Question Answering
By Pattern Matching, Web-Proofing, Semantic Form Proofing

Min Wu, Xiaoyu Zheng, Michelle Duan, Ting Liu and Tomek Strzalkowski
ILS Institute, Computer Science Department
SUNY Albany

Abstract

In this paper, we introduce the University at Albany's question answering system, ILQUA. It is developed on the following methods: pattern matching over annotated text, web-proofing and semantic form proofing. These methods are currently used in other QA systems, however, we revised them to work together in our QA system.

1. Introduction

This is our first time to participate in QA TREC with three runs and we focus on how to build a QA system in a short time (around half a year) with available methods and get a reasonable performance. Our system evaluation results show that it is possible.

2. Overview

2.1 Architecture

The system components are: question analysis, document retrieval and annotation, pattern matching, web-proofing and semantic form proofing. See Figure 1.

2.2 Question Analysis

There are two main tasks in question analysis: question categorization and query expansion.

We classify questions according to their answer target. The answer targets of TREC questions usually fall into several categories such as person, location, date, quantity, manner, works, organization and so on. These categories also have subcategories. For example, location contains the subcategories nation, city, mountain, lake, river, etc. So our question is categorized according to the main category and subcategory. It is easy to classify "who", "where" and "when" questions. For "what" and "how" questions, we need the help of dictionary tools like WordNet to identify what is asked. This type is usually identified by the noun or adjective following the question word, e.g., "What country…" or "How long …", etc.

