge bases can be used to answer definition questions such as "Who is J. F. Kennedy?"

One can argue that the same answer from multiple sources would increase the confidence level [3,4]. Depending on the type of question and the nature of Question Answering (QA) module, however, this type of redundancy may not be necessary [5]. For example, a question like, "When was Madam Curie born?", can be answered without ambiguity in an encyclopedia-based QA system, if an answer exists, because it can be handled by a pre-constructed knowledge base. Besides, multiple answers may end up lowering the confidence level of the correct answer if a straightforward merging method is used. We take the position that some redundancy would be useful for answer verification but should be used more judiciously for both efficiency and effectiveness.

More recent research tries to comprehend heterogeneous sources with the aim of improving the performance of QA system. PIQUANT from IBM [6] firstly proved that a multi-source approach to question answering achieve a good correlation of confidence values and correctness. A re-ranker of CHAUCER [7] compiled answers from each of the five answer extraction strategies into a single ranked list and an Answer Selection module identifies the answer which best approximates the semantic content of the original question. However, these researches are focused on the similarity of candidate answer and the given question.

As an advanced research, PowerAqua [8] explores the increasing number of multiple, heterogeneous knowledge sources available on the Web. A major challenge faced by PowerAqua is that answers to a query may need to be derived from different ontological facts and even different semantic sources and domains. To overcome this problem, they presented merging and ranking methods for combining results across ontologies [9]. However, the ontology-based method needs a lot of human efforts.

In Information Retrieval (IR) area, the base technology of QA, Wu and Marian [10] proposed a framework to aggregate query results from different sources. To return the best answers to the users, we assign a score to each individual answer by taking into account the number, relevance and originality of the sources reporting the answer. They took into account the quality of web pages.

In this paper, we build a distributed QA system to handle different types of questions and web sources. Especially, to select the best answer for a given user question, we propose an answer selection algorithm for re-ranking candidate answers from distributed multiple web sources.

To distill characteristics questions and answers, we differentiate varieties of user questions collected from commercial portal services, and distinguish a wide spectrum of potential answers from multiple web sources. Based on these observations, we build vertical sub-QA modules specialized for different domains and genres of web sources. Each sub-QA has own answer extraction methods tailed to various answer types that are identifiable from documents. They built multiple inverted index databases and distributed with Hadoop system [11].

To merge candidate answers from multiple web sources, we develop a special broker to interact with sub-QAs. When a user question is entered, the broker distributes the question over multiple sub-QAs according

doi:10.4304/jetwi.4.1.35-42

to question types. The selected sub-QAs find local optimal candidate answers, however, the weights are calculated by own ranking mechanisms so they have a big diversity.

The merged candidates are re-ranked by adjusting confidence weights based on the question analysis result. The re-ranking algorithm aims to find global optimal answers. We borrow the concept from the margin separation and slack variables of SVM classification algorithm [12, 13], and modify to project confidence weights into the same boundary by training.

In the following, we (1) discuss characteristics of questions and illustrate an overview of our distributed QA model consists of multiple sub-QAs in Section 2, (2) describe how to analyze a given user question and distribute it over corresponding sub-QAs in Section 3, (3) concentrate on the re-ranking algorithm based on the SVM training model in Section 4, (4) analyze the effect of the proposed ranking method with several experiments together with an in-depth analysis of weight distribution in Section 5. Finally we conclude with a suggestion for possible future works in Section 6.

II. Distributed Question Answering

Our ultimate goal is to build a QA system that can handle a variety of types of questions and answers. In order to make full use of various QA techniques corresponding to different types of questions it is critical to classify user questions in terms of the nature of the answers being sought after. To this end, we collected more than 7,000 questions from the commercial manual QA sites[1]. They were analyzed to characterize the types of questions and answer sources with two points of view: *domains* (or categories) and *genres*.

As shown in Table 1, the most Top-4 domains of user questions are education, game/computer, life (including travel, sports, and local), and entertainment. However, the education domain covers very broad subjects (e.g. mathematics, physics, or chemistry) and over 80% of answers of questions in education could be found in the encyclopedia or Wikipedia [5], so we substituted this domain for the Wikipedia genres. Meanwhile, we also excluded the entertainment domain since many questions belong to this domain are related with gossip or entertainer scandals, so these problems are very difficult to judge that which answers are correct or not. In this paper, four major domains are selected: game/computer, travel, local, and life. The other questions are considered as the "open" domain.

