iQA: An Intelligent Question Answering System

Zhiguo Gong[1] and Mei Pou Chan[1,2]

[1] Faculty of Science and Technology,
University of Macau,
Macao, PRC
[2] Macao Polytechnic Institute,
Macao, PRC
zggong@umac.mo, calanachan@ipm.edu.mo

Abstract. Question answering (QA) is the study on the methodology that returns exact answers to natural language questions. This paper attempts to increase the coverage and accuracy of QA systems by narrowing the semantics gap between questions with terms written in abbreviations and their potential answers. To achieve this objective, the processing includes (1) identifying terms that might be abbreviations from the user's natural language question; (2) retrieving documents relevant to that abbreviation term; (3) filtering noun phrases that are considered to be potential long forms for that abbreviation within the returned result.

1 Introduction

Question answering (QA) is the study on the methodology that returns exact answers to natural language questions, rather than a list of potentially relevant documents, which users have to scan through in order to dig out the necessary information. In other words, question answering is a step closer to information retrieval rather than document retrieval.

The challenge with QA system is how to return answers to user's natural language questions. The whole process is quite complicated as it involves quite a number of different techniques to work closely together in order to achieve the goal, including query rewrites and formulations, question classification, information retrieval, passage retrieval, answer extraction, answer ranking and justification. The end-to-end performance of a complete QA system hence depends on each of these independent factors. Over the past few years, individual research groups have been continuing to refine each of these steps with the intention to increase the coverage and accuracy of QA systems.

Several question answering systems have been made available for use on the Web. However, these QA systems have not taken questions with terms written in abbreviations into consideration. They can present the relevant information among the returned answers in response to the question "*Where is University of Macau?*". However, when the abbreviation for "University of Macau" is used instead, that is, when *"Where is Umac?"* is submitted, though the returned result set contains information related to "umac", it has nothing to do with "University of Macau" because

E.A. Fox et al. (Eds.): ICADL 2005, LNCS 3815, pp. 332 – 341, 2005.

the abbreviation "umac" can stand for many different things besides "University of Macau". In other words, system-wise speaking, these answers are justified to be correct. Only that in this case, it so happens that the semantics of the returned answers does not meet the expectation need of the user.

This work addresses this problem by attempting to reduce the semantics gap between questions with terms written in abbreviations and the potential answers. WordNet[1] is used here to help solve this problem. The terms in the user's natural language question will be sent to WordNet. Due to the large coverage of WordNet on English-language word, terms that cannot be found in WordNet will be considered as abbreviations, though there are cases when this is not true. A query solely consisting of that term will be sent to a search engine, and the retrieved relevant documents will be processed to obtain the possible long forms for that term to be used in feedback loop for the user's original question. The details of this process will be discussed in the later sections.

The rest of the paper is organized as follows: section 2 briefly talks about the motivation of this work; section 3 details the implementation of the proposed QA system—iQA, including the solution to reduce the semantics gap between questions with terms written in abbreviations and the potential answers, and section 4 is the evaluation and conclusion of this work.

2 Motivations

Question answering systems have their history dated back to the 1960's, using highly edited knowledge bases, edited list of FAQ, sets of newspaper articles and encyclopedia as knowledge base. Given the limited size of these corpora, it is necessary to have a deep understanding on the language in order to find an answer to a question because the chance of finding strings/sentences that closely match the question string within a relatively small textual collection is small. Thus, many complex natural language processing (NLP) techniques have to be used because syntactic information about how a question is phrased and how sentences in documents are structured potentially provides important clues for the matching of the question and answer candidates in the sentences, e.g. in discovering the sentence *"Columbus Day celebrates the Italian navigator who first landed in the New World on Oct 12, 1492."* to be the answer to the question *"Who discovered America?"*.

However, the emergence of the Web has made way for a brand new perspective for question answering systems. Given the Web's huge data size, it is highly possible that an answer string that occurs in a simple relation to the question exists in the Web. Hence, the degree of difficulty of question answering systems does not primarily depend on the question per se, but rather on how closely a given corpus matches the question. Taking an example given by Hermjakob et al. [1] as a demonstration,

Q: Who discovered America?
S1: Columbus discovered America.

