Information Processing and Management 50 (2014) 683–692

Contents lists available at ScienceDirect

Information Processing and Management

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

# Open domain question answering using Wikipedia-based knowledge model

Pum-Mo Ryu [*], Myung-Gil Jang, Hyun-Ki Kim

*Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon 305-700, Republic of Korea*

ARTICLE INFO

*Article history:*
Received 8 September 2012
Received in revised form 25 April 2014
Accepted 28 April 2014
Available online 6 June 2014

*Keywords:*
Question-answering
Wikipedia
Semi-structured knowledge

ABSTRACT

This paper describes the use of Wikipedia as a rich knowledge source for a question answering (QA) system. We suggest multiple answer matching modules based on different types of semi-structured knowledge sources of Wikipedia, including article content, infoboxes, article structure, category structure, and definitions. These semi-structured knowledge sources each have their unique strengths in finding answers for specific question types, such as infoboxes for factoid questions, category structure for list questions, and definitions for descriptive questions. The answers extracted from multiple modules are merged using an answer merging strategy that reflects the specialized nature of the answer matching modules. Through an experiment, our system showed promising results, with a precision of 87.1%, a recall of 52.7%, and an F-measure of 65.6%, all of which are much higher than the results of a simple text analysis based system.

## 1. Introduction

The goal of a question answering (QA) system is to directly return answers, rather than documents containing answers, in response to a natural language question. The answers can be fact-based short answers, lists of instances, or descriptions about a particular topic. Many of the initial efforts in QA research, ignited by the QA track in TREC (Dang, Kelly & Lin, 2007; Voorhees, 2004), have focused on mining unstructured texts such as news sites and blogs. However, these systems show a relatively low performance, with at most 71% accuracy for factoid questions, 48% F-score for list questions, and 33% F-score for descriptive questions. On the other hand, specialized QA systems have relied on well-structured knowledge bases in specific domains (Demner-Fushman & Lin, 2007; Frank et al., 2007). Although these works achieved high accuracy, building large-scale, well-structured knowledge bases for a general domain QA is a very expensive task.

Wikipedia is a semi-structured and wide covering, rapidly growing knowledge source that has been built through a collaborative effort of volunteers. Wikipedia has become a stable and sufficiently large knowledge source for many knowledge-based engineering works (Bizer et al., 2009; Hoffart et al., 2013; Nastase & Strube, 2008; Suchanek et al., 2007). Furthermore, Wikipedia was applied to QA systems as a knowledge base (Ahn et al., 2004; Buscaldi & Rosso, 2006; Simmons, 2012). However, these systems utilized only parts of Wikipedia information. Thus, we developed an open-domain QA system that fully utilizes semi-structured Wikipedia knowledge model. The knowledge model can serve as certified information sources and enable a QA system to generate correct answers in a general domain. We exploit the category structure, article structure, infoboxes, definitions, redirection, and article contents of Wikipedia as knowledge sources for

[*] Corresponding author. Tel.: +82 42 860 5327; fax: +82 42 860 4889.
*E-mail addresses:* pmryu@etri.re.kr (P.-M. Ryu), mgjang@etri.re.kr (M.-G. Jang), hkk@etri.re.kr (H.-K. Kim).

http://dx.doi.org/10.1016/j.ipm.2014.04.007
0306-4573/© 2014 Elsevier Ltd. All rights reserved.

a QA system. We assume each knowledge source has its own strengths for answering different types of answer formats such as factoid, list, and description. For example, an infobox is effective in answering factoid questions, and the category structure is effective in answering lists of questions.

Well-organized knowledge bases do not guarantee high performance if the questions are in natural language instead of formal query. Mapping linguistic expression in questions to knowledge representation in knowledge-base is another hard task if we build full-featured knowledge from Wikipedia like YAGO (Hoffart et al., 2013). F-scores of QA systems in QALD task are about 50%, which are much lower than expected result (http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/). The task covers extracting answers from well-organized knowledge-bases for given natural language questions. So, our system is based on a conventional QA system, which consists of a question analysis module, document retrieval module, and answer matching module. To this end, the different types of knowledge sources are converted into text documents for the document retrieval module. Instead, specialized answer matching modules are developed for the knowledge types.

Section 2 describes the question analysis, while Section 3 describes our Wikipedia QA system. Section 4 describes the experiment used, and concluding remarks are given in Section 5.

## 2. Question analysis

To make full use of knowledge sources of Wikipedia for many types of questions, it is critical to analyze user questions in terms of the nature of the answers being sought. The availability of a question categorization scheme will help not only in analyzing an incoming user question but also in identifying QA capabilities and techniques to be developed in the future (Oh et al., 2011). To this end, we collected 600 questions from a commercial Korean website, Naver™ Manual QA Service (http://kin.naver.com). The questions were analyzed to characterize the types of questions and answers. The results of our analysis are shown in Table 1 for surface-level question type classes determined based on interrogative pronouns. These results are further divided into answer formats corresponding to the classes used in TREC (Voorhees, 2004). While TREC used a "definitional" type, we generalized it as a "descriptive" type, which includes "definitional," "reasons," and "methods" types (Oh et al., 2009).

