Asian Journal of Information Technology 5 (4) : 437-441, 2006

Factual Question Answering by Automatic Surface Pattern Learning Using Reformulation Rules

S.M. Rafizul Haque, Niazur Rahman and Shahabul Islam
Computer Science and Engineering Discipline, Khulna University, Khulna-9208, Bangladesh

Abstract: Many of the high performing factual question answering systems in the recent TREC (Text Retrieval Conference)'s use a fairly extensive list of surface text patterns. In this study, an automatic surface pattern learning using reformulation rules is proposed. In essence, this is an adaptation of surface pattern learning first proposed by Deepak Ravichandran and Hovy. In our proposed system, predefined sets of representative question and answer patterns, instead of question answer pairs, are used for answer extraction. The performance of the modified system is measured by using two conventional and standard metrics-MRR (Mean Reciprocal Rank) and precision. The system's performance is also contrasted with that of Hovy's, so as to elicit improvements due to proposed modifications, using the same metrics.

Key words: Surface pattern learning, reformulation rules, MRR, precision

INTRODUCTION

The explosive growth of information available electronically has given people potential access to more knowledge than they have had ever before. Massive repositories of information are available at everyone's fingertips (the most prominent example is, of course, the World Wide Web). However, much of these potential remains unrealized due to the lack of an effective information access method. Simply stated, it is difficult to find knowledge that one is looking for. There is an overwhelming amount of information available and existing search services often do little to reduce the information overload. To further compound the problem, textual information exists in different forms and is often varied and unorganized. Question answering systems aim at finding the intended answer to a question given in natural language form[1]. Factual questions are questions whose answers are limited to one or two sentences. Many of the modern Question Answering (QA) systems use surface pattern learning to reduce detailed parsing and other natural language processing techniques[2,3]. However there are still some pitfalls in pure surface pattern learning. Therefore we propose some modifications by using reformulation rules which eliminates the need for maintaining a database i.e. it avoids the use of external knowledge[4]. The proposed system also avoids the problem of the pure pattern learning technique which would miss some special type of questions.

FACTUAL QUESTION ANSWERING

Given a corpus, the problem is to find answers to factual questions. Factual questions have answers limited to one or two sentences. Some of the possible factual questions are given for illustration-

Where is Mount Everest?
Who was the first person to reach the South Pole?
How tall is the Eiffel Tower?
Who was the first American in space?
What is the second tallest mountain in the world?
When did Nixon visit China?
How tall is Mt. Everest?

When the corpus is the World Wide Web, the traditional online Search engines produce a ranked list of potentially relevant documents along with their associated hyper-links which the user must pursue on his own to find an exact answer for his given question (s).

System Overview: Hovy[2] has used surface pattern learning in which, for the same question type (such as BIRTHDATE) already known answers are used, such as-Gandhi 1869, Newton 1642, etc. The problem with this approach is two-fold. In the first place, it requires external knowledge as it must maintain a database for storing the already-known question-answer pairs for the pre-defined question categories. Second, as the

Corresponding Author: S.M. Rafizul Haque, Computer Science and Engineering Discipline, Khulna University, Khulna-9208, Bangladesh

question-answer pairs are hand-crafted, construction of such a database manually is time consuming. Furthermore, for some specific question categories such as DEFINITION questions, finding an exact question-answer pair is extremely difficult. As for example, although we find numerous instances of question-answer pairs for BIRTHDATE questions (i.e., Mozart 1756, Newton 1642, Gandhi 1869), for most DEFITION (e.g., What is the solar system) no such direct question-answer pairs are available. We propose an alternate technique to solve this problem in particular which derives the question-answer pairs automatically thus eliminating the need for maintaining a database. Our proposed system uses an efficient procedure for learning of sentence patterns which are candidates for a given question. First the key terms of the question are extracted. The passages in the corpus are retrieved and ranked according to the frequency of occurrence of the key terms in the particular passage. Then the question is converted to the affirmative form using the reformulation rules. The forms act as a template for generating the question-answer pairs automatically by the machine. The system is comprised of the following components[3]:

(i) Question Analyzer,(ii) Passage Ranker, (iii) Query Generator, (iv) Question-Answer Pair Extractor, (v) Pattern Learner and (vi) Frequency Based Answer Extractor
The overall system architecture (Fig. 1) and each module in particular are described later in details under respective headings.

System architecture

System components

Question analyzer: The role of the question analyzer is to:

(a) Classify a question into a list of pre-defined semantic categories
(b) Extract content words from a question and send them to *passage retriever* to retrieve potentially relevant passages.

