A Question/Answer Typology with Surface Text Patterns

Eduard Hovy, Ulf Hermjakob, Deepak Ravichandran
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695
(310) 448-8731
{hovy, ulf, ravichan}@isi.edu

ABSTRACT

In this paper we announce the release of ISI's QA Typology, which is being made available on the web to support the rapid construction of new QA systems. The Typology has been augmented with surface-level patterns associated with answer types, allowing systems to locate answers of the desired type in text by simple string matching. These patterns are extracted from the web automatically. We describe the process of their extraction, compression, and accuracy determination.

1. Introduction: QA Types and Typologies

The recent TREC-10 Question Answering competition included over 65 participating systems. The vast majority (if not all) performed some variant of answer typing, in which a process analyzes the input question to determine the desired type of the answer. Answer types are used by systems as a matching criterion to filter out candidate answers that look likely (because, for example, they occur within a window of words that include good question words). For example, "who is the richest person in the world?", "where is Cambodia?", and "when did Marilyn Monroe marry Arthur Miller?" may have answer types *Person*, *Location*, and *Date* respectively.

The same answer type can be intended by various forms of question, as in "what is the name of the person who invented xeroxing?", "who invented xeroxing?", and "who was the inventor of xeroxing?". Similarly, the answer can occur in different forms, such as "Chester F. Carlson invented xeroxing", "the man who invented photocopying was Chester Carlson", and "inventing the xeroxing process was a high point of Chester F. Carlson's life". What we called an answer type is thus a kind of equivalence class of all these phrasings, or a relation that links all the question forms to all their answer forms. We call this equivalence class a *Qtarget*; for this example it might be *Person*. Most QA systems include a list of such Qtargets, typically ranging in number from around 10 to around 50, and starting with *Who, When, Where,* and *What*.

Since many QA systems associate specific matching information (indicative words, surface word patterns, etc.) with their Qtargets, it is useful to create more specific alternatives that narrow the equivalent sets. Thus *Person* might be specialized to *Inventor*, and be associated with words such as "invent", "invention", "discover", "discovery", and "create". Other specializations of *Person* might be *Artist* (with "perform", "sing") and *Author* ("write", "book", "publish").

The hierarchicalized Qtargets form a typology that defines the types of questions the system can handle. The hierarchicalization can be exploited for backoff matches, to allow more general Qtargets to apply in cases where specific ones fail. QA lists or typologies are reported in almost all QA system papers; see for example (Harabagiu et al. 2000, Abney et al., 2000).

2. ISI's QA Typology

Over the past two years, we have created at ISI a QA Typology that currently contains 140 Qtargets. Our initial Typology of about 75 nodes was derived from an analysis by one of our students of over 17,000 questions, downloaded from answers.com (Hovy et al., 2000; 2001); see

http://www.isi.edu/natural-language/projects/webclopedia/

Taxonomy/taxonomy_toplevel.html.

Subsequently, we have been restructuring and extending the Typology to its current form.

Qtargets are of five types. Some are not typical semantic types but are specific to QA. For example, the usual reading of "who was Mother Theresa?" is "why is the individual known as Mother Theresa famous?" (the Qtarget *WhyFamous*, which is not a semantic class). Other Qtargets apply when the system cannot determine a specific semantic type for the answer, but can specify the syntactic type. The Syntactic Qtargets *S-NP* and *S-NOUN* apply to "what does Peugeot company manufacture?", and *S-VP* to "what did John Hinkley do to impress Jodie Foster?". The Role and Slot Qtargets specify constituents or aspects associated with phrases, as in *SLOT-TITLE* for "name a novel written by John Steinbeck" and *ROLE-REASON* for "why was the game cancelled?" in the sentence "the game was cancelled due to bad weather".

The types and uses of Qtargets, including their combination in questions, are available on the Web at http://www.isi.edu/natural-language/projects/webclopedia/qtargets.html

3. Answer Patterns

At the recent TREC conference, several systems emphasized the value of a surface-oriented pattern matching approach to QA. The Insight system from Moscow (Soubbotin and Soubbotin 2001) used some hundreds of surface-level patterns to identify answer strings without (apparently) applying Qtargets or similar reasoning. For example, for *BirthYear* questions such as "which year was Mozart born?" the phrase "Mozart (1756 – 1791)…" provides the answer using the general template

NAME_OF_PERSON (BIRTHYEAR – DEATHYEAR)

Several other systems also defined word-level patterns indicating specific Qtargets; e.g., (Oh et al. 2001). The Microsoft system (Brill et al. 2001) extended the idea of a pattern to its limit, by reformulating the input question as a declarative sentence and then retrieving the sentence verbatim, with its answer as a completion, from the web using the normal search engines. For example, "who was Chester F. Carlson?" was transformed to "Chester F. Carlson was" and submitted. Although this approach yielded many wrong answers (including "Chester F. Carlson was born February 8, 1906, in Seattle"), the sheer number of correct answers often won the day.

Our estimate is that word-level patterns can provide at least 25% of the MRR score defined for TREC (although some systems claimed considerably higher results; see (Soubbotin and Soubbotin 2001) and discussion in (Hermjakob 2002)). In order to determine their power and reap their benefits, we collected all the patterns associated with as many Qtargets as made sense (some Qtargets, such as *Planets* and *Oceans*, are known closed sets that require no patterns).