[1] WordNet is an online lexical reference system developed at Princeton University's Cognitive Science Laboratory by Psychology Professor George Miller. It is an extensive English-language word database developed over the last thirty years.

S2: Columbus Day celebrates the Italian navigator who first landed in the New World on Oct 12, 1492.

The question above can be answered more easily from sentence S1 than sentence S2 because the string Q is "closer" to string S1 than string S2. Since the Web's size dwarfs any human-collected corpora by orders of magnitude, it is not uncommon to have the same piece of information written and expressed in various ways. This property can be exploited to eliminate the need to understand both the structure and meaning of natural language, yet, be able to extract the answer. The greater the redundancy in the source, the more likely an answer can occur in a simple relation to the question, without the need to solve the difficulties with NLP systems.

In fact, several studies [2, 3] have shown that by consulting the Web as the knowledge base, QA systems can still achieve a satisfactory level of performance without the need to solve the difficulties with NLP systems. The trick is to take advantage of the redundancy of data present in the Web and use simple pattern matching techniques.

The possibility of finding an answer to a factoid question without the need for a deep understanding of the language forms the launching pad for this work. Driven by the stimulation of TREC QA track[2], individual research groups have worked on refining each of the various components that make up a complete QA system, with the intention to increase the coverage and accuracy of QA systems.

Sharing the common goal as to increase the coverage and accuracy of QA systems, the driving force behind this work is to develop a QA system that can precisely answer users' factoid questions stated in the form of natural language, using the Web as the knowledge base. In particular, the system should handle questions with terms written in abbreviations, thus narrowing the semantics gap between the questions and the potential answers.

Everyone agrees that an abbreviation term can stand for many different things depending on the context used. For instance, taking the term ATM as an example. For the general public, the first concept that comes to their mind might be "automatic teller machine", from which they enjoy convenient money withdrawal service from time to time. Nevertheless, for people working in the computer network area, ATM implies "asynchronous transfer mode" in their day-to-day work conversation.

Hence, if a QA system does not keep this in mind, though the returned answers are related to ATM, which is considered to be correct system-wise speaking, might yet be considered to be an incorrect answer by the user because the user has a different expectation on the context of the abbreviation term.

In order to reduce this semantics difference, iQA includes questions with terms written in abbreviations into consideration. The proposed procedures for attacking this particular challenge are as follows: (1) identifying terms that might be abbreviations from the user's natural language question; (2) retrieving documents relevant to that abbreviation term; (3) filtering noun phrases that are considered to be potential long forms for that abbreviation within the returned result.

[2] TREC (the Text REtrieval Conferences)[2] is a series of workshops co-sponsored by the National Institute of Standards and Technology (NIST) and DARPA (Defense Advanced Research Projects Agency).

A complete QA system, iQA, will be developed incorporating the proposed solution for questions with terms in abbreviations into it so as to increase the coverage of the system. iQA will use shallow parsing techniques to obtain the necessary information for answer extraction. Shallow parsing, also called partial parsing or chunking, is the task of identifying phrases, possibly of several types, in natural language sentences based purely on part-of-speech tags and without a deeper understanding of the content. It is simpler, conceptually and computationally, than full parsing, but still provides fundamental sentence structure information such as noun phrases and verb phrases. The following sections will detail the steps needed to build such a system.

3 Methodologies in Implementing iQA

The overall architecture of our proposed system, iQA, can be divided into four main modules: (1) question analysis module, (2) document analysis module, (3) answer extraction module, and (4) abbreviation analysis module. Figure 1 shows the system architecture of iQA. The functions performed by each of these modules are discussed below.