A user question in natural language form is analyzed using multiple linguistic analysis techniques including POS tagging, chunking, and named entity tagging (Lee et al., 2006; Lee & Jang, 2011). The analyzed result of a question has three components including answer format (AF), answer theme (AT) and question target (QT). The AF has three possible values: factoid, list, and descriptive. They can be distinguished based on the surface-level description of questions. For example, "Where is the Nile River located" looks for a single factoid answer, whereas "Who are American politicians who have emigrated from Austria" requires a list of answers. A descriptive question needs an answer that contains definitional, causal, or method information about a key term, as in "What is X," "What is the cause of X," or "What is the method for X?". An AT is the class of the object or description sought by the question, such as PERSON, LOCATION, and DATE for a factoid; list answer format; and DEFINITION, REASON, and METHOD for a descriptive answer format. We used a total of 147 answer themes, which are organized into a hierarchical structure (Lee et al., 2006). A QT consists of two parts: object and property. The former is the main object or event that the question is about, whereas the latter is the property of interest that a question attempts to get at regarding the object. In "Where is the Nile River located," for example, the object is "Nile River" and the property is "be located." The key elements in detecting the question target are the predicate-argument structure or noun phrase structure in the dependency structure of the given question. When the property is not clear, it can remain empty.

Given a question $q$, we want to find a question analysis result $r = (af, at, qt)$ which most likely explains what the question means as follows;

$$\hat{r} \leftarrow \arg\max_{r} S_Q(r|q)$$
$$S_Q(r|q) = S_{AF}(af|q) \cdot S_{AT}(at|q) \cdot S_{QT}(qt|q)$$

**Table 1**
Distribution of surface-level question types.

| Surface level | Answer format | # Questions | Example |
|---|---|---|---|
| Who | Factoid | 44 | Who is the president of Korea? |
| | List | 50 | Who are American politicians emigrated from Austria? |
| | Descriptive (definition) | 72 | Who is Barack Obama? |
| What/Which | Factoid | 87 | What is the name of the oldest aircraft? |
| | List | 71 | Who are an American politicians emigrated from Austria? |
| | Descriptive (definition) | 84 | What is the movie based on Facebook? |
| Where | Factoid | 62 | Where is the Nile river located? |
| When | Factoid | 28 | When did World War I end? |
| Why | Descriptive (reasons) | 52 | Why do typhoons occur? |
| How | Descriptive (methods) | 50 | How to fix a flat tire? |
| Total | | 600 | |

where $S_Q(\,)$ is a score for question analysis, and $S_{AF}(\,)$, $S_{AT}(\,)$, $S_{QT}(\,)$ are scores for analyzing AF, AT and QT, the scores are normalized between 0 and 1.

## 3. Wikipedia QA system

Ideally, questions should be answered through a direct comparison to a well-structured knowledge base. However, the cost of building and searching a well-structured large scale knowledge base is too expensive, so we rely on a conventional QA system architecture consisting of a question analyzing module, document retrieval module, and answer matching module. The system select the best answer $a$ for given question $q$ that maximizes the multiplication of question analysis score $S_Q(r|q)$, document retrieval score $S_D(d|r)$ and answer matching score $S_{A(M)}(a|q, r, d)$ by module $M$ as follows:

$$\hat{a} \leftarrow \arg\max_a S(a|q)$$

$$S(a|q) = S_Q(r|q) \cdot S_D(d|r) \cdot S_{A(M)}(a|q, r, d)$$

where $r$, $a$, $d$ are question analysis result, answer candidate and retrieved document, respectively. The scores are normalized between 0 and 1. An additional answer merging module combines and ranks the answers generated from the answer matching modules (Fig. 1). To this end, article section titles, infoboxes, and article categories are converted into text documents. Wikipedia definition and redirection databases are managed for accurate and wide coverage answers.

### 3.1. Article content module

We extract answers from article content using a traditional answering method. Most state-of-the-art QA systems implement a technique for extracting paragraph-sized passages of text from a large corpus (Khalid & Verberne, 2008). Therefore, we split articles into small passages based on their article structure. Entities that match to the answer theme are selected as answer candidates from retrieved documents for factoid and list questions, and all sentences in the retrieved documents are answer candidates for descriptive questions. Answer matching score is measured based on the distance between question words and answer candidate $a$ in document $d$ for factoid, list questions. For descriptive questions, the score is measured based on the content similarity and the pattern similarity measures. The content similarity is a similarity between words in the question and words in the answer candidate. The pattern similarity is the similarity between sentence patterns for descriptive answers and answer candidates. We defined lexico-syntactic patterns in regular expression format that embed Korean sentence styles for definition, reason, and method descriptions (Table 2).