We differentiated answer sources according to genres. Because the corpus such as the Wikipedia contains facts about many different subjects or explains one particular subject in detail, there are many sentences that present definitions such as "X is Y." On the other side, some sentences describe the process of a special event (e.g. World War I), so that they consist of particular syntactic structures (5W1H) similar to those found in news documents. In blogs, there are many personal opinions or know-how for a certain problem. In this paper, major genres of web sources which have many chances to get answers for most of user question are concentrated on four types: news, blogs, Wikipedia, and the others.

TABLE I. Domain Distribution of User Questions

Domain		# of Question	Ration
Education		1,813	25.64%
Game/Computer		1,218	17.23%
Life	Local	891	12.60%
	Sports	238	3.37%
	Travel	133	1.88%
Entertainment		720	10.18%
Shopping		521	7.37%
Health		516	7.30%
Economics		510	7.21%
Social/Politics		510	7.21%
Total		7,070	

Based on overall analysis, we built a distributed QA system to have multiple sub-QA modules as shown in Figure 1. We combined eight sub-QAs: four *domain-QAs* and four *genre-QAs*. While the sub-QA modules are complementary to each other in providing answers of different types, their answer spaces are not completely disjoint.

All sub-QAs except the web-QA have own answering methods tailed to various answer types that are identifiable from documents [5]. Especially, the number of documents indexed in the News-QA and the Blog-QAs are approximately 13,800,000 and 33,600,000, respectively. Thus we needed to distribute local indexing database for efficient. As shown in Figure 1, the News-QA and the Blog-QA consist of 3 and 5 *Hadoop clusters*, respectively. For the Web-QA to complement web documents which we neglected to crawl, we used the Yahoo! API from Yahoo! Search Web Services[2].

To interact with sub-QA modules, we developed special brokers, the B1 and B2 components. The B1 combines multiple indexing blocks for a particular sub-QA and merges candidate answers, whereas the B2 communicates with multiple sub-QAs and ranks their candidates. The B2 has own ranking algorithm to find the local optimal answer for each sub-QA.

The *Answer Manager* component has two roles. The first one is to determine a user question and spread them to appropriate sub-QAs. The other is to re-rank candidate answers merged by B2 and find the best answer as the global optimal solution.

III. Distributing Questions

Based on question types, the B2 in Figure 1 can determine which sub-QAs should be involved to find the best answer. A user question in the form of natural language is entered into the system and analyzed by the

[1] the *Naver*tm Manual QA Service (http://kin.naver.com). When a user upload a question, then the other users answer the question by manually and get points depending on how satisfied with the answer by the question owner.

