Question Answering
Considering Semantic Categories and Co-occurrence Density

Soo-Min Kim, Dae-Ho Baek, Sang-Beom Kim, Hae-Chang Rim
Dept. of Computer Science and Engineering, Korea University
{smkim,daeho,sbkim,rim}@nlp.korea.ac.kr

Abstract

In this paper, we present a Question Answering system called KUQA (Korea University Question Answering system) developed by using semantic categories and co-occurrence density. Semantic categories are used for computing the semantic similarity between a question and an answer, and co-occurrence density is used for measuring the proximity of the answer to the words of the question. KUQA is developed based on the hypothesis that the words that are semantically similar to the question and locally close to the words appeared in the question are likely to be the answer to the question.

1. Introduction

Question Answering (QA) is defined to find the exact answer to the user's question in a large text collection. In other words, the answer is not the whole document that is relevant to the question, but the parts of the document that can meet the users' need more precisely. On the other hand, current IR systems allow us to locate documents but most of them leave it to the user to extract the information from top ranked documents. Recently, documents have rapidly increased in number, and we need a system that can retrieve *information*, not document. As a result, there has been a growing interest to QA in NLP community.

In this paper, we introduce the KUQA system developed by NLP Lab. in Korea University for the QA track of TREC-9. We try to incorporate NLP techniques with conventional IR techniques. To do this, we utilize WordNet as a kind of linguistic knowledge and a POS tagger for linguistic analyzer.

In the next section, we describe three components of KUQA system. In section 3, we analyze the performance of the system. And finally, we discuss future work in section 4.

2. System Description

Our system consists of three modules: the question analysis module for capturing the meaning of a natural language question, the document retrieval and analysis module for selecting

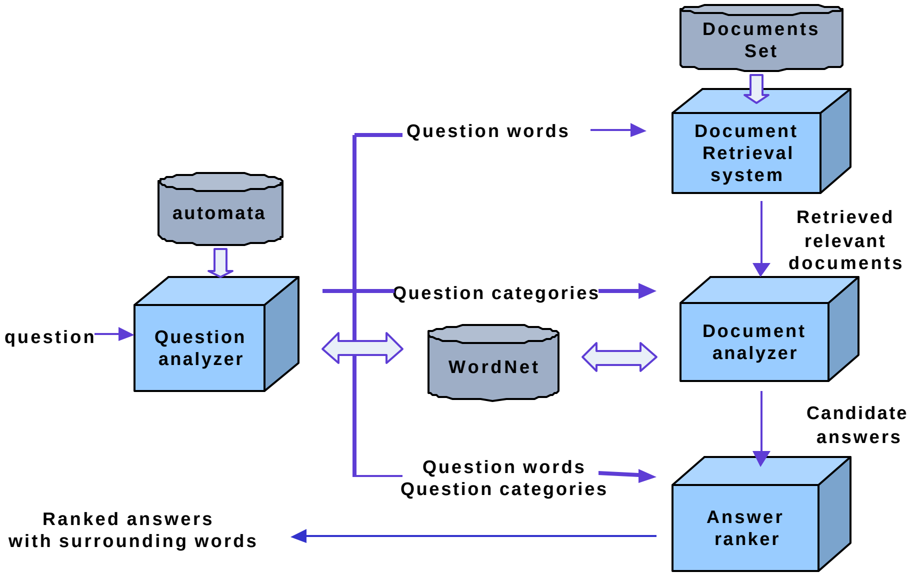

Figure 1: Architecture of KUQA system

candidate answers from documents, and the answer extraction module for ranking candidate answers and extracting surrounding words. These modules integrated into KUQA system are represented in Figure1. Each module is described in detail in subsequent sections.

2.1 Question Analysis

The question analyzer reads a question, determines the semantic categories of it by consulting automata and WordNet, and produces a category vector which represents the semantic categories assigned to the question. These classified question categories are used for computing semantic similarity between the question and candidate answers. Also, a list of question words is extracted from the given question, and it is used for retrieving relevant documents and measuring co-occurrence density.