To classify a question, the first step is to determine its type[5]. The following *wh*-words are used to determine the question types: who, why, where, whom, what, when, how much, how many, how (rich, long, big, tall, hot, far, fast, large, old, wide, etc.).A list of heuristics will help to map the question types to the pre-defined semantic categories:

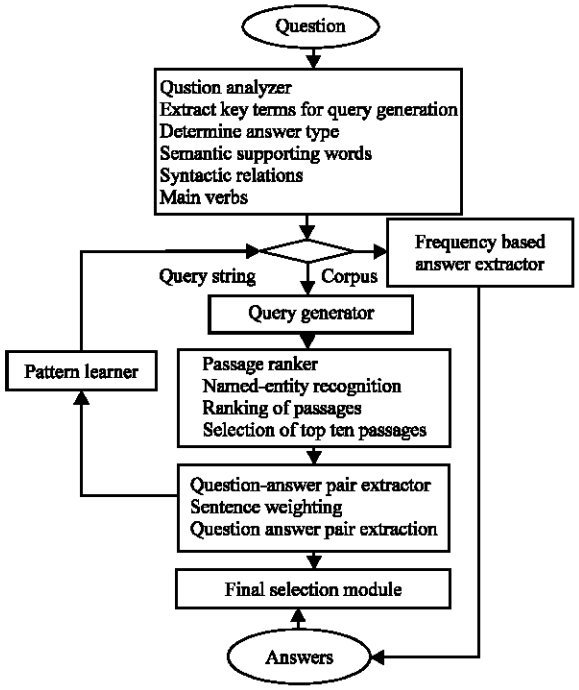

Fig. 1: System architecture

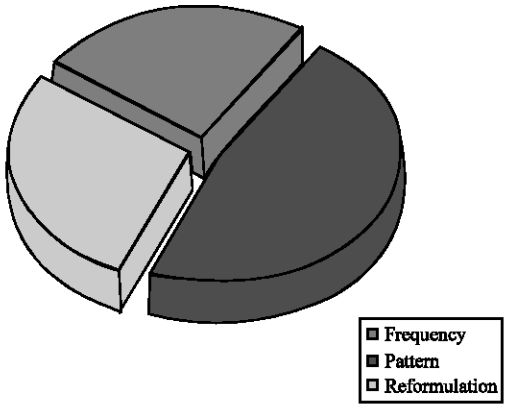

Fig. 2: The relative contribution of the different modules

who is (was) person name => occupation
other who types => personal name
other how much types => number
how hot (cold) => temperature
how fast => speed
how old => age
how long => period of time or length
how big => length or square-measure or cubic-measure
how tall (wide, far) => length

Passage ranker: All the passages in the corpus are searched for the key terms in order to give them a weight that reflects both the possibility that the passage contains

the answer and there is high possibility that the QA system locates the answer within the passage.

The following criteria from Chalendar and Dalmas[6] are considered in ranking passages:

- Occurrence of exact words of the question,
- Variants of question words,
- Mutual closeness of question words,
- Word whose type is the same as the expected answer type
- Some words are not taken into account, i.e., determinants or prepositions, transparent nouns and auxiliary verbs

Query generator: After having generated the ranked list of passages, the highest ranked top ten passages are taken for detailed syntactic processing by the query generator. The use of only the top ten passages instead of the whole corpus is due to the fact that detail syntactic processing of the corpus is time consuming. The role of the query generator is to generate queries to be placed as the query string. To generate the query, we have first recognized the following set of sample questions and their corresponding answer reformulations as follows-

QPattern-what Be the name of the NP?
AnsPatern-the name of the NP Be <NAME>
QPattern-what CNoun Be Modf
AnsPatern-Modf Be <QTarget of type CNoun>
QPattern-what AuxV Sub VP
AnsPatern-Sub PrinV <QTarget>
QPattern-*what* Be NP?
AnsPatern-NP Be <DEFINITION>
QPattern-where Be NP?
AnsPatern-NP Be PP <LOCATION>
Qpattern-where AuxV Sub VP
AnsPatern-Sub VP PP <LOCATION>
QPattern-who Be NP?
AnsPatern-<PERSON/PROFESSION> Be NP
QPattern-who VP NP?
AnsPatern-<PERSON> VP NP
Qpattern-how Adj Be NP?
AnsPatern-NP Be <NUMBER Unit of type Adj>
Qpattern-when AuxV Sub VP?
AnsPatern-Sub VP PP
Qpattern-which Be the NP?
AnsPatern-<QTarget > Be the NP
Qpattern-which CNoun VP NP?
AnsPatern-<QTarget of type CNoun> VP NP

Question-Answer pair extractor: Each candidate sentence provided by the sentence selection module is analyzed using the extraction pattern determined by question analysis.