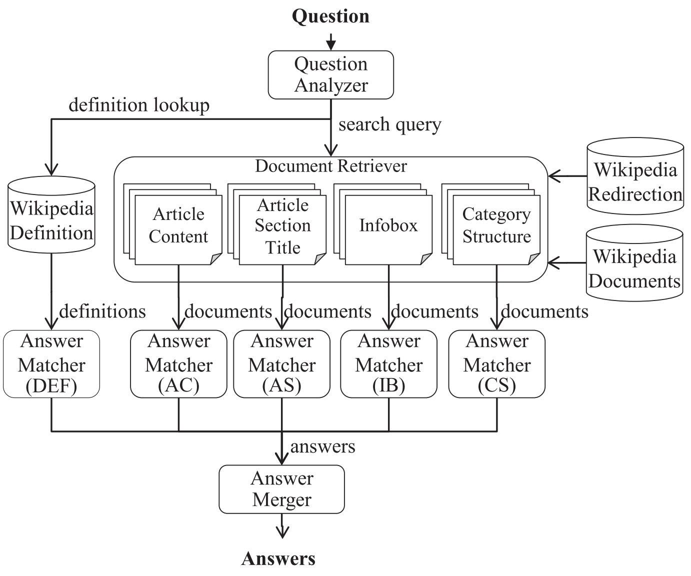


**Fig. 1.** System overview.

**Table 2**
Examples of Korean lexico-syntactic patterns in regular expression format for descriptive questions. QBJ is object in question target.

| Type | Lexico-syntactic pattern | English translation |
|---|---|---|
| DEFINITION | OBJ (은\|는). + (의미\|뜻\|말) ? (한\|합니)다\.? | OBJ means that ~ |
| REASON | OBJ + 의 ([^ ] + )?(원인은\|이유는) | The reason of OBJ is that ~ |
| METHOD | .+(하면\|하시면) OBJ . + (할\|하실) 수 있(습니다\|어요\|다)\.? | You can do OBJ by doing ~ |

$$S_{A(AC)}(a|q,r,d) = \begin{cases} S_{dist}(q,a,d) & \textit{for factoid, list questions} \\ S_{csim}(q,a)\cdot S_{psim}(q,a) & \textit{for descriptive questions} \end{cases}$$

### 3.2. Article structure module

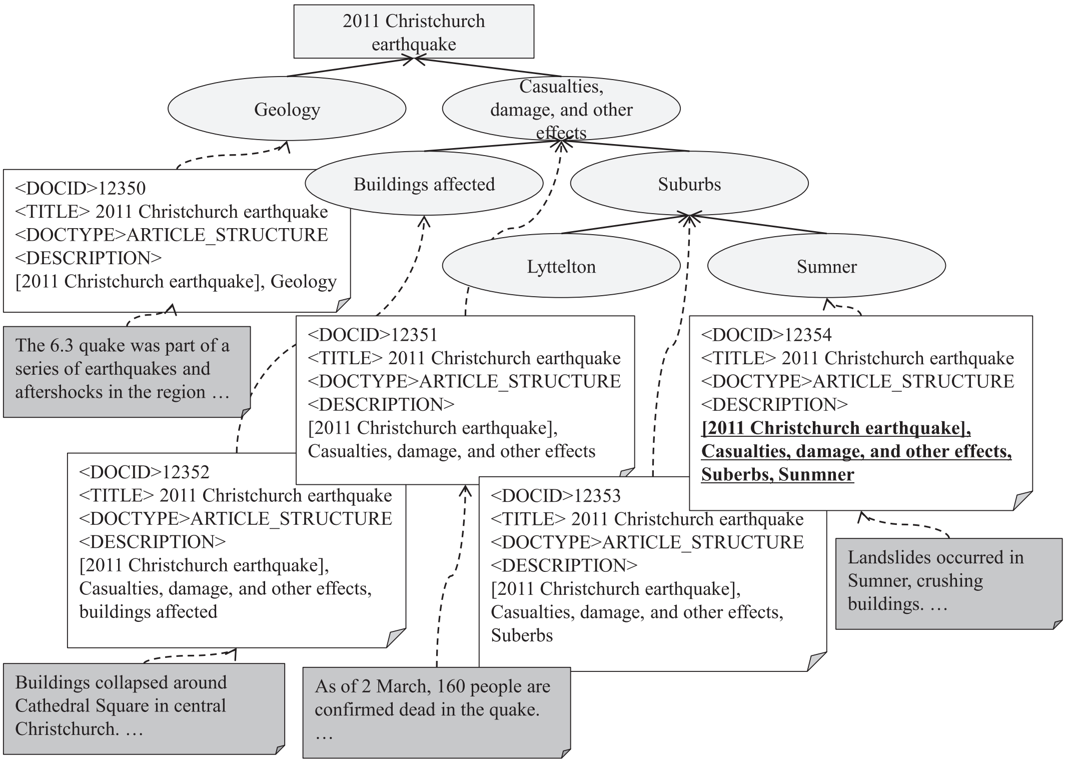
Because the sections are divided based on important properties or issues that many users are interested in, the article's section structure is a valuable knowledge source in a QA system. Sometimes, it is very hard to determine proper answer themes for certain questions. For example, the answer theme for the question, "What about damage in Sumner area by 2011 Christchurch earthquake," is not clear because our answer theme structure does not have "damage" type. The answer theme for this question is a description rather than a pre-defined type. We can find answers for such questions using article title, section titles and QT of questions. To this end, an article is divided into multiple sections where a document pair <$d_t$, $d_c$> is assigned to each section, where $d_t$ = <$t_A$, $t_{U1},\ldots,t_{UN}$, $t_S$> is a title document that includes article title $t_A$, section title $t_S$, and its upper level section titles $t_{U1},\ldots,t_{UN}$, and $d_c$ is a section content that can be an answer candidate of a matched question. Thus, the object and property names of the QT are used as a query for the document retrieval module. The answer matching module finds the object name at the beginning of the retrieved document and matches the property in the remaining parts of the document as follows;

$$S_{A(AS)}(a|q,r,d) = \begin{cases} 1 & \textit{if question target matches article title and property matches section title} \\ 0 & \textit{otherwise} \end{cases}$$

Fig. 2 shows an article structure for "*2011 Christchurch earthquake*" and its generated documents. The top rectangle is the article title, and the ellipses are subsection titles. The white and gray rectangular papers are section title documents and section content documents, respectively. For above question, a section title document with an id of 12354 is retrieved and its corresponding section content document selected as an answer candidate.