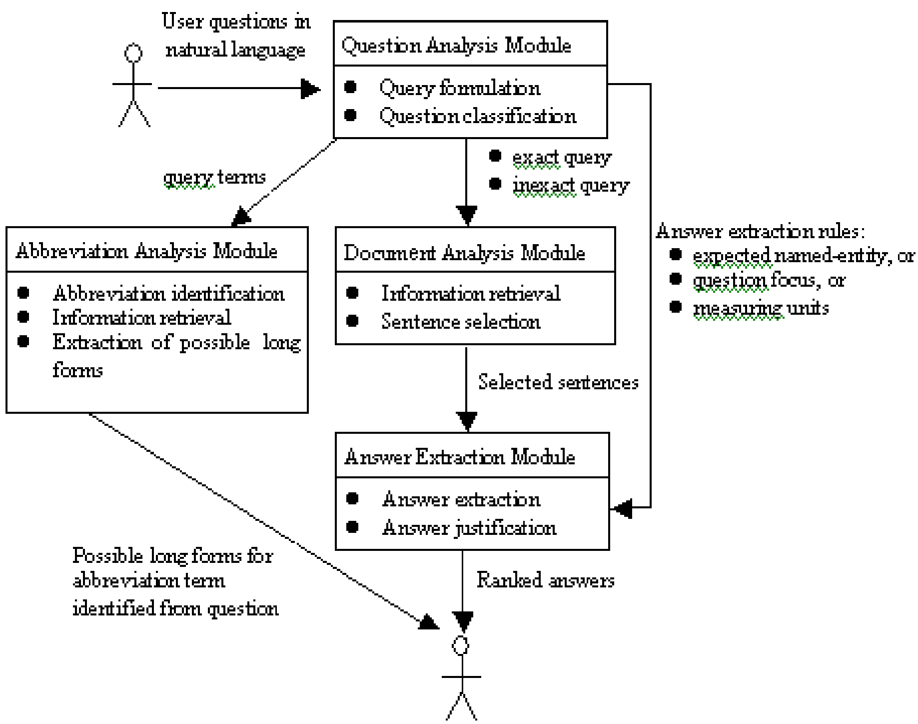

Fig. 1. System architecture of iQA

3.1 Question Analysis Module

Question analysis module is mainly consisted of two operations: query formulation, and question classification. As the name suggests, query formulation handles the formulation of queries to be used in the document analysis module. The query formulation process of iQA involves the generation of two types of queries, exact queries and inexact queries, similar to the idea proposed by [4].

An exact query is composed by simply removing the question stem from the factoid question, and re-arranging the position of the verb if necessary, based on the idea from [5]. The syntactic structure of the question has to be analyzed in order to re-position the verb accordingly. For example, the exact query for *"What are pennies made of?"* is *"pennies are made of"*.

Brill et al. [2] has proposed a much more exhaustive rewrite approach. Given a query such as "Who is w_1 w_2 … w_n", where each of the w_i is a word, a rewrite is generated for each possible position the verb could be moved to (e.g. "w_1 is w_2 … w_n", "w_1 w_2 is …w_n", etc). This approach guarantees that the proper movement position is found.

Exact query has a low recall but a high precision rate because if the retrieved documents contain sentences that match with the exact query, it is highly possible that a potential answer can be located within it. In fact, this is an attempt to fully exploit the data redundancy property of the Web where it is possible that a sentence written in the form that closely matches the question can be found.

An inexact query is based on the belief that an answer is likely to be found within the vicinity of a set of query terms. An inexact query is composed by treating the natural language question as a bag of query terms. Given a factoid question, a query in a form of $q^{(0)} = [q_1^{(0)}, q_2^{(0)}, \ldots q_k^{(0)}]$ is produced, where each query term $q_i^{(0)}$ might be a noun phrase, a verb, an adjective, or any other content words present in the question. For example, the inexact query for *"What city is the home to the Rock and Roll Hall of Fame?"* is [city, home, Rock and Roll Hall of Fame]. Such kind of query has a higher recall but a lower precision rate compared to the result returned by using an exact query.

In our system, queries of both types are generated and used to retrieve data for processing. The two types of queries supplement each other to maintain a balance between recall and precision.