Extraction patterns are composed of a set of constraint rules on the candidate sentence. Rules are made up of-

- Syntactic patterns that are used to locate potential answer within the sentence,
- Semantic relations that are used to validate answer.

Pattern learner: The pattern-learning algorithm is described with an example. A table of patterns is constructed for each individual question type by the following procedure-

- Select an example for a given question type. Thus for BIRTHYEAR questions we select Mozart 1756 (we refer to Mozart as the question term and 1756 as the answer term).
- Submit the question and the answer term as queries to a search engine. Thus, we give the query+Mozart+1756.
- Retain only those sentences that contain both the question and the answer term.
- Pass each retained sentence through a suffix tree constructor. This finds all substrings, of all lengths, along with their counts.

For example consider the sentences The great composer Mozart (1756-1791) achieved fame at a young age Mozart (1756-1791) was a genius and The whole world would always be indebted to the great music of Mozart (1756-1791). The longest matching substring for all 3 sentences is Mozart (1756-1791), which the suffix tree would extract as one of the outputs along with the score of 3.

- Pass each phrase in the suffix tree through filter to retain only those phrases that contain both the question and the answer term. For the example, we extract only those phrases from the suffix tree that contain the words Mozart and 1756.
- Replace the word for the question term by the tag <NAME> and the word for the answer term by the term <ANSWER>. This procedure is repeated for different examples of the same question type. For BIRTHDATE we also use Gandhi 1869, Newton 1642, etc., For BIRTHDATE, the above steps produce the following output:

born in <ANSWER_TERM>, <NAME>

<NAME> was born on <ANSWER_TERM>,
<NAME> (<ANSWER_TERM>-
<NAME> (<ANSWER_TERM-)
...

These are some of the most common substrings of the extracted sentences that contain both <NAME> and <ANSWER_TERM>.

Calculate the precision of each pattern by the formula $P = C_a / C_o$.
where C_a = total number of patterns with the answer term present.

C_o = total number of patterns present with answer term replaced by any word.

From the set of QA Typology we selected 7 different question types: BIRTHDATE, LOCATION, INVENTOR, DISCOVERER, QUANTITY, PROFESSION, DEFINITION and WHY-FAMOUS. The pattern table for each of these question types was constructed using algorithm above. Some of the patterns obtained are as follows-

BIRTHYEAR

<NAME> (<ANSWER_TERM>-)
<NAME> was born on <ANSWER_TERM>,
<NAME> was born in <ANSWER_TERM>
<NAME> was born <ANSWER_TERM>
<ANSWER_TERM> <NAME> was born
<NAME> (<ANSWER_TERM>
<NAME> (<ANSWER_TERM>-
<NAME> (<ANSWER_TERM>),
born in <ANSWER_TERM>, <NAME>
<NAME> (<ANSWER_TERM>

INVENTOR

<ANSWER_TERM> invents <NAME>
the <NAME> was invented by
<ANSWER_TERM>
<ANSWER_TERM>
<ANSWER_TERM> invented the <NAME> in
<ANSWER_TERM> ' s invention of the
<NAME>
<ANSWER_TERM> invents the <NAME>.
<ANSWER_TERM> ' s <NAME> was
<NAME>, invented by <ANSWER_TERM>
<ANSWER_TERM> ' s <NAME> and
that <ANSWER_TERM> ' s <NAME>
<NAME> was invented by <ANSWER_TERM>,

DISCOVERER

when <ANSWER_TERM> discovered
<NAME>
<ANSWER_TERM>' s discovery of <NAME>
<ANSWER_TERM>, the discoverer of
<NAME>
<ANSWER_TERM> discovers <NAME>.
<ANSWER_TERM> discover <NAME>
<ANSWER_TERM> discovered <NAME>, the
discovery of <NAME> by <ANSWER_TERM>.
<NAME> was discovered by
<ANSWER_TERM>
of <ANSWER_TERM>' s <NAME>
<NAME> was discovered by
<ANSWER_TERM> in

DEFINITION

<NAME> and related <ANSWER_TERM>s
<ANSWER_TERM> (<NAME>,
<ANSWER_TERM>, <NAME>.
, a <NAME> <ANSWER_TERM>,
(<NAME> <ANSWER_TERM>),
form of <ANSWER_TERM>, <NAME>
for <NAME>, <ANSWER_TERM> and
cell <ANSWER_TERM>, <NAME>
and <ANSWER_TERM> > <ANSWER_TERM> >
<NAME>
as <NAME>, <ANSWER_TERM> and