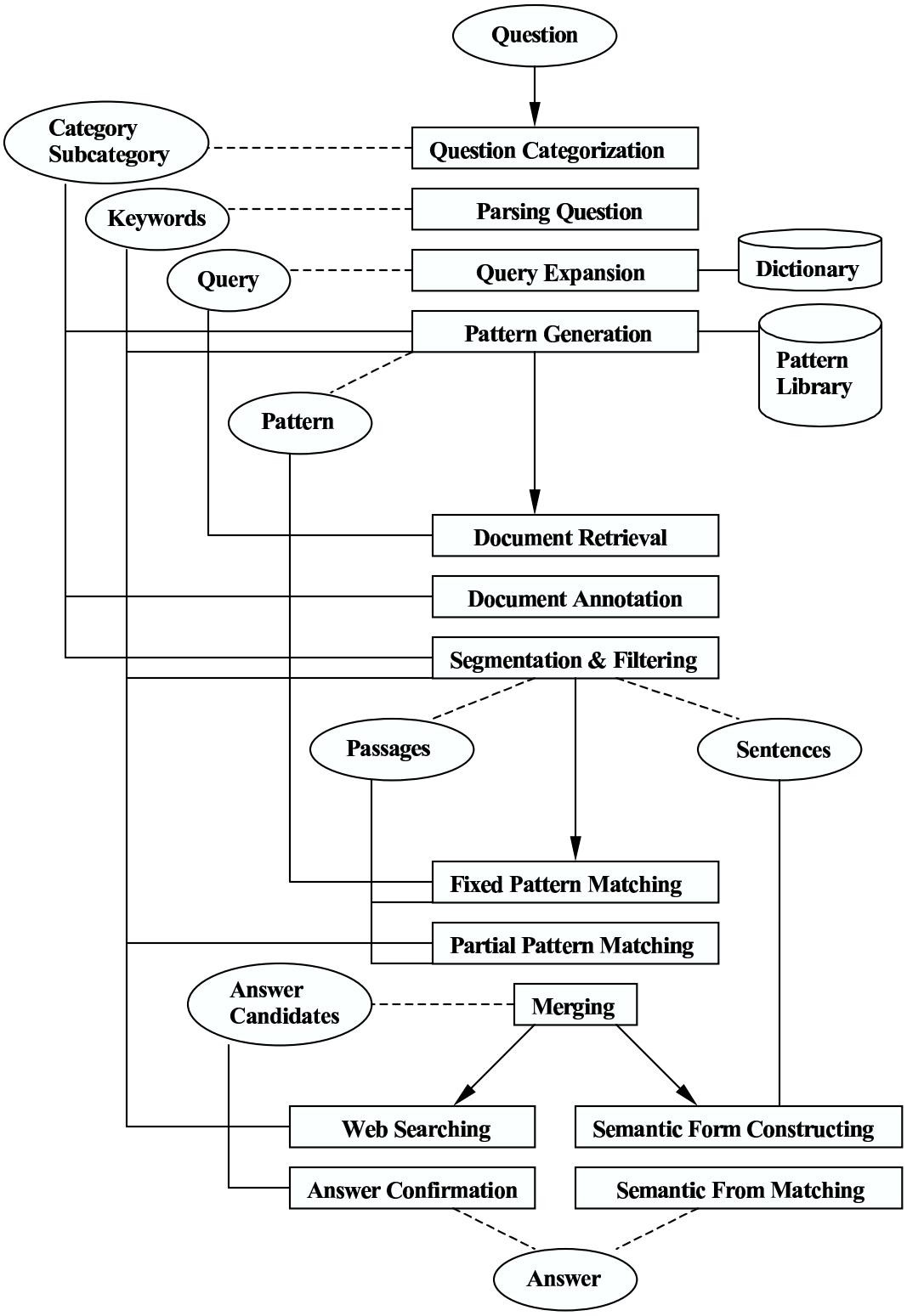

Figure 1. ILQUA System Architecture
(Note: Solid line shows input or tool, dashed line shows output, arrow line shows system processing flow)

Once a question falls into some category, the possible answer patterns can be retrieved from our pattern library. These patterns are general to the category and replacing keywords is necessary to get the specific patterns to the question. However, for some definition questions such as "What is golden parachute", it is difficult to find the answer target. We classify these questions as "No-Pattern" questions.

The first step in the answer finding process is information retrieval (IR), which fetches a number of "relevant" documents from the database. Query modification is usually necessary to increase the recall of IR. It involves both deleting and adding terms to the initial query, which is obtained from the user question through the usual stemming and stopping process. For some questions, we delete some common terms that are not helpful in retrieval. For example, in the question such as "What country is Aswan High Dam located in?", the term "country" is not of much use in retrieving relevant documents. So it is necessary to delete it from our query in order to increase the recall. However, for some nouns and verbs, it may be helpful to get their synonyms and related forms, for example, it is often necessary to find noun and adjective forms for verbs. These keywords and their expansions are kept for future use. WordNet and other dictionaries are used to finish these tasks. Finally, the query is constructed as a Boolean formula that can be processed by the IR component, in our case the UMass' Inquery system.

2.3 Document Retrieval & Annotation

The IR system we use to pre-fetch documents is the Inquery system developed at Univ. of Mass. at Amherst. ILQUA selects the top 50 documents from the file list returned by Inquery and passes those documents to the annotation component.

The annotation tool we use is BBN's Identifinder system. It annotates documents according to the answer target of the question. The annotated entity types include ANIMAL, DISEASE, FAC, GAME, EVENT, GPE, LANGUAGE, LAW, LOCATION, NATIONALITY, PERSON, PLANT, PRODUCT, SUBSTANCE, WORK_OF_ART, CARDINAL, MONEY, ORDINAL, PERCENT, QUANTITY, DATE and TIME. We use most of these types in our system.

The annotated documents are filtered according to the keywords. We cut documents into passages and keep those passages that contain the annotated answer target and keywords.

2.4 Pattern Matching

Our pattern matching component consists of two parts, fixed pattern matching and partial pattern matching. For questions with a simple answer pattern, the answer candidates can be found by fixed pattern matching. As for those with complex answer patterns, we try to locate answer candidates via partial pattern matching.

Fixed pattern matching scans each passage and does pattern matching. Patterns are organized in a list according to their scores. Once a pattern is matched, the answer is

extracted from the text and put into the primary answer list along with the score of the pattern which has been used to extract it. After all of the passages are processed, the system merges what appear to be the same answer in the answer list and calculates a final score for each distinct answer. Figure 2 shows an example of how this fixed pattern matching works. It is very effective for questions such as "When was Adolf Hitler born?", "What does ACLU stand for?", "When was the first camera invented", etc.