[2] http://developer.yahoo.com/search/web/

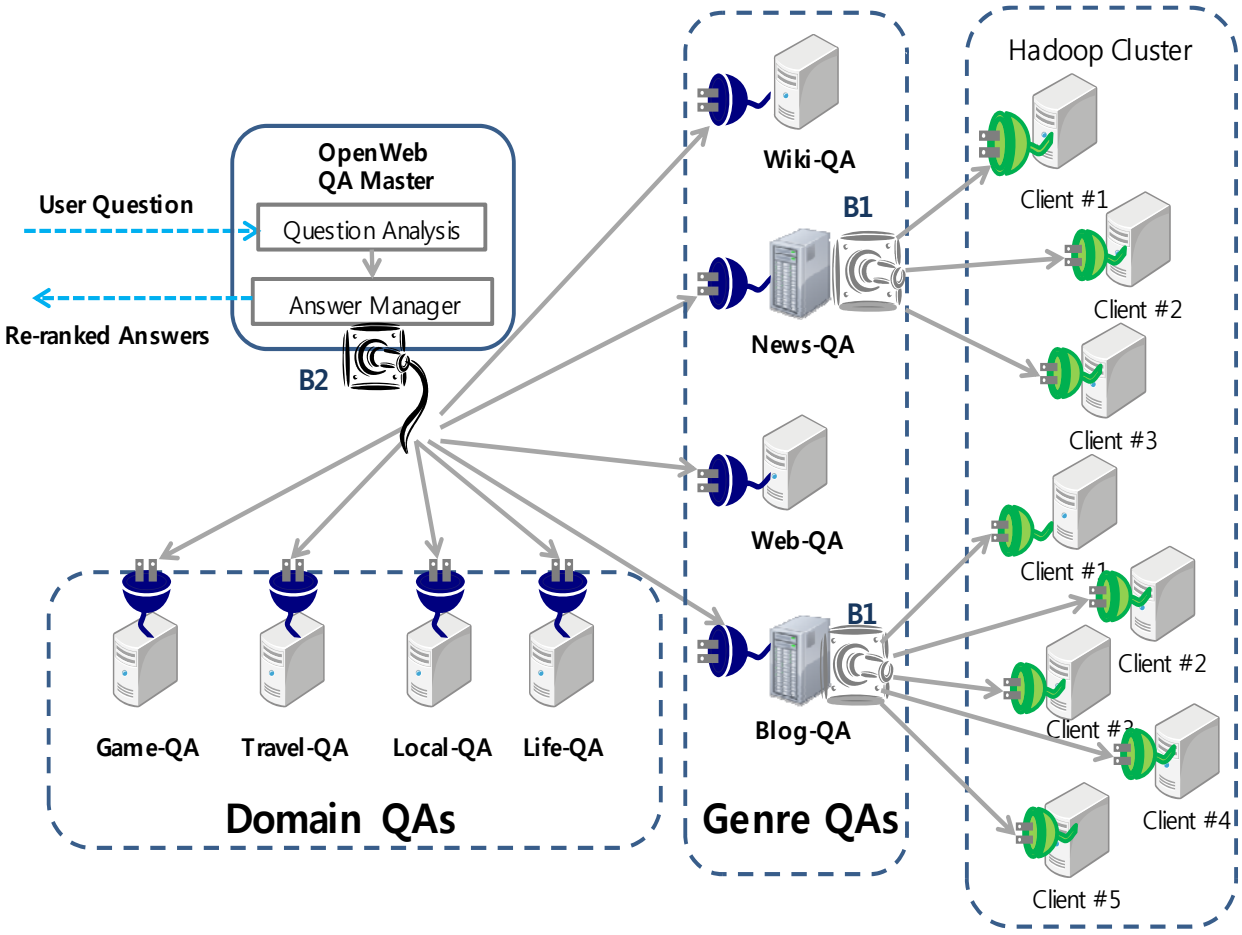

Figure 1. System Overview

Question Analysis component that employs various linguistic analysis techniques such as POS tagging, chunking, answer type (AT) tagging [9], and some semantic analysis such as word sense disambiguation [10]. An internal question generated from the Question Analysis component has the following form:

$$Q = <AF, AT, QT, AD> \quad (1)$$

where AF is the expected answer format, AT is the expected answer type, QT is the theme of the question, and AD is the domain related to the expected answer source or sub-QA module from which the answer is to be found.

- The answer format (AF) of a question is determined to be one of these four types: a single, multiple, descriptive, or yes/no question. For example, single is the AF value in the question "*Who killed President Kennedy*?"
- There are 147 fine-grained ATs organized in a hierarchical structure with 15 nodes at the level right below the root, each of which has two to four lower levels [9]. The AT gives information about the type of the entity being sought [11]. The sub-type/super-type relations among the ATs give flexibility in matching. For the example above, the AT would be "people" because of "who", which can be matched with "president" in a passage.
- A question theme (QT) has two parts: a target and a focus. The target of a question is the object or event that the question is about, whereas the focus is the property being sought by the question. In the example above, the target is "J. F. Kennedy" and the focus is "killer".
- The answer domain (AD) of a question indicates the most likely source (sub-QA module) from which an answer can be found, which is determined based on the other traits of the question (AF, AT, and QT). It also contains some detailed information about what should be sought after in the QA module. For example, the answer for the question, "When was Madam Curie born?" might be found in Wikipedia. In contrast, for "*How to play Starcraft*[3] *well using Protos*?", personal blogs or community boards for that game are more suitable.

[3] *StarCraft*[tm] is a military science fiction real-time strategy video game developed by Blizzard Entertainment©.