### 3.3. Category structure module

Some Wikipedia knowledge is encoded in the network structure of articles and categories. In particular, the categories are organized in a taxonomy-like structure (Ponzetto & Strube, 2007). Categories for an article are generalized descriptions of the article. Thus, we can find article names via the descriptions encoded in their categories. For example, the article "Arnold


2011 Christchurch earthquake
Geology
Casualties, damage, and other effects
Buildings affected
Suburbs
Lyttelton
Sumner

<DOCID>12350
<TITLE> 2011 Christchurch earthquake
<DOCTYPE>ARTICLE_STRUCTURE
<DESCRIPTION>
[2011 Christchurch earthquake], Geology

The 6.3 quake was part of a series of earthquakes and aftershocks in the region …

<DOCID>12351
<TITLE> 2011 Christchurch earthquake
<DOCTYPE>ARTICLE_STRUCTURE
<DESCRIPTION>
[2011 Christchurch earthquake], Casualties, damage, and other effects

<DOCID>12352
<TITLE> 2011 Christchurch earthquake
<DOCTYPE>ARTICLE_STRUCTURE
<DESCRIPTION>
[2011 Christchurch earthquake], Casualties, damage, and other effects, buildings affected

<DOCID>12353
<TITLE> 2011 Christchurch earthquake
<DOCTYPE>ARTICLE_STRUCTURE
<DESCRIPTION>
[2011 Christchurch earthquake], Casualties, damage, and other effects, Suberbs

<DOCID>12354
<TITLE> 2011 Christchurch earthquake
<DOCTYPE>ARTICLE_STRUCTURE
<DESCRIPTION>
**[2011 Christchurch earthquake], Casualties, damage, and other effects, Suberbs, Sunmner**

Landslides occurred in Sumner, crushing buildings. …

Buildings collapsed around Cathedral Square in central Christchurch. …

As of 2 March, 160 people are confirmed dead in the quake. …

**Fig. 2.** Section structure for the article "2011 Christchurch earthquake" and generated documents.

Schwarzenegger" is under categories such as "Austrian immigrants to the United States" and "American actor-politicians," as shown in Fig. 3. In this case, we can answer the question "Who are American politicians emigrated from Austria?" The category structure can find answers for list questions. Unfortunately, Wikipedia categories do not form taxonomy with a fully-edged subsumption hierarchy, but only a thematically organized thesaurus. Paths through a non-isa link may connect to noisy category names. We applied the method of Ponzetto and Strube (2007) to filter non-isa relations from the upper categories of articles in Korean Wikipedia. Because the category naming conventions in English and Korean Wikipedia differ, we devised a name analysis rules for Korean. Plural nouns in English category names are strong evidences to determine types of articles. We can infer "Arnold Schwarzenegger" is a kind of "Person" from the word "immigrants" in category names. But it is not obligatory to use plural nouns in Korean category names. So we predefined clue words for each entity type and filtered the categories of which entity types of head words are differ from other majority category names (Table 3). For example, "민속(*folklore*)" is not a clue word for "Person" unlike other category names, so "American folklore" has non-isa relation to "Arnold Schwarzenegger" in Fig. 3.

For each article, after isa upper categories are determined, a category structure document is generated. The document contains a one-line description where article name and its upper category names are listed sequentially: <*article name*, (*upper category names*,)$^{+}$>. Because more generic category names are less informative, we set the maximum depth of the upper category path as three. Fig. 4 shows an example document for the category structure in Fig. 3. After a category structure document is retrieved through the document retrieval module, the article name, located at the beginning of the document, is selected as an answer candidate. Answer matching score for category structure module is as follows;

$$S_{A(CS)}(a|q,r,d) = \begin{cases} 1 & \textit{if all words in q are found in the retrieved document} \\ 0 & \textit{otherwise} \end{cases}$$

For the above question, the category document of Fig. 4 is retrieved as a relevant document, and "Arnold Schwarzenegger" is selected as an answer.