As for the operation of question classification, the goal is to analyze the questions to derive the detailed question classes and the expected answer type in terms of named-entity. This information enables the later process of extracting the exact answer from the candidate sentences more accurately. When a system is aware of being asked a when-question, it can focus on time or date as potential answers, and a where-question is asking for location, etc.

However, not all questions allow the derivation of an expected answer type in terms of named-entity, including what, which, and how questions. For such cases, the question focus can be taken advantage of in guiding the later process of answer extraction. A question focus is a phrase in the question that disambiguates it and emphasizes the type of answer being expected. In most cases, the head (main noun) of the first noun phrase of the question is the question focus. For instance, in question *"What book did Rachel Carson write in 1962?"*, the question focus *"book"* can be used to provide supporting clues to locate the answer in the later process, by looking for phrases containing the question focus.

Question classification, i.e., putting the questions into several semantic categories, can significantly reduce the search space of plausible answers. The accuracy of question classification is very important to the overall performance of a question answering system.

3.2 Document Analysis Module

After the question analysis module, the next step is document analysis module. The main objective of this module is to retrieve data from the Web based on the exact and inexact queries generated from the question analysis module, and then select sentences likely to contain an answer from the returned data for further processing.

iQA uses Google, an existing generic search engine, as information retrieval back-end, and effort can be mainly driven to answer extraction and justification process. In this case, post-processing procedures to extract the potential answers have to be implemented.

The returned snippets and summaries are split into separate sentences. In iQA, sentences that contain either the exact query, or the query terms of the inexact query are selected to be passed to the answer extraction module.

3.3 Answer Extraction Module

With the clues derived from question classification of the question analysis module, and the candidate sentences retained by the document analysis module, the answer extraction module is ready to start. The main objective of this module is extraction of simple potential answers, and answer justification.

If an expected answer type in terms of named-entity can be derived during the question analysis module, the candidate sentences retrieved from the document analysis module are first processed by a named-entity tagger in order to obtain phrases of expected answer type, which become the candidate answers. Otherwise, the answer extraction depends on the recognition of question focus or any other clues derived during the question analysis module.

These candidate answers are then scored based on the frequency each of them appears within the potential candidate answers retrieved. Clustering will then be done to group the similar candidate answers together so that shorter answers are used as evidence to boost the score of longer answers. Voting based on the number of scores will determine the rank of the potential candidate answers [3, 6]. That is, to use data redundancy to validate the accuracy of answer. The multiple occurrences of the same answer in different documents lend credibility to the proposed answer.

3.4 Abbreviation Analysis Module

After all the previous processing, the answers are now ready to be displayed. However, what if the user question contains terms written in abbreviations? The answers in hand are relevant to that abbreviation term. However, there are so many different interpretations to each abbreviation term, will the answers lie in the same context as expected by the users? This uncertainty drives the inclusion of the abbreviation analysis module to increase the coverage of QA systems. The work mainly includes (1) identifying terms that might be abbreviations from the user's natural language question; (2) retrieving documents relevant to that abbreviation term; (3) filtering noun phrases that are considered to be potential long forms for that abbreviation within the returned result.

3.4.1 Which are Abbreviation Terms?

Given a user's natural language question, how can the system automatically identify the abbreviation term? WordNet is used here to help solve this problem. Query terms consisting of one word only will be sent to WordNet. That is, if the user question is *"Where is umac?"*, the term "umac" will be sent to WordNet , whereas the term "University of Macau" in *"Where is University of Macau?"* will not be sent. This is based on the observation that terms consisting of more than one word are seldom abbreviations. Due to the large coverage of WordNet on English-language word, terms that cannot be found in WordNet will be considered as abbreviations, though there might be cases when this is not true.

When the thought-to-be abbreviation term has been filtered out, a separate query consisting solely of that abbreviation term will be sent to a search engine to retrieve documents relevant to that abbreviation term.

3.4.2 Filter Corresponding Potential Long Forms

In this step, the returned result from the search engine will be processed to obtain the possible long forms for that term to be used in feedback loop for the user's original question. Two approaches are adopted to filter the possible long forms.