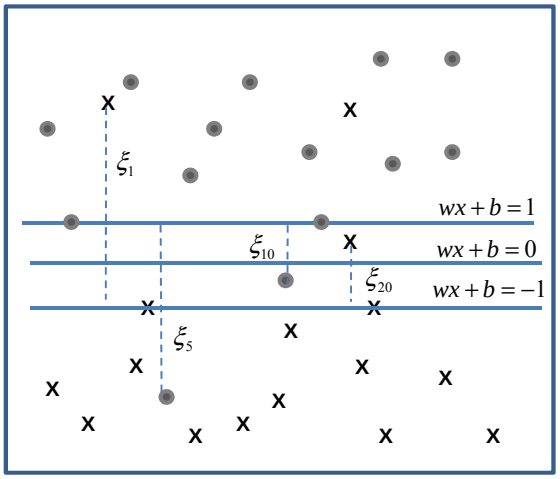

Figure 2. Modeling of SVM

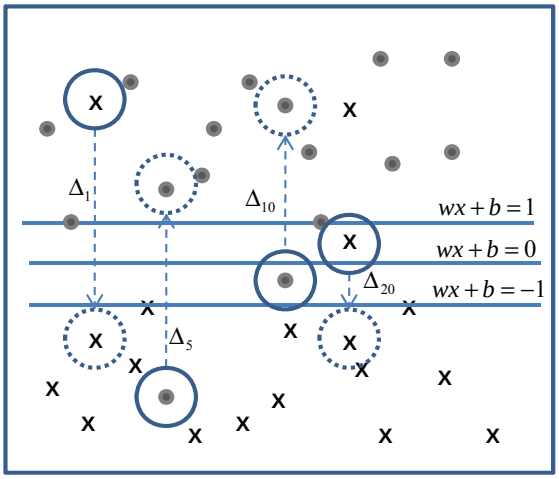

Figure 3. Examples of Adjust Slacks

Based on the question analysis, The B2 in Answer Manager invokes one or more sub-QA modules. For the former example question about Madam Cuire, the B2 might select the wiki-QA at first. If the calculated evidence of answer candidates from the wiki-QA is strong enough, then the B2 provides the answer to the user. If none of the answers from the first module have a confidence weight higher than the threshold, the B2 invokes other sub-QAs to merge candidates.

IV. Merging and Re-Ranking Answers

The local optimal answers from multiple web sources are collected in the Answer Manager. As mentioned in Section 2, each sub-QA has own ranking mechanism, so that the confidence weights of merged answers are very diverse. For example, the weights from the News-QA are between 0 and 2, while Web-QA weights are between 0 and 0.8 (refer Figure 5). To adjust these variations and to project confidence weights into the same boundary, we devise a new re-ranking algorithm. We borrow the idea from the margin separation of SVM classification algorithm [12, 13], and modify to adjust confidence weights into the same boundary by training.

Figure 2 captures correct answers and uncertain answers in the SVM model and they are marked with "●" and "x", respectively. If the training answer set D is not linearly separable, the standard approach is to allow the fat decision margin to make a few mistakes. We then pay a cost for each misclassified example, which depends on how far it is from meeting the margin requirement given in Equation (2).

$$y_i(\vec{w}^T\vec{x}_i+b)\geq 1 \qquad (2)$$

Asking for small $w\cdot w$ is like "weight decay" in Neural Nets and like Ridge Regression parameters in Linear regression and like the use of Priors in Bayesian Regression-all designed to smooth the function and reduce overfitting [14]. In SVM, slack variable

```
<Original Q> What is the population of the Bahamas?
<Question Analysis>
   <AF> Factoid </AF>
   <AT> QT_NUMBER </AT>
   <QT> target: Bahamas / focus: population </QT>
   <AD> Wikipedia > news > blogs > others </AD>
<Answer rank =1>
   <Ans> 370267</Ans>
   <Ans_sent> The population of The Bahamas on
   January 1st 2010 is approximately 370267.
<Ans_sent>
    <Ans_source> Wiki-QA </Ans_source>
    <org_weight> 0.3 </ org_weight >
</Answer>
<Answer rank =2>
   <Ans> 294,982 </Ans>
   <Ans_sent> The population in the Bahamas is
        currently about 294,982 persons. <Ans_sent>
    <Ans_source> News-QA </Ans_source>
    <org_weight> 0.5 </ org_weight >
</Answer>
…
```

Figure 4. An Example of Evaluation Set.

ξ_i measures by how much example $(\vec{x}_i, y_i)$ fails to archive a target margin of δ. We adjust the weights which are located out of margin area as slacks according to traits of the question analysis result (AF, AT, QT, and AD in Equation (1)) and answer evidences such as snippet, matched keywords, or position in the answer document. After adjusting, the slack weights can be moved the safe boundary, as illustrated in Figure 3.

For training, we built <question, answer> set of various sorts in terms of questions and answer sources. Figure 4 is an example of a <question, answer> pair. We can notice that the weight of wiki-QA should be boosted from 0.3 to 0.6 (or any higher scores than other lower ranked answers), even though the original local confidence weight of the answer from the News-QA (0.5) is higher than wiki-QA's (0.3). On the other hand, some cases should be decreased.

By training, we learned the confidence weight distribution and slack boundaries ξ_i and set boosting ratio Δ_i in Equation (3):

$$y_i(\vec{w} \cdot \vec{x}_i \Delta_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \qquad (3)$$

According to question types, the boosting ratios are different and they are updated whenever training questions are leaned. We also determined the threshold values for each of sub-QA to avoid superfluous calls for other QA modules.