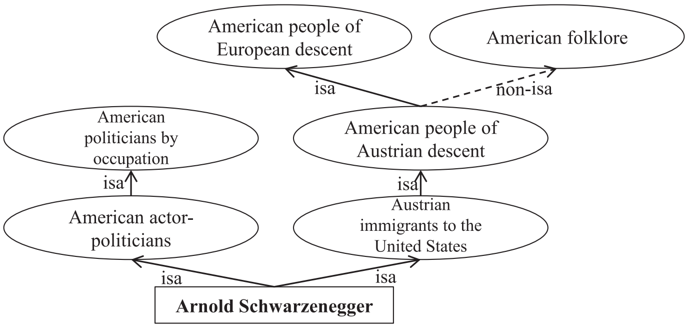
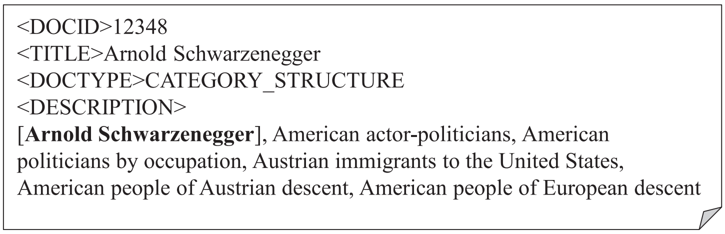

American people of European descent
American folklore
isa
non-isa
American politicians by occupation
American people of Austrian descent
isa
isa
American actor-politicians
Austrian immigrants to the United States
isa
isa
Arnold Schwarzenegger

**Fig. 3.** Upper categories of "Arnold Schwarzenegger".

**Table 3**
Examples of Korean clue words to identify types of category names and their corresponding English words.

| Entity type | Clue words (Korean) | Clue words (English) |
|---|---|---|
| Person | 동문, 태어남, 배우, 사람, 출신, 정치인 | Alumni, births, actors, people, immigrants, politicians |
| Company | 기업, 그룹 | Companies, groups |
| Event | 행사, 전쟁, 지진 | Events, wars, earthquakes |



<DOCID>12348
<TITLE>Arnold Schwarzenegger
<DOCTYPE>CATEGORY_STRUCTURE
<DESCRIPTION>
[**Arnold Schwarzenegger**], American actor-politicians, American politicians by occupation, Austrian immigrants to the United States, American people of Austrian descent, American people of European descent

**Fig. 4.** Document for the category structure of Fig. 3.

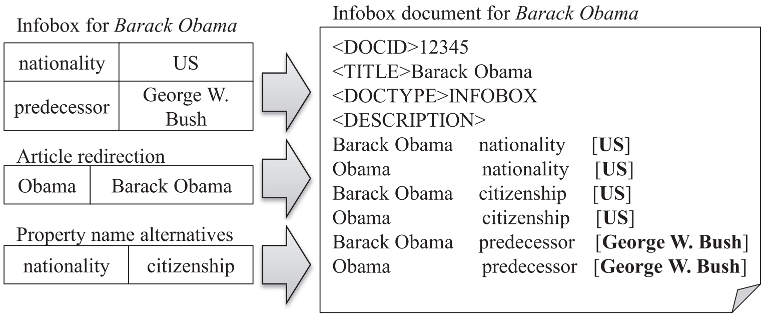

Infobox for *Barack Obama*
nationality
US
predecessor
George W. Bush
Article redirection
Obama
Barack Obama
Property name alternatives
nationality
citizenship
Infobox document for *Barack Obama*
<DOCID>12345
<TITLE>Barack Obama
<DOCTYPE>INFOBOX
<DESCRIPTION>
Barack Obama nationality [**US**]
Obama nationality [**US**]
Barack Obama citizenship [**US**]
Obama citizenship [**US**]
Barack Obama predecessor [**George W. Bush**]
Obama predecessor [**George W. Bush**]

**Fig. 5.** Infobox document for "Brack Obama".

### 3.4. Infobox module

An Infobox displays the most relevant facts of an article as a table of property-value pairs. The basic form of the infobox document is a list of triples: <*article title*, *property name*, *property value*>. All possible alternations of triples are expressed in the document using an article redirection and property name alternatives. Article redirection has different notations for an article assigned by authors. For example, the article "*Obama*" is redirected to "*Barack Obama*." A property name can be expressed in different lexicals based on the author's selection. For example, "*nationality*" for the "*president*" infobox template is semantically equivalent to "*citizenship*" for the "*writer*" infobox template. We manually built a property name alternative database for persons, organizations, locations, entertainment, and product templates. Fig. 5 shows a part of an infobox document for "*Barack Obama*."

For a given question, the document retrieval system searches relevant infobox documents, and the answer matcher compares questions and each line in the retrieved documents. Property values are extracted as answer candidates if an article title and property name are matched to the question. For the question, "*Who is Obama's predecessor?*" after the document in Fig. 5 is selected, the answer matcher scans each line in the description and finds "*George W. Bush*" as an answer candidate. The answer matching score of answer candidate *a* by infobox module is as follows:

$$S_{A(IB)}(a|q,r,d) = \begin{cases} 1 & \textit{if article title and property name of a given line in d matches to the object and property of QT in r} \\ 0 & \textit{otherwise} \end{cases}$$

### 3.5. Definition module

We manage Wikipedia article titles and their first paragraph as a definition knowledge base. If the answer theme of a question is DEFINITION, our QA module looks up the definition knowledge base using the object of the QT. For questions such as "What is a tsunami" or "Who is Obama," the system searches Wikipedia article titles that match "tsunami" or "Obama." If one or more articles are found, the system suggests contents of the articles as answer candidates.