WHY-FAMOUS

<ANSWER_TERM> <NAME> called
laureate <ANSWER_TERM> <NAME>
by the <ANSWER_TERM>, <NAME>,
<NAME>-the <ANSWER_TERM> of
<NAME> was the <ANSWER_TERM> of
by the <ANSWER_TERM> <NAME>,
the famous <ANSWER_TERM> <NAME>,
the famous <ANSWER_TERM> <NAME>
<ANSWER_TERM> > <NAME>
<NAME> is the <ANSWER_TERM> of

LOCATION

<ANSWER_TERM> ' s <NAME>.
regional: <ANSWER_TERM>: <NAME>
to <ANSWER_TERM>' s <NAME>,
<ANSWER_TERM>' s <NAME> in
in <ANSWER_TERM>' s <NAME>,
of <ANSWER_TERM>' s <NAME>,
at the <NAME> in <ANSWER_TERM>
the <NAME> in <ANSWER_TERM>,
from <ANSWER_TERM>' s <NAME>
near <NAME> in <ANSWER_TERM>

Frequency based answer extractor module: When the expected answer type can be known in advance, the most

frequently occurred term in close proximity to the key words is taken as the answer as demonstrated by Soubbotin[7], when no obvious answer strings can be found in the text. For instance, consider the question:

When was Newton born?

Assume the system is unable to find any obvious answer strings, but does find the following sentences containing Newton and born or its variant birth-date, as well as a number:

Newton …….born ……..1642 ……..
Birth-date ……. Newton …….. 1642 ……..
……. gave birth to …….. Newton …….. 1599
born ……. Newton…….1642.

By virtue of the fact that the most frequent number in these sentences is 1642, we can accept it as the most likely answer.

COMPARISON AND DISCUSSION

Two standard metrics are chosen for performance evaluation-

i. MRR: Mean Reciprocal Rank, where rank is the index at which the correct answer is found among the top candidates.
ii. Precision: Precision is the ratio of correct answers and total no of test questions.

Results of the conducted experiments, measured by precision and MRR scores are:

Question type	No. of questions	Precision	MRR
Birthyear	8	0.48	.421
Inventor	6	0.17	.398
Discoverer	4	0.13	.380
Definition	102	0.34	.338
Quantity	3	0.33	.372
Location	16	0.75	.410

The contribution made by different modules are given below-

Module	No. of questions	Precision
Reformulation	35	70%
Pattern Learning	58	72%
Frequency Based	30	50%

The system reduces dependency on external knowledge and natural language processing such as parsing which are time and space consuming. The system eliminates the need for maintaining a data base which is difficult to build and maintain. In pure pattern learning technique, for some question patterns such as those of definition questions, a direct question answer pair was not available, but the proposed system does not suffer from such limitations by learning the question answer pair automatically by using reformulation rules.

CONCLUSION

The work aims primarily at corpus based QA but the approach is also applicable to web based QA. Besides, the principles involved in factual QA provides the foundation for more complex questions such as list type and multiple questions on a single context.

REFERENCES

1. Voorhees, E., 2001. Overview of the question answering track. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 157-165.
2. Ravichandran, D. and H. Eduard, 2002. Learning surface text patterns for a question answering system. In Proc. ACL Conf. Inform. Sci. Institute, University of Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695, USA.
3. Hovy, E.H., U. Hermjakob and D. Ravichandran, 2002. A Question/Answer Typology with Surface Text Patterns. Proceedings of the Human Language Technology (HLT conference. San Diego, CA. Hovy, E.H., U. Hermjakob, C.-Y. Lin and D. Ravichandran (Eds.), 2002b. Using Knowledge to Facilitate Pinpointing of Factoid Answers. Proceedings of the COLING-2002 conference. Taipei, Taiwan.
3. de Chalendar, G., T. Dalmas, F. Elkateb-Gara, O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba and A. Vilnat, 2002. The question answering system QALC at LIMSI: Experiments in Using Web and Wordnet. Proceedings of the Trec-11, Conference-2002. Limsi-Cnrs (France).
4. Hovy, E.H., U. Hermjakob and C.Y. Lin, 2001. The use of external knowledge in factoid QA. Proc. TREC-10 Conf. NIST, Gaithersburg, MD,v, pp: 166-174.
5. Hovy, E., G. Laurie and U. Hermjakob, 0000. Question answering in webclopedia, information sciences institute University of Southern California, 4676 Admiralty Way, Marina del Rey, CA, 90292-6695.
6. De Chalendar, G., T. Dalmas, F. Elkateb-Gara, O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba and A. Vilnat, 2001. The question answering system QALC at LIMSI: Experiments in Using Web and Wordnet, Limsi-Cnrs (France).
7. Soubbotin, M.M. and S.M. Soubbotin, 2001. Patterns of potential answer expressions as clues to the right answer. Proceedings of theTREC-10 Conference. NIST, Gaithersburg, MD, pp: 175-182.