At the result, when a new user question which is similar with training questions is entered, we can predict which QA modules are likely to find answers and how much we should increase or decrease the confidence weights from multiple sub-QAs.

V. Evaluation and Analysis

A. Test Collection and Measure

Among 7,070 questions collected from real users, 577 questions are selected with considering various question/answer types. We used 260 pairs of training and 317 pairs of testing, which is part of the entire set of 577 <question, answer> pairs.

For effectiveness comparisons, we employ precision, recall, and F-score, sometimes with the mean reciprocal rank (MRR) [15] and the well-known "top-n" measure that considers whether a correct nugget is found in the top n-th answers. Because of the large number of comparisons to be made for different cases, we use F-scores for summary tables.

B. Experimental Background

As described in Section 2, our distributed QA model contains four different genre-specific QAs (News-QA, Blog-QA, Wiki-QA, and Web-QA) and four domain-specific QAs (Game-QA, Travel-QA, Local-QA, and Life-QA), which are covered most frequently asked questions.

Before the main experiment, we have to announce that characteristics of our distributed QA. Table 2 summarizes performance of individual Genre-QAs and Figure 5 shows weight distribution of four Genre-QAs.

As shown in Table 2, the News-QA answered for 244 questions among 317 of entire set. Out of 244 candidate answers, 139 answers are correct so the News-QA precision is 0.570. The best performed Genre-QA is Web-QA since Yahoo web search is covered all kind of web documents, whereas other genre-QAs are focused on specific genres.

Figure 5 depicts original confidence weights of the first answers (Top-1) merged from each four genre-QAs for 317 testing questions. If a sub-QA cannot find appropriate answers for a given question, then the weight regards as 0. In general, weights from the Blog-QA (marked with "■") are higher than others, while the Web-QA weights (marked with "x") are lower. The scores from the Wiki-QA (marked with "▲") are between 0.6 and 0.98, but the Blog-QA and the News-QA (marked with "♦") show inconsistent values from 0 to even 8. We had pruned weights over 2 to avoid overffiting to extremely higher values.

To prove reliability of our proposed QA model and re-ranking algorithm, we measured the lower and upper boundaries.

The accuracy of lower bound can be estimated when selecting the top-ranked answer among the merged candidates from multiple sub-QAs without any adjustment. As shown in the right side of Table 2, the accuracy of simply merging and selecting the top-ranked answer shows just 0.568, even though micro-average precision of all genre-QAs is 0.722. That is poor than the case of performing only one sub-QAs. Because the News-QA and the Blog-QA usually are usually higher than others as shown in Figure 5, the final selected weight depends on them. This result supports that re-ranking is very important, which is the main focus in this paper.

Under assumption that the correct answer for a given question might be exist at least among all local optimal candidate answers generated from multiple sub-QAs, we evaluated all Top-3 ranked answers for each sub-QA. The number of candidates in answer pool can be from 0 to 24 at most (3 answer x 8 sub-QAs). We regarded this result as the upper boundary. Out of 317 questions, user can get correct answers for 273 questions. The fact that the recall is quit high (0.861) prove our assumption is reasonable. Our upper boundary precision is 0.907 and MRR is 0.849.

C. Analysis of Experimental Results

This paper had pursued adjusting diverse confidence weights from multiple sub-QAs as shown in Figure 5. For a detail example, Figure 6 shows the original weights from the Blog-QA. They are divided into correct and incorrect answers which are indicated with "●" and "x", respectively. Because some fake answers have very high scores as shown in Figure 6, they lead to ignore the other sub-QAs results and ultimately cause false alarm.

After the training process which is described in Section 4, we learned which sub-QAs are more relevant and how to boost their weights depending on question types. As a result, we observed that the Blog-QA is suitable to answer for factoid questions about game/computer or personal life domains. In contrast, answers for questions looking for interesting information about a particular person or thing such as "*Who is Vlad the Impaler?*" or "*What is a golden parachute?*" are founded in the Wiki-QA. Figure 7 presents the adjustment result for the Blog-QA. While some wrong answers are located in higher position, we can cut-off candidates less than 0.2.

Table 3 and 4 summarize the accuracy of our distributed QA adopted the proposed re-ranking algorithm. As mentioned in Section 5.1, multiple sub-QAs answered for 301 questions and could not find any candidate for 16 questions, out of 317 questions. We merged the Top-3 candidates for each sub-QA by the B2 broker, and adjusted their confidence weight by the trained model.