### 3.6. Answer merging module

Our QA system makes use of multiple QA modules employing different answer finding methods. A strategy that determines the sequence of module invocations to be invoked when finding an answer is selected based on several factors such as the expected AF, AT and QT of the question as in Table 4. Given two modules, QA1 → QA2 shows that QA1 and QA2 are invoked in sequence, whereas QA1 + QA2 indicates that they are to be processed in parallel. The algorithm for the sequence and parallel invocations are described in Tables 5 and 6 respectively. The composition of module invocations are determined based on the module tests for each question type (Table 9). For factoid question, precisions of IB and AS are higher than those of other modules. So, the strategy invokes IB and AS modules first, and the scores of answer candidates generated from the

**Table 4**
QA strategy.

| Answer format | Answer theme | Question target | Strategy |
|---|---|---|---|
| Factoid | 146 Theme | O-P | (IB + AS) → (CS + AC) |
| List | | O-P | CS → (AS + AC) |
| Descriptive | Definition | O-X | DEF → AC |
| | Reason | O-P | (CS + AS) → AC |
| | Method | O-P | (CS + AS) → AC |

IB: infobox module, CS: category structure module, AS: article structure module, AC: article content module, DEF: definition module.

**Table 5**
Procedure for sequence invocation of two QA modules $M_1$ and $M_2$.

```
Procedure sequence invocation

Input: Module $M_1$, Module $M_2$, Question $q$
Output: Answer $best\_answer$

Answer $best\_answer$ = NULL
Answer $a_1$ = the answer of the highest score generated by $M_1$ for $q$
IF $S_{M1}(a_1|q) > \text{threshold}$
THEN
    $best\_answer = a_1$
ELSE
    Answer $a_2$ = the answer of the highest score generated by $M_2$ for $q$
    IF $S_{M2}(a_2) > \text{threshold}$
    THEN
        $best\_answer = a_2$
    ENDIF
ENDIF
RETUEN $best\_answer$
```

**Table 6**
Procedure for parallel invocation of two QA modules $M_1$ and $M_2$.

```
Procedure parallel invocation

Input: Module $M_1$, Module $M_2$, Question $q$
Output: Answer $best\_answer$

Answer $best\_answer$ = NULL
Answer[ ] $a_1\_list$ = Top $n$ answers generated by $M_1$
Answer[ ] $a_2\_list$ = Top $n$ answers generated by $M_2$
Answer[ ] $merge\_list$ = EMPTY LIST
FOR each answer $a$ in $a_1\_list$ and $a_2\_list$
    IF $a$ is an common element of both $a_1\_list$ and $a_2\_list$
    THEN $Score(a) = S_{M1}(a) + S_{M2}(a)$
    ELSE IF $a$ is an element of $a_1\_list$
    THEN $Score(a) = S_{M1}(a)$
    ELSE $Score(a) = S_{M2}(a)$
    ENDIF
    Add $a$ into $merge\_list$
ENDFOR
Sort answers in $merge\_list$ by $Score(a)$ in decreasing order
$best\_answer$ = the first answer in $merge\_list$
RETURN $best\_answer$
```

two modules are merged based on the parallel invocation algorithm (Table 6). If the score of top-ranked answer is higher than predetermined threshold, the answer is final by the sequential invocation algorithm (Table 5), otherwise, CS and AC modules are invoked in parallel. We set the thresholds of QA modules based on repeated experimental results.

## 4. Experiment

We downloaded Korean Wikipedia from a Wikipedia dump site.[1] The articles were preprocessed into predefined document types, as shown in Table 7. All the documents were analyzed and indexed using multi-level linguistic analysis techniques such as POS tagging, and named entity tagging. Evaluations were made by three human judges who understand the functionality of QA modules. When the decisions conflicted we followed the majority. For factoid questions, we used the TREC "exact answer" criterion. For descriptive questions, our evaluation was based on whether candidate answer sentences contain "key phrases," similar to the TREC "nugget" criterion (Dang & Lin, 2007). For an effective comparison, we employed the mean reciprocal rank (MRR), precision, recall, and F-score. We grouped our QA modules as follows:

- *Group 1*: a traditional QA module for article contents (AC, Baseline).
- *Group 2*: single modules including the infobox module (IB), category structure module (CS), article structure module (AS), and definition module (DEF).
- *Group 3*: dual combined modules including a merged module of AC and IB (+IB), AC and CS (+CS), AC and AS (+AS), AC and DEF (+DEF).
- *Group 4*: a merged module of all modules (+ALL).

[1] http://dumps.wikimedia.org, 2010/11/04 ver. for Korean Wikipedia.