We developed an automated procedure to learn such patterns from the web, using Altavista (because it returns 1000 documents per query), and to measure their Precision. More formally this experiment can be phrased as "Given a QA pair such as (NAME_OF_PERSON BIRTHYEAR), extract from the web all the different patterns (TEMPLATEs) that contain this QA pair along with the precision of each pattern". We inserted into the Typology the patterns with their Qtargets, recording their Precision scores and relative frequencies of appearance.

The procedure contains two parts:

1. Extracting the patterns
2. Calculating the precision of each pattern

BBN's IdentiFinder named entity tagger (Bikel et al., 1999) was used to remove the variations caused by writing a name or a date in different forms.

Algorithm 1: Extracting patterns

1. An example of the question-answer pair for which the pattern is to be extracted is passed to a search engine. To learn the pattern for the pair (NAME_OF_PERSON BIRTHYEAR) we submit the query "Gandhi 1869" to Altavista.

2. The top 1000 documents returned by the search engine are retrieved.

3. These documents are broken into sentences by a simple sentence breaker.

4. Only sentences that contain both the Question and the Answer term are retained.

5. Each of these sentences is converted into a Suffix tree, to collect counts on all phrases and subphrases present in the document.

6. The phrases obtained from the Suffix tree process are filtered so that only those containing both the Question and the Answer terms are retained. This yields the set of patterns for the given QA pair.

Algorithm 2: Calculating the precision of each pattern

1. The Question term alone (without the Answer term) is given as query to Altavista.

2. As before, the top 1000 documents returned by the search engine for this query are retrieved.

3. Again, the documents are broken into sentences.

4. Only those sentences that contain the Question term are saved.

5. For each pattern obtained in step 6 of Algorithm 1, a pattern-matching check is done against each sentence obtained from step 4 here, and only the sentences containing the Answer are retained. This is used to calculate the precision of each pattern according to the formula

$$\text{Precision} = \frac{\text{\# patterns matching the Answer (step 5)}}{\text{Total \# patterns (step 4)}}$$

6. Only those patterns are retained for which sufficient examples are obtained in step 5.

To increase the size of the data, we apply the algorithms with several different examples of the same Qtarget. Thus in Algorithm 1 for *BirthYear* we used Mozart, Gauss, Gandhi, Nelson Mandela, Michelangelo, Christopher Columbus, and Sean Connery, each with its birth year. We then applied Algorithm 2 with just these names, counting the yields of the patterns on the exact birth years (no additional words or reformulations, which would increase the yield score).

The results were quite good in some cases. For the rather straightforward *BirthYear* patterns are:

Prec.	**#Correct**	**#Found**	**Pattern**
1	122	122	<NAME> (<BD>- <DD>
1	15	15	<NAME> (<BD> - <DD>) ,
1	13	13	, <NAME> (<BD> - <DD>)
0.9166	11	12	<NAME> was born on <BD> in
0.9090	10	11	<NAME> : <BD> - <TIME>
0.6944	25	36	<NAME> was born on <BD>

Note the overlaps among patterns. By not compressing them further we can record different precision levels.

The *Definition* Qtarget posed greater problems:

Diseases (names jaundice*, measles, cancer, and tuberculosis* to pair with the term *disease* but not also with *illness, ailment* etc., which would have increased the counts):

Prec.	**#Correct**	**#Found**	**Pattern**
1	46	46	heart <TERM>, <NAME>
1	35	35	<NAME> & tropical <TERM> weekly
1	30	30	venereal <TERM>, <NAME>
1	26	26	<NAME>, a <TERM> that
1	24	24	lyme <TERM>, <NAME>
1	22	22	, heart <TERM>, <NAME>
1	21	21	's <TERM>, <NAME>
0.9565	22	23	lyme <TERM> <NAME>
0.9	9	10	s <TERM>, <NAME> and
0.8815	67	76	<NAME> , a <TERM>
0.8666	13	15	<TERM> , especially <NAME>

Metal (*Gold*, *silver*, *platinum*, and *bronze* to pair with *metal*):

1	13	13	of <NAME> <TERM> .
1	12	12	the <TERM> <NAME> .
0.9	9	10	<TERM> : <NAME>.
0.875	14	16	the <NAME> <TERM> .
0.8181	27	33	<TERM> (<NAME>)
0.8	8	10	s <TERM> (<NAME>
0.7575	25	33	<TERM> , <NAME> , a
0.75	177	236	the <NAME> <TERM>
0.75	12	16	<TERM> : <NAME> ,
0.7427	179	241	<NAME> <TERM> ,
0.7391	17	23	<TERM> (<NAME> ,
0.7	7	10	<TERM> - <NAME> ,

The similar patterns for Disease and Metal definitions indicate that one should not create specialized Qtargets *Definition-Disease* and *Definition-Metal*.

References

Abney, S., M. Collins, and A. Singhal. 2000. Answer Extraction. *Proceedings of the Applied Natural Language Processing Conference (ANLP)*. Seattle, WA (296–301).

Bikel, D., R. Schwartz, and R. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning—Special Issue on NL Learning*, 34, 1–3.

Brill, E., J. Lin, M. Banko, S. Dumais, and A