TABLE II. PERFORMANCE OF INDIVIDUAL GENRE-QAS AND BOUNDARIES

	Individual Genre-QAs (Top-3)				*Lower bound*	*Upper bound*
	News-QA	Blog-QA	Wiki-QA	Web-QA	*Selecting the top-ranked*	*All Top-3 ranked answers*
#of answered Q.	244	272	135	228	301	301
# of missed Q	73	45	182	89	16	16
# of corrected Q.	139	204	103	184	130	273
Precision	0.570	0.750	0.763	0.807	0.568	0.907
Recall	0.541	0.794	0.401	0.716	0.539	0.861
F-score	0.555	0.771	0.526	0.759	0.553	0.883
MRR	0.488	0.645	0.740	0.711	0.568	0.841
Micro-Average	Precision = 0.722 / Recall = 0.613					

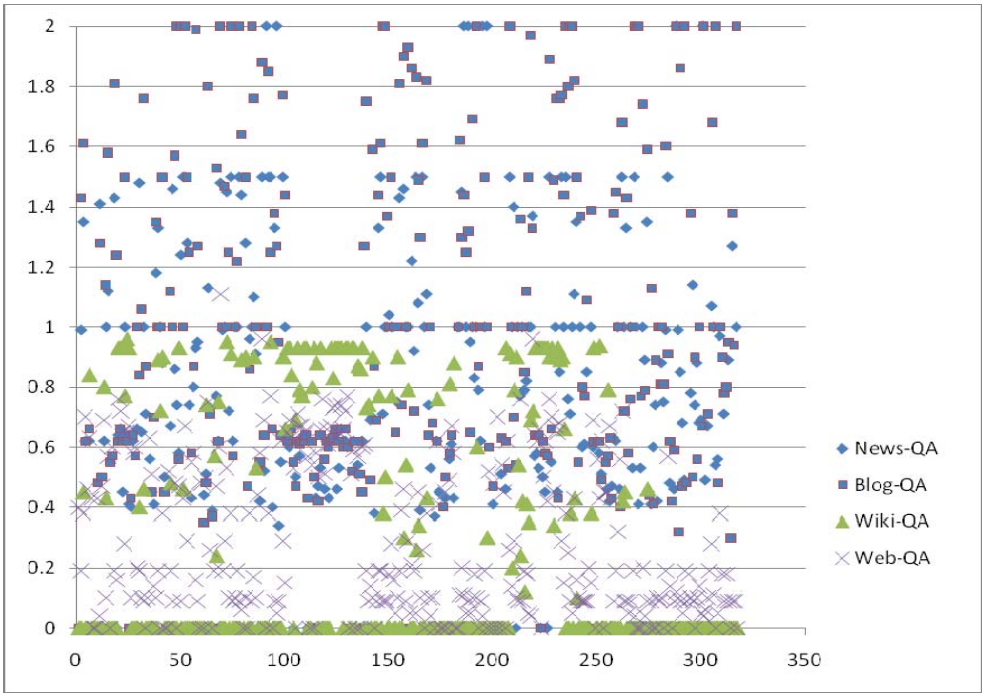

Figure 5. Weight Distribution of four Genre-QAs

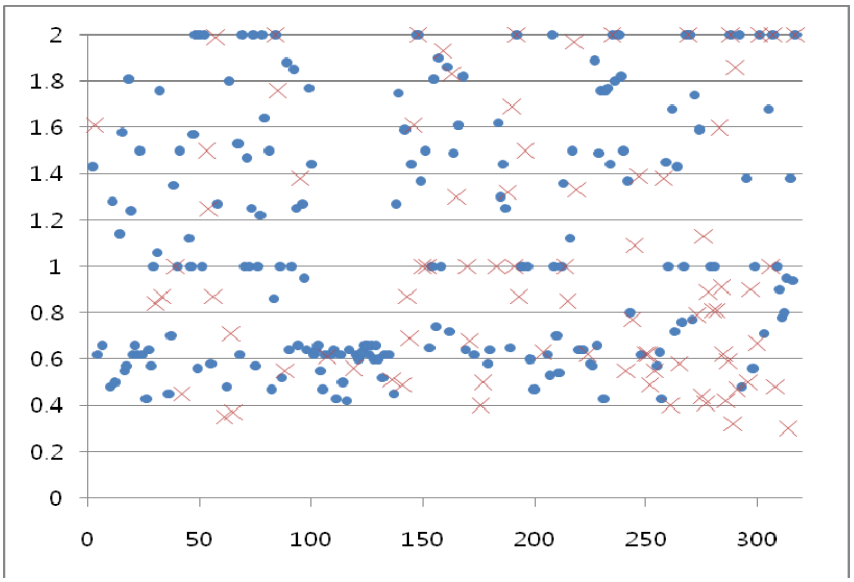

Figure 6. The original weights from the Blog-QAs

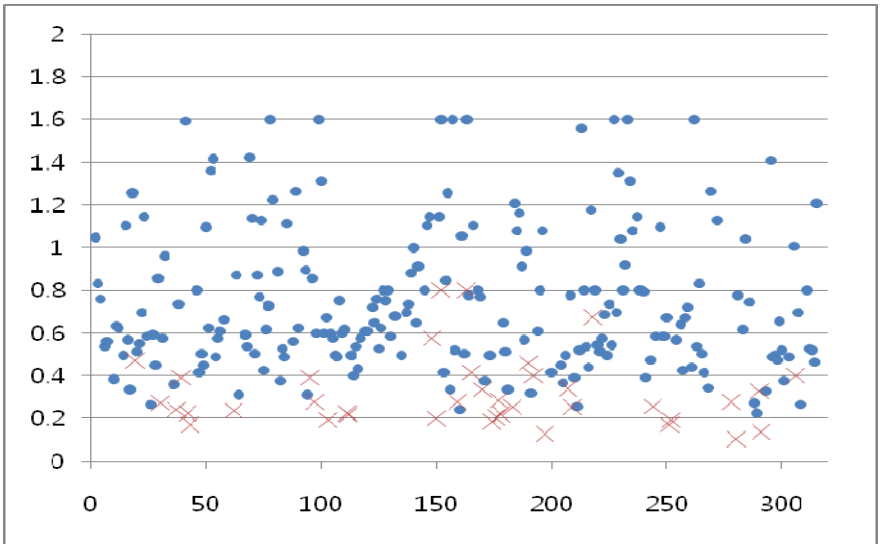

Figure 7. Adjusted weights of the Blog-QAs

TABLE II. EVALUATION RESULT OF THE TOP-1

	Lower B.	**Top-1**
# of corrected	130	202 *(+72)*
Precision	0.568	0.671 *(+18.13%)*
Recall	0.539	0.637 *(18.18%)*
F-score	0.553	0.654 *(18.26%)*

TABLE II. EVALUATION RESULT OF THE TOP-3

	Top-3	Upper B.
# of corrected	248 *(-25)*	273
Precision	0.824 *(-9.15%)*	0.907
Recall	0.782 *(-9.18%)*	0.861
F-score	0.803 *(-9.06%)*	0.883
MRR	0.801 *(-4.76%)*	0.841

To compare with the lower boundary, we evaluated only the top-ranked answers for 301 questions. As shown as Table 3, we obtained an increased accuracy by about 18% of precision, recall and F-score. 72 (130 to 202) answers are adjusted by our re-ranking algorithm. In Top-1 evaluation, MRR is same with precision.