**Table 7**
Statistics of Wikipedia documents.

| Document type | Num. of documents |
|---|---|
| Article contents | 430,829 |
| Article structure | 430,829 |
| Definition | 128,634 |
| Category structure | 128,634 |
| Infobox | 49,489 |

**Table 8**
Overall evaluation results (total number of questions: 600).

| | Module | #Response | #Correct | MRR | P (%) | R (%) | F (%) | Improvement[a] (%) | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | MRR | F |
| G1 | AC (BASELINE) | 205 | 162 | 0.840 | 79.0 | 27.2% | 40.2% | – | – |
| G2 | IB | 20 | 17 | 1.000 | 85.0 | 2.8 | 5.5 | – | – |
| | CS | 32 | 27 | 1.000 | 84.4 | 4.5 | 8.5 | – | – |
| | AS | 31 | 29 | 0.800 | 93.5 | 4.8 | 9.2 | – | – |
| | DEF | 99 | 93 | 1.000 | 93.9 | 15.5 | 26.6 | – | – |
| G3 | +IB | 223 | 178 | 0.857 | 79.8 | 29.7 | 43.3 | 2.1 | 7.5 |
| | +CS | 234 | 188 | 0.860 | 80.3 | 31.3 | 45.1 | 2.6 | 12.0 |
| | +AS | 223 | 184 | 0.841 | 82.5 | 30.7 | 44.7 | 0.2 | 11.1 |
| | +DEF | 299 | 252 | 0.897 | 84.3 | 42.0 | 56.1 | 6.8 | 39.3 |
| G4 | +ALL | 363 | 316 | 0.910 | 87.1 | 52.7 | 65.6 | 8.3 | 63.1 |

[a] Improvement over the baseline.

### *4.1. Overall results*

The overall evaluation results are shown in Table 8. The +ALL module shows the highest performance as expected. When a baseline module was chosen for the given query set, a total of 162 answers were correct. On the other hand, 316 correct answers were returned when all the modules were invoked and the answers merged. This indicates that the additional correct answers were extracted from semi-structured knowledge sources. All modules in group 2 show high precision but low recall. This means that we can easily extract correct answers from the semi-structured information, but coverage of the information is still low. All modules in group 3 show higher precision and recall than the baseline module. This means that each semi structured information source covers its own unique types of questions. For example, infobox has the advantage of finding an entity's properties such as "Who designed the Eiffel Tower?" In particular, the +DEF module showed the highest F-measure over other modules in group 3. This indicates that our question set includes many definitional questions, and Wikipedia covers many of the concepts or entities referred to in these questions. The +AS module can find answers for complex questions for which it is difficult to identify an answer theme. The +CS module can find a list of entities as answers, as most Wikipedia categories represent abstract concepts and have member entities.

All modules show high MRR scores of between 0.840 and 0.910 in groups 1, 3, and 4. Single modules in group 2 show 1.0 MRR scores except the AS module. Because we focused on precision rather than recall, we set a high threshold in the answer matching modules. Although our system extracts a maximum of five answers for a question, the high threshold values cut off answer candidates even if they are in the fifth rank. Therefore, the improvement in MRR of the +ALL module is only 8.3%, which is relatively lower than the improvement of the F-measure.

### *4.2. Effects on different question types*

We analyzed the effects of Wikipedia's semi-structure information on different types of questions: factoid, list, and descriptive questions. Table 9 shows a comparison among the six methods for the three question types. The AC (baseline) module shows a lower performance than the +ALL module in all question types. The +IB module improved the performance of the factoid questions, and generated 18 additional answers, 16 of which are correct. The +IB module does not generate additional answers for list or descriptive questions. The +CS module improved the performance of list questions, responding to 24 more questions. The +AS module improved the performance of factoid and descriptive questions. The module responded with correct reason- and method-type answers to 10 more questions. The +DEF module responded to 92 more descriptive questions, showing the highest impact in our experiment.

### *4.3. Error analysis*

Since multiple components involved in answering process, an incorrect answer should be traced back to identify the first place where the error occurred (Moldovan et al., 2003). We analyzed errors based on the factors described in Oh et al. (2009).

**Table 9**
Comparison among different question types.

| AF | | AC (Baseline) | +IB | +CS | +AS | +DEF | +ALL |
|---|---|---|---|---|---|---|---|
| Factoid | #Response | 115 | 133 | 118 | 123 | 115 | 143 |
| | #Correct | 98 | 114 | 98 | 110 | 98 | 126 |
| | P/R/F (%) | 85.2/44.3/58.3 | 85.7/51.6/64.4 | 83.1/44.3/57.8 | 89.4/49.8/64.0 | 85.2/44.3/58.3 | 88.1/57.0/69.2 |
| List | #Response | 56 | 56 | 80 | 56 | 58 | 82 |
| | #Correct | 42 | 42 | 66 | 42 | 42 | 66 |
| | P/R/F (%) | 75.0/34.7/47.5 | 75.0/34.7/47.5 | 82.5/54.5/65.7 | 75.0/34.7/47.5 | 72.4/34.7/46.9 | 80.5/54.5/65.0 |
| Descriptive | #Response | 34 | 34 | 36 | 44 | 126 | 138 |
| | #Correct | 22 | 22 | 24 | 32 | 112 | 124 |
| | P/R/F (%) | 64.7/8.5/15.1 | 64.7/8.5/15.1 | 66.7/9.3/16.3 | 72.7/12.4/21.2 | 88.9/43.4/58.3 | 89.9/48.1/62.6 |
| Total | #Response | 205 | 223 | 234 | 223 | 299 | 363 |
| | #Correct | 162 | 178 | 188 | 184 | 252 | 316 |
| | P/R/F (%) | 79.0/27.0/40.2 | 79.8/29.7/43.3 | 80.3/31.3/45.1 | 82.5/30.7/44.7 | 84.3/42.0/56.1 | 87.1/52.7/65.6 |