2.1.1. Classifying Question Categories

The question categories indicate the possible type of semantic categories with which the expected answer corresponds. They are decided by different methods according to the types of questions. Questions can be grouped into following three types depending on their interrogatives and sentence structures:

(1) Who, Where, When
(2) How
(3) What, Which, Others

question		Question categories
Who		PERSON
Where		COUNTRY CITY CAPITAL PENINSULA ISLAND CONTINENT PROVINCE MOUNTAIN MOUNTAIN_PEAK RIVER OCEAN
When		DAY YEAR TIME_PERIOD TIME_UNIT TIME
How	far , tall	LENGTH LINEAR_UNIT
	long	LENGTH LINEAR_UNIT YEAR TIME TIME_PERIOD TIME_UNIT TIME
	rich	MONETARY_VALUE MONETARY_UNIT ECONOMIC_CONDITION FINANTIAL_LOSS
	much , many	NUMBER
What Which others	Applied to proper automata	The semantic categories of key phrase
	No automata	No category

Table 1: Category assignment table

Table 1 shows the categories assigned to each type of questions. The category of the question with an interrogative *Who, Where, or When* is decided by its meaning of the interrogative. The category of the question with the interrogative *How* is determined by the meaning of an adjective or an adverb followed by the interrogative. For example, *How long* questions may have categories related to time or length and *How many* questions may have categories related to number. However, the categories of *What, Which* and other questions can't be determined just by the meaning of the interrogative or an adjacent adjective or adverb. To analyze these questions, we try to manually construct automata. By using the automata, the system recognizes the key phrase of the question, and then assigns the semantic categories of the question based on the semantic categories of the key phrase. The semantic categories of the key phrase are classified into one of 46 preclassified categories by using WordNet.

Figure 2 shows an example of the process of assigning question categories to the question: *What is the fare cost for the round trip between New York and London on Concord?* In this example, the key phrase "fare cost" is recognized by the automata, and the semantic category of the question is classified into "FINANCIAL LOSS" by using WordNet.

Category vector

Question categories are represented by a *category vector*. The category vector consists of 46 categories manually selected from words in WordNet. If only one category is assigned to a

(Number of categories: 46)

COUNTRY	CITY	PENINSULA	PERSON	LENGTH	....	FINANTIAL_LOSS	MONETARY_UNIT
1	1	1	0	0	.....	0	0

Table 2: An example of category vector for the question "*Where is the Orinoco*?"

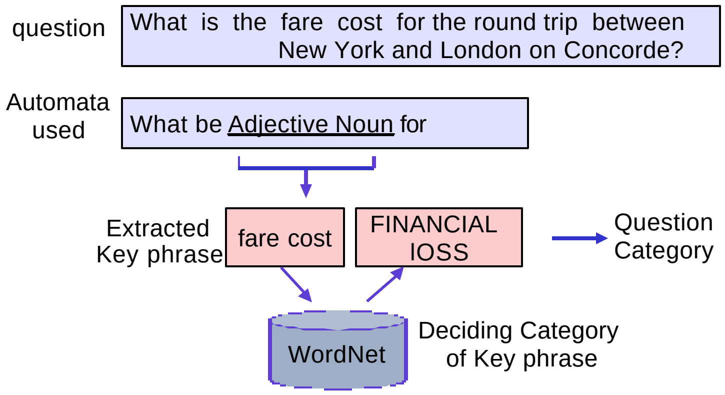
question
What is the fare cost for the round trip between New York and London on Concorde?
Automata used
What be Adjective Noun for
Extracted Key phrase
fare cost
FINANCIAL IOSS
Question Category
WordNet
Deciding Category of Key phrase

Figure2: an example of the process of assigning a question category

question, the value of that category in the category vector is set to 1, and all other categories in the vector are set to 0. If several categories are assigned to a question, then those categories in the vector are set to 1.

Table 2 shows an example of category vector for the question: *Where is the Orinoco?* Since the semantic categories of that question are related to the *location* category, like COUNTRY, CITY, PENINSULA, CONTINENT, and PROVINCE, all the categories related to *location* are set to 1 in the vector.