The first approach is to retrieve all noun phrases that have the first character matches with the first character of the term being considered as abbreviation. Subsequent character matching is not necessary as in the case of "umac" for "University for Macau". The three characters together, "mac", represent the word "Macau".

However, this approach alone is not sufficient to solve the problem. There are cases where the first character of the abbreviation is not the same as the first character of the long form, as in the case of "CPTTM" for "Macau Productivity and Technology Transfer Center". Hence, the second approach is to take advantage of the patterns

<A; parenthesis; X; parenthesis> and
<X; parenthesis; A; parenthesis> and
<A; dash; X>

Example: "IFT (Institute for Tourism Studies)"

where A denotes the abbreviation and X denotes the possible long form. The pattern elements are divided by a semicolon. The idea of this pattern is based on the work of M.M. Soubbotin et al. [7]. Presence of such patterns in the noun phrase serves as an indication of the presence of an acronym.

Those noun phrases that are retained by these two approaches are then listed for users to choose as feedback loop to the original question, to return answers falling into the same semantics expectation of the users, hence, significantly boosting up the performance of QA systems.

3.4.3 Work in Action

The difference between iQA and the other available search engines in the Web is the consideration of questions with terms written in abbreviations. Taking the question "*Where is ift?*" as an example. After user inputs the question and presses the "Find Answer" button, the preliminary answers are displayed, as shown in the following figure. In addition, since the term "ift" is not included in WordNet, the possible long forms for "ift" will be retrieved and listed as well.

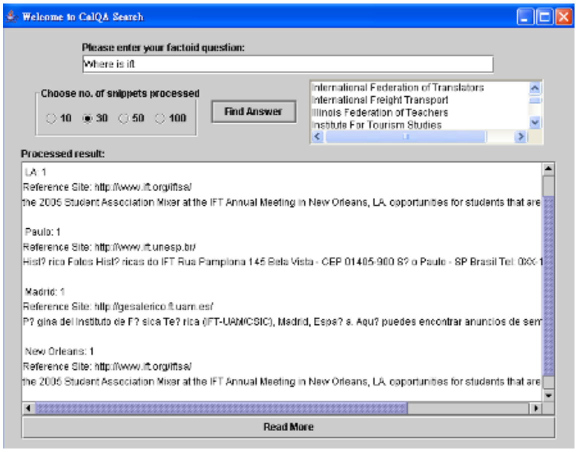

Fig. 2. Screen shot of iQA's response to "*Where is ift?*"

As seen from figure 2, the answers include LA, Paulo, Madrid, New Orleans. System-wise speaking, all these answers are relevant to the question "*Where is ift?*". This is a good example showing that one abbreviation term can have more than one interpretation.

However, in fact, "ift" stands for many other things in addition to those answers listed above. But, in order not to overload the users with too much information, which is exactly the goal of QA systems, only those answers within the top 5 rankings are displayed in iQA. This means that the "ift" users have in mind might not be taken into consideration within the answers displayed. To supplement this shortage, in addition to the answers displayed, the possible long forms that are available for the term "ift" are simultaneously listed for users to choose as feedback loop into the original question, as shown in figure 2.

For instance, if the user wants to know the location of "Institute for Tourism Studies" instead, the user can choose that option from the list and press "Find Answer" button again. In this case, "ift" will be substituted by "Institute for Tourism Studies" in the query sent to the search engine. The corresponding answers for *"Where is Institute for Tourism Studies?"* will then be displayed, as shown is figure 3.

Hence, with such an implementation, the coverage and accuracy of QA systems is increased, through narrowing the semantics difference between questions with terms written in abbreviations and the potential answers.