**Table 10**
Error analysis in Wikipedia QA.

| Component | | # Error (%) |
|---|---|---|
| Question analysis | Linguistic analysis | 4 (8.5) |
| | Answer format analysis | 6 (12.8) |
| | Answer theme analysis | 8 (17.0) |
| | Question target analysis | 3 (6.4) |
| Document indexing/retrieval | | 10 (21.3) |
| Answer selection | Set top 5 as cut-off | 13 (27.7) |
| | Answer merging | 3 (6.4) |
| Total | | 47 (100.0) |

Table 10 shows the analysis results of 47 errors. The top-5 cut-off strategy charges the largest proportion (27.7%), as many correct answers were cut-off, especially for factoid questions in the AC module. The answer theme analysis and document indexing/retrieval module generated a considerable number of incorrect answers (38.3%) mostly in the AC module where possible answers in a document should be preliminarily indexed with their entity types. This observation indicates that a traditional QA module based on simple text documents shows lower performance than rich knowledge based modules. Document retrieval module can be improved by utilizing Wikipedia title matching strategy rather than simple content matching strategy. Errors in the question format analysis also induced incorrect answers (12.8%). When the incorrect question format is identified, the system relies on an unreliable answer finding strategy. Answer merging strategy induced three wrong answers (6.4%), when prior module generated incorrect answers with high confidence in the module invocation sequences. Because IB module is more reliable than others for factoid questions, errors in IB module likely propagate to final decision. However the percentage of the strategy error is relatively lower than other error types. This means that the modules have their own strength to specific question types, and the strict answer merging strategy is effective.

## 5. Conclusion

The main motivation behind this work was to devise a way to utilize the existing semi-structured, large-size Wikipedia database as a knowledge source for a QA system without building high-cost knowledge base. To this end, we categorized the Wikipedia structure into article contents, infoboxes, category structures, article structures, redirections and definitions. While we focused on the utilization of Wikipedia knowledge source in a traditional QA framework, more research is required for specialized question analysis and document retrieval modules for the types of knowledge sources. Also we will build and English QA system using the proposed method and compare the performance based on famous benchmark such as TREC datasets.

## References

Ahn, D., Jijkoun, V., Mishne, G., Müller, K., Rijke, M., & Schlobach, S. (2004). Using Wikipedia at the TREC QA track. In *Proceedings of TREC 2004*.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becke, C., Cyganiak, R., et al (2009). DBpedia – A crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 154–165.

Buscaldi, D., & Rosso, P. (2006). Mining knowledge from Wikipedia for the question answering task. In *Proceedings of 5th international conference on language resources and evaluation* (pp. 727–730).

Dang, H.T., & Lin, J. (2007). Different structures for evaluating answers to complex questions: Pyramids won't topple, and neither will human assessors. In *Proceedings of the 45th ACL* (pp. 768–775).

Dang, H.T., Kelly, D., & Lin, J. (2007). Overview of the TREC 2007 question answering track. In *Proceedings of TREC 2007*.

Demner-Fushman, D., & Lin, J. (2007). Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics, 33*(1), 63–103.

Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jörg, B., et al (2007). Question answering from structured knowledge sources. *Journal of Applied Logic, 5*(1), 20–48.

Hoffart, J., Suchanek, F. M., Berberich, K., & Weikum, G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence, 194*, 28–61.

Khalid, M.A., & Verberne, S. (2008). Passage retrieval for question answering using sliding windows. In *Proceedings of workshop: information retrieval for question answering* (pp. 26–33).

Lee, C., & Jang, M.-G. (2011). A prior model of structural SVMs for domain adaptation. *ETRI Journal, 33*(5), 712–719.

Lee, C., Hwang, Y.-G., Oh, H.J., Lim, S., Heo, J., Lee, C.-H., et al (2006). Fine-grained named entity recognition using conditional random fields for question answering. In *Proceedings of Asia information retrieval symposium* (pp. 581–587).

Moldovan, D., Paşca, M., Harabagiu, S., & Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions of the Information Systems, 21*(2), 133–154.

Nastase, V., & Strube, M. (2008). Decoding Wikipedia categories for knowledge acquisition. In *Proceedings of AAAI08* (pp. 1219–1224).

Oh, H.J., Myaeng, S.H., & Jang, M.-G. (2009). Enhancing performance with a learnable strategy for multiple question answering modules. *ETRI Journal, 31*(4), 419–428.

Oh, H.J., Sung, K.-Y., Jang, M.-G., & Myaeng, S.H. (2011). Compositional question answering: A divide and conquer approach. *Journal of Information Processing & Management, 47*(6).

Ponzetto, S.P., & Strube, M. (2007). Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22st national conference on artificial intelligence* (pp. 1440–1445).

Simmons, J. (2012). *True knowledge: The natural language question answering Wikipedia for facts*. <http://www.evi.com/>.

Suchanek, F.M., Kasneci, G., & Weikum, G. (2007). YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of WWW* (pp. 697–706).

Voorhees, E.M. (2004). Overview of the TREC 2004 question answering track. In *Proceedings of TREC 2004* (pp. 52–62).