2.1.2 Extracting Question Words

In the question analysis module, a list of question words is also extracted from the given question, and it is used for retrieving relevant documents and measuring co-occurrence density. A question word is extracted from a question if their part-of-speech is noun, verb, adjective, adverb, or cardinal number, and it is restored to its root form. For example, the list of questions words <fare, cost, round, trip, New, York, London, Concord> is extracted from the question "What is the fare cost for the round trip between New York and London on Concord?"

2.2. Relevant Document Retrieval and Candidate Answer Selection

2.2.1 Retrieving Relevant Documents

The document retrieval module retrieves documents relevant to the question. The document retrieval system is basically implemented based on the OKAPI ranking. However, we assign different weights to the question words according to their part-of-speeches. When the part-of-speech of a question word is proper noun, then the weight of the question word is doubled so that the documents with the same proper noun are highly ranked.

2.2.2 Selecting Candidate Answers

We try to select several candidate answers from top 10 ranked documents according to their part-of-speeches.

As shown in table 3, we manually classify semantic categories of a question and their corresponding part-of-speeches. With one or more semantic categories of a question, we can expect the part-of-speeches of candidate answers by using table 3, and then select several candidate answers from top ranked documents based on their part-of-speeches.

In fact, the POS tagger does not tag properly to the word whose part-of-speech is cardinal number. So, we also select the word including a number in its string as candidate answers when the semantic categories of a question expects cardinal number for its candidate answers. For example, the word $600,000 can be a candidate answer when the semantic category of a question is FINANCIAL LOSS, because it contains a number in its string.

COUNTRY, CITY, CAPITAL, PENINSULA, ISLAND, CONTINENT, PROVINCE, MOUNTAIN, MOUNTAIN_PEAK, OCEAN, RIVER, . …… .	Proper Noun
COMPOUND, MATERIAL, DISEASE, SORT, WORD, BOOK, CINEMA, MOVIE, MUSIC, . …… .	Proper Noun Noun
NUMBER, LENGTH, LINEAR_UNIT, MAGNITUDE_RELATION, TIME, TIME_PERIOD, YEAR , MONETARY_VALUE, …..	Cardinal Number

Table 3: POS of candidate answers corresponding to semantic categories

2.2.3 Determining Semantic Categories of Candidate answers

We also use WordNet to determine the semantic categories of a candidate answer. Semantic categories of a candidate answer are also represented by a category vector in the same way as the question category vector. The vector consists of 46 categories manually chosen from a pool of words in WordNet.

The system obtains a set of hypernyms and synonyms of a candidate answer by using WordNet. If the set of hypernyms and synonyms of a word contains the words used as categories in the category vector, the system sets the values of those categories in the vector to 1.

Some categories used in the system can be grouped into the classes called *similar category classes*, as shown in table 4. If the category of the candidate answer belongs to one of the class of similar category classes, other categories in the same class are also set to 1 in the vector of that word. For example, the word *cost* has FINANCIAL_LOSS as its hypernym and FINANCIAL_LOSS belongs to one of the similar category classes. Thus, all other categories in the same class: MONETARY_VALUE, MONETARY_UNIT, ECONOMIC_CONDITION are also set to 1.

MOUNTAIN	MOUNTAIN_PEAK
MONETARY_VALUE FINANTIAL_LOSS	MONETARY_UNIT ECONOMIC_CONDITION
TIME TIME_UNIT	TIME_PERIOD
CINEMA	MOVIE
LENGTH	LINEAR_UNIT
WORD	NAME
CAPITAL	CITY

Table 4: Similar category classes

In some cases, two or more words have one category. In the case of words *New York*, although the category of New York is CITY by using WordNet, each word *new* and *York* doesn’t belong to the CITY category. To solve this problem, we consider not only the current keyword but also the adjacent words of the current keyword in assigning keyword categories. If there are proper categories for two adjacent words in WordNet, the categories are assigned to the candidate answer.