Question: *When was MTV started?*
Major Target: Date
Keywords: NP MTV VP started
Query: query: #passage350(MTV,#syn(start,started,starts,starting))
Pattern: ("started" can be replaced by any of it's synonym)
started[ ]+?MTV[ ]+?in[ ]+?<Date>([^<>]+?)<\/Date>
MTV[ ]+?started[ ]+?in[ ]+?<Date>([^<>]+?)<\/Date>
Started[ ]+?MTV[^<>]*?<Date>([^<>]+?)<\/Date
MTV[^<>]*?started[^<>]*?<Date>([^<>]+?)<\/Date>
started[^<>]*?<Date>([^<>]+?)<\/Date>[^<>]*?MTV
<Date>([^<>]+?)<\/Date>[^<>]*?started[^<>]*?MTV
<Date>([^<>]+?)<\/Date>[^<>]*?MTV[^<>]*?started
Matched Passage:
MTV's parent company, Viacom, also owns Nickelodeon and VH1. Nickelodeon is the most popular of the outlets (with MTV second and VH1 third) and celebrates its 20th anniversary on April 1. *MTV started in 1981 and VH1 in 1985.* VH1's older-demographic audience consists of ``graduates of MTV who still love music," said Freston.
Matched Pattern:
MTV[ ]+?started[ ]+?in[ ]+?<Date>([^<>]+?)<\/Date>

Figure 2. Example of Fixed Pattern Matching

The idea of partial pattern matching is based on the assumption that the answer is usually surrounded by keywords and their synonyms. Let E_k ($1 \le k \le$ m) denote the kth named entity in the annotated passage, T_i denotes the ith query keyword ($1 \le i \le$ n), W_i ($0 \le i \le 1$) denotes the weight of T_i and D_i denotes the distance from T_i to E_k. If T_i or its synonyms occur in the passage, D_i is equal to the count of words between T_i and E_k, otherwise, D_i is equal to the maximum average length of the passages *Max*. The matching score of E_k is:

$$S_k = \frac{Max - \left(\sum_{i=1}^{n} D_i \times W_i \right) * \frac{1}{n}}{Max}$$

If the matching score, S_k, is above a threshold, the named entity E_k is extracted and added to the secondary answer list. Finally, the secondary answer list is merged and sorted. Figure3 and Figure4 show a simple example of how partial matching works.

ILQUA selects the top 5 answer candidates from the two answer lists and passes them to the web-proofing. It is important that pattern matching gets the correct answer included in the top 5 ranked answers because the final selection is based on relative likelihood among the 5 candidates.

For definition questions, it is difficult to decide their answer targets and answer patterns in advance. Instead, after the relevant passages are processed, sentences containing a possible answer are passed to the semantic proofing component.

Question: *Who created the literary character Phineas Fogg*
Major Target: Person
Question verbs: create created creates creating
Question verb nouns: creator
Question verb synonyms:
make made makes making produce produced produces producing
Question noun tokens: literary character Phineas Fogg
Answer:
Jules Verne's Phileas *Fogg made literary* history when he traveled ``around the world in 80 days'' in 1873.
Score: 0.5017

Figure 3. Example of Partial Matching

Question: *What band did the music for the 1970's film "Saturday Night Fever"*
Major Target: Organization
Minor Target: Band
Question noun tokens: band music 1970 film Saturday Night Fever
Answer:
First to appear will be ``*Saturday Night Fever*,'' already boasting a $14 million advance and the benefit of a surefire title and score audiences know from the 1977 *film* that featured John Travolta and *music* by the ***Bee Gees***.
Score: 0.5612

Figure 4. Example of Partial Matching

2.5 Web-Proofing

Web-proofing is used to select an answer from the answer candidate list. As the most widely used search engine, Google is chosen to search the Web. The query is submitted to Google and the number of occurrences of each answer candidate is calculated. The assumption here is that the correct answer will have more occurrences in the list returned

by Google than other answer candidates. This simple method works for factoid questions. For list questions, if any of the answer candidates occur in the top documents returned by Google, we will add it to the final answer list. This web-proofing method needs to be improved to deal properly with cases where the correct answer is not among the candidates being proofed or when a slight nuance in the question requires us to find a less popular answer.