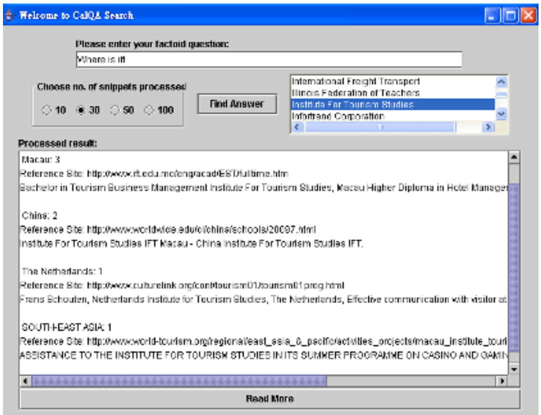

Fig. 3. Screen shot of iQA's response to "*Where is ift?*" using "Institute for Tourism Studies" in the feedback loop

4 Evaluation and Conclusion

In this paper, we address our methodologies in implementing iQA (an intelligent question answering system). Our system mainly includes 4 sub-systems, namely, question analysis module, abbreviation analysis model, document analysis module and answer extraction module.

Question analysis module takes users' natural language questions as input, and tries to classify the original questions into several groups. By consulting corresponding patterns or rules in the knowledge base of the system, the original questions are transformed into native queries to general-purpose search engines. Google is used as our only backend search engine in our current implementation. In order to handle the problems caused by abbreviation in user's questions, we develop abbreviation analysis module for this sake. As a matter of the fact, this sub-system uses WordNet firstly to learn if the terms in the questions are possible abbreviations. To expand the abbreviation to some potential phrases or concepts, this module tries to retrieve from Google by matching some patterns. Then, the abbreviation terms will be replaced in the original question with users' feedbacks. The document analysis module takes charge of document access through general-purpose search engine. And finally, the answer extraction module will try to mine out potential answers in response to the original question.

In order to evaluate the general performance of iQA, the first 100 TREC 2003 questions are sent to iQA, and the MRR is 0.49. Borrowing the idea used by TREC, the answer accuracy of the system is measured by the mean reciprocal rank (MRR) which assigns a number equal to 1/R where R is the rank of the correct answer. If none of the answers returned are correct, or if the system does not return any answers, the precision score for that question will be zero. The system's overall score is calculated as the MRR across all the 100 questions.

Since iQA has a particular function to handle questions with abbreviation terms, besides using the TREC questions as a general evaluation basis, questions with abbreviation terms present in the Macao local region are also used to test the performance of iQA. The evaluation result is displayed in table 1.

Table 1. Evaluation result on questions with abbreviation terms

Questions with abbreviation terms	Answers correct?	Remark – local context for Macao region
Where is Umac?	Yes	Umac – University of Macau
Where is IFT?	Yes	IFT – Institute for Tourism Studies
Where is IIUM?	Yes	IIUM – Inter-University Institute of Macau
Where is AIOU?	Yes	AIOU – Asia International Open University
Where is UNU-IIST?	Yes	UNU/IIST – United Nations University
Where is CPTTM?	Yes	CPTTM – Macau Productivity and Technology Transfer Center

References

1. U. Hermjakob, A.E., D. Marcu. *Natural Language Based Reformulation Resource and Web Exploitation for Question Answering*. in *TREC 2002*. 2002.
2. E. Brill, J.L., M. Banko, S. Dumais, A. Ng. *Data-Intensive Question Answering*. in *TREC-2001*. 2001.
3. Charles L. A. Clarke, G.V.C., Thomas R. Lynam, *Exploiting Redundancy in QA*. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001: p. 358-381.
4. J. Lin, A.F., B. Katz, G. Marton, S. Tellex. *Extracting Answers from the Web Using Data Annotation and Knowledge Mining Techniques*. in *TREC 2002*. 2002.
5. Dell Zhang, W.S.L., *Web based pattern mining and matching approach to question answering.*
6. Cody Kwok, O.E., Daniel S. Weld, *Scaling QA to the Web.* ACM transactions on Information Systems, 2001. **19**(3): p. 242-262.
7. M.M. Soubbotin, S.M.S. *Patterns of Potential Answer Expressions as Clues to the Right Answers*. in *TREC-2001*. 2001.