There are many named entities which are unknown in WordNet. In order to determine the semantic categories of the unknown named entities, we try to use the semantic categories of the adjacent word of the unknown named entity as a clue. As a simple example, consider the phrase: *President Kim said.* The category of the named entity *Kim* can’t be determined by using WordNet. But, the category of the preceding word *President* can be determined as PERSON, and the category of unknown named entity *Kim* can be also determined as PERSON.

In the case that several words are connected by hyphens, or a number and a unit together comprise one word, we have to tokenize them as separate words. We divide *92km*, for example, as *92* and *km, and* then determine a category of *92km*. Table 5 shows some examples of words and their corresponding categories.

President Steven	PERSON
New York	CITY
Seoul	CITY
92m	LENGTH, NUMBER
5 may	TIME_PERIOD , NUMBER
$600,000	ECONOMIC_CONDITION FINANCIAL_LOSS MONETARY_VALUE

Table 5: Some examples of words and their categories

2.3 Ranking Candidate Answers

We use three factors to rank candidate answers: average distance weight, co-occurrence ratio, and semantic category similarity. Candidate answers are ranked according to the product of these three factors. By doing that, both semantic category similarity and co-occurrence density are reflected in computing the similarity between a question and an answer.

2.3.1 Average distance weight

By considering the phenomena that an answer to some questions tends to appear in a document locally close to the same words occurred in the question, we use the average distance weight to measure the proximity. Distance weight means the degree of proximity between the keywords of the candidate answer and words in the set of question words. It varies between 0 and 1. Average distance weight is determined by computing the average of distance weight between one candidate word and all question words in the fixed number of words around a candidate answer in a document.

2.3.2 Co-occurrence ratio

Although two candidate answers have the same value of average distance weight, one clustered with many words in the set of question words must have a higher score than the other clustered with just a few words. This is reflected in the following formula of R_i :

$$R_i = \frac{\text{Number of question words appeared in the passage}}{\text{Total Number of question words}}$$

2.3.3 Semantic category similarity

Average distance weight and the co-occurrence ratio are not able to reflect the semantic similarities between a question and a candidate answer. Thus, we define the *category similarity* between a question category vector and a candidate answer category vector. It can have one of three values: high, middle, or low. When two categories are same or similar, the category vector similarity is high. When there is no relevance to each other, the category vector similarity is low.

3. Experimental Results

Our system uses the question set used in TREC8 as a training data. It uses TreeTagger (Helmut Schmid) as a POS tagger and WordNet as a thesaurus. The document retrieval system implemented by using OKAPI algorithm is used for retrieving relevant documents. Our TREC-9 results of 250-byte run are shown in table 6. There are 682 questions in TREC-9 test questions. Unlike the last year, the judgment field can be one of three values: -1 (*Wrong*), 1 (*Correct*), and 2 (*Unsupported*). The Unsupported judgment is given to responses that would have been judged correct but, in the judge's opinion, we could not tell it was a correct answer from the document returned with it.

There are two different evaluations: a strict evaluation which counting only the *Correct* as right and a lenient evaluation that counting both *Correct* and *Unsupported* as right. The first row of the table 6 indicates the result of the strict evaluation and the second row indicates that of the lenient evaluation.

(Total number of test questions: 682)

	Number of answers						MRR	Percentage of correct answers in top 5
	rank1	rank2	rank3	rank4	rank5	Total		
Strict	194	78	35	23	14	344	0.371	50.40%
Lenient	206	74	36	21	16	353	0.386	51.80%

Table6: Result for the 250-byte answer category

4. Discussion and Future Work

In this paper, we introduced our Question-Answering system, named KUQA. With the system, we tried to integrate NLP techniques and IR techniques in a way that makes maximal use of their complimentary ability. KUQA utilizes WordNet as a source of word class information and TreeTagger as a tool for linguistic analysis. Experimental results are encouraging and suggest that NLP techniques are useful for Question-Answering. There is certainly much room for improvement. A problem arises with questions or candidate answers containing words unknown to WordNet. Their semantic categories cannot be classified properly. Another problem arises from limited utilization of NLP techniques. In the future work, we will extend our system to include various NLP techniques including partial parsing, named entity tagging and anaphora resolution.