2.6 Semantic Form Proofing

In our system, the semantic form proofing is only applied to definition questions. We try to find the answer from the sentence list returned (without a match) by the pattern matching step. The system then builds semantic representation for both the question and the selected sentences. The semantic representation is in the form of an acyclic directed graph where each link is assigned a weight. The system then attempts to match the semantic form of the question to the semantic forms of the candidate sentences. For each matching, it assigns a score. The answer is chosen from the highest scoring sentence.

To build up semantic forms, the head driven parser from M. Collins is adopted. The parser provides the head of phase information for each sentence and semantic form (or say semantic tree) is built up based on this phase head information. Figure 5 shows an example.

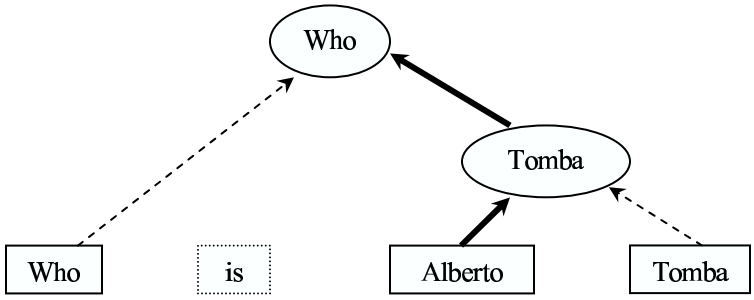

Figure 5. Semantic Form

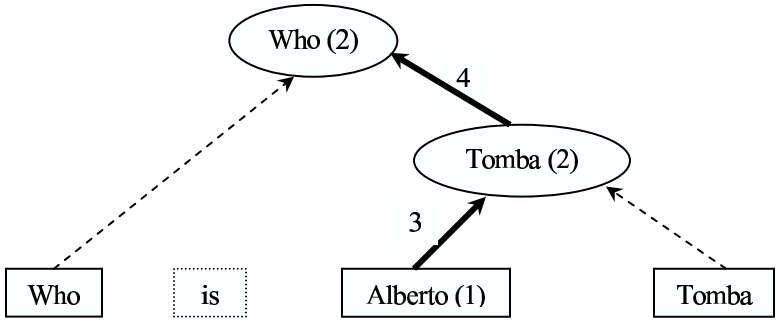

Figure 6. Weighted Semantic Form

The weighted semantic form is built upon semantic form. As we can see from the example that semantic form is an acyclic directed graph and each word is assigned a value to indicate its importance. Since the head of a phase is likely to be more important than other words in the phase, its value is its in-degree plus one (to avoid 0-value links). In Figure 6, the value of "Tomba" is 2. Also a weight is assigned to each link. The value of a link is the sum of the two nodes' weights in that link. In figure 6, the weight of link between "Who" and "Tomba" is 4.