Compensating the factor that the upper boundary considered all candidates in the Top-3 for each sub-QA, we also evaluated the Top-3 of re-ranked answers, but not all answer pool. As in Table 4, we missed 25 questions' answers while we handled just three candidates on top. We got the loss about 9% (-9.06%) of F-score and 5% (-4.76%) of MRR. The fact that the gap between MRRs (0.801 vs. 0.841) is smaller than other measures indicates the answers of our method are located in higher position. That is by our re-ranking algorithm; we can not only suppress erroneous answers but also save time.

VI. CONCLUSION

The main motivation behind this work was to devise a way to combine multiple QA modules to answer various user questions. To this end, we analyzed real-life questions for their characteristics and classified them into different domains and genres. In the proposed distributed QA framework, 8 specialized sub-QAs are combined and an advanced re-ranking algorithm are adopted to adjust confidence weights calculated by own ranking methods in sub-QAs.

We ran a series of experiments to see the effects of the proposed re-ranking algorithm against two different cases: (1) the lower boundary when considering only the first answers from sub-QAs, (2) the upper boundary when evaluating all local optimal candidate answers. The result based on 317 questions show that our re-ranking method outperforms the lower boundary by about 18%. Compared with the upper case, the loss is narrowly about 5% in MRR.

Based on the result of question analysis, The B2 in Answer Manager determined invocation of sub-QAs. In particular, the expected answer type and answer domain analysis for a question presents a critical problem because it influences the re-ranking process. We plan to improve upon the answer type classification and domain expectation modules by expanding training corpus toward including various question types.