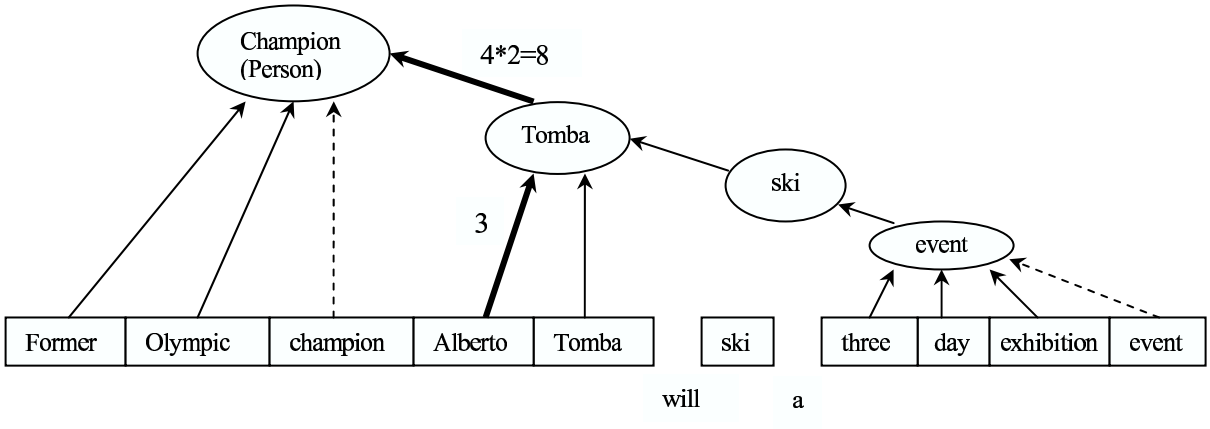

Figure 7. Answer Semantic Form

Similarly, the weighted semantic forms of sentences containing answer candidate are built. However, for some important links, their weight is doubled. In Figure 7, the weight of the link between "Champion" and "Tomba" is doubled. After the assignment of weights to each link, we compare links in sentences containing answer candidate with the links in the question. Once matched links are found, the sum of weights of matched links is the matching score of the answer candidate. In Figure 7, the matching score of word "champion" is 8+3=11. If the score is above some threshold, the answer candidate is extracted and put in the answer list. The answer candidate with the highest matching score will be chosen as the final answer.

3. Experiment

We submitted three runs for evaluation. The evaluation results are shown in Table 1.

We try to assign different weights to keywords in each run. In run AlbanyI2, all of the keywords receive the same weight. In runs AlbanyI3 and AlbanyI4, keywords containing digits and proper nouns receive smaller weights. The evaluation results show that the weighting method is more useful to list questions because the accuracy of list questions in Albany03I4 is better than Albany03I2. However, for factoid questions, the weighting method produces a reduction in performance.

	Albany03I2	Albany03I3	Albany03I4	Median of 54 Runs
Factoid (Accuracy)	0.24	0.206	0.228	0.177
List (Average F)	0.085	0.075	0.096	0.069
Definition (Average F)	0.146	0.133	0.134	0.192

Table 1. Evaluation Result

4. Discussion & Future Work

Since ILQUA has been built in a short period of time as a semester project for a small group of CS students, some of our ideas and implementations are not yet mature. The system needs to be improved in many ways, and we hope this will happen by the next year TREC meeting.

Actually, from the analysis of our experiment results, pattern matching can retrieve the correct answer in the top 5 answer candidates for nearly 45 percent of the factoid questions, and in the top 10 answer candidates for 50 percent of factoid questions. Some techniques need to be applied to increase the rank of correct answers while decreasing the rank of incorrect answers. For partial matching, different weighting methods should be applied to different types of questions.

A more robust and efficient web-proofing method is necessary in our future development, especially a component to deal adequately with junk web content. It is also very difficult for semantic form proofing to handle definition questions whose answer needs to be inferred from the context. Improving ILQUA in this aspect is also in our future plan.

5. Acknowledgements

Thanks to all the other students in our CSI660(Spring 2003) class. Especially to Zhenyu Dai who help us set up Inquery system.

6. Reference

John M. Prager, Jennifer Chu-Carroll, Eric W. Brown, Krzysztof Czuba. Advances in Open Domain Question Answering, Kwwer Academic Publisher, 2004. Question Answering by Predictive Annotation

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, Chin-Yew Lin, 2000. TREC-9 Proceedings. Question Answering in Webclopedia. http://trec.nist.gov/pubs/trec9/t9_proceedings.html

Deepak Ravichandran, Eduard Hovy. ACL 2002. Learning Surface Text Patterns for a Question Answering System.
http://www.isi.edu/natural-language/projects/webclopedia/pubs/02ACL-patterns.pdf

Martin M. Soubbotin, Sergei M. Soubbotin. TREC-10 Proceedings. 2002. Patterns of Potential Answer Expressions as Clues to the Right Answers.http://trec.nist.gov/pubs/trec10/t10_proceedings.html

Sanda Harabagiu, Dan Moldovan. 2000. TREC-9 Proceedings. FALCON: Boosting Knowledge for Answer Engines. http://trec.nist.gov/pubs/trec9/t9_proceedings.html