ACKNOWLEDGMENT

This work was supported in part by the Korea Ministry of Knowledge Economy (MKE) under Grant No. 2011-SW-10039158.

REFERENCES

[1] M. Voorhees, "The TREC-8 Question Answering Track Report." Proc. of the 8th Text REtrieval Conference (TREC-8), 1999, pp. 77-82.

[2] Harabagiu, D. Moldovan, C. Clack, et. al., "Answer mining by combining extraction techniques with abductive reasoning" Proc. of the 12th Text REtreival Conference(TREC-12) , 2003, pp. 375-382.

[3] J. Chu-Carroll et al., "A Multi-strategy and Multi-source Approach to Question Answering," Proc. of the 11th Text REtrieval Conference (TREC-11), 2002, pp. 281-288
[4] B. Katz et al., "Answering Multiple Questions on a Topic from Heterogeneous Resources," Proc. of the 13th 11th Text REtrieval Conference (TREC 2004).
[5] H-J. Oh, S. H. Myaeng, and M. G. Jang, "Enhancing Performance with a Learnable Strategy for Multiple Question Answering Modules." ETRI Journal, vol.31, no.4. 2009, pp. 419-428
[6] J. Chu-Carroll, J. Prager, C. Welty et al, "A Multi-Strategy and Multi-Source Approach to Question Answering," Proc. of the 11th Text REtrieval Conference (TREC-11), 2002, pp.281-288.
[7] A. Hickl, J. Williams, J. Bensley et al, "Question Answering with LCC's CHAUCER at TREC 2006," Proc. of the 15th Text REtrieval Conference (TREC 2006).
[8] V. Lopez, M. Sabou, V. Uren, and E. Motta, "Cross-Ontology Question Answering on the Semantic Web –an initial evaluation", Proc. of the Knowledge Capture Conference, 2009
[9] V. Lopez, A. Nikolov, M. Fernandez, et al., "Merging and Ranking Answers in the Semantic Web: The Wisdom of Crowds", Proc. of the ASWC 2009, LNCS 5926, 2009, pp. 135–152.
[10] M. Wu and A. Marian, "A Framework Corroboration Answers from Multiple Web Sources". Information Systems, 2010, doi:10.1016/j.is.2010.08.008, online press.
[11] Dhruba Borthaku. "The Hadoop Distributed File System: Architecture and Design", http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf , 2007.
[12] C.K. Lee and M.G. Jang, "A Prior Model of Structural SVMs for Domain Adaptation," ETRI Journal, vol. 33, no. 5, 2011, pp. 712-719
[13] H-J, C.K. Lee and C-H Lee, "Analysis of the Empirical Effects of Contextual Matching Advertising for Online News", ETRI Journal, vol. 34, no. 2, April 2012 (to be appeared)
[14] A. W. Moore, tutorial of "Support Vector Machine", http://www.cs.cmu.edu/~awm/tutorials
[15] E. M. Voorhees and M. T. Dawn. "Building a question answering test collection", Proc. of the 23rd Annual International ACM SIGIR, 2000, pp. 200-207

Hyo-Jung Oh received the B.S. and the M.S. degrees in computer science from Chungnam National University, Daejeon, South Korea, in 1998 and 2000, respectively. She received the Ph.D. degree in computer engineering from KAIST, Daejeon, South Korea, in 2008. Currently she is a senior researcher in Electronics and Telecommunications Research Institute (ETRI), Deajeon, South Korea. Her research interests include machine learning, question answering, and listening platform for business intelligence.

Pum-Mo Ryu received the B.S. degree in computer engineering from Kyungpook National University, Daegu, South Korea in 1995 and the M.S. degree in computer engineering from POSTECH, Pohang, South Korea, in 1997. He received the Ph.D. degree in computer science from KAIST, Daejeon, South Korea, in 2009. Currently he is a senior researcher in Electronics and Telecommunications Research Institute (ETRI), Deajeon, South Korea. His research interests include natural language processing, text mining, knowledge engineering and question answering.

Hyunki Kim received the B.S. and the M.S. degrees in computer science from Chunbuk National University, Daejeon, South Korea, in 1994 and 1996, respectively. He received the Ph.D. degree in computer science from University of Florida, Gainesville, USA, in 2005. Currently, He is a principal researcher in Electronics and Telecommunications Research Institute (ETRI), Deajeon, South Korea. His research interests include natural language processing, machine learning, question answering, and listening platform for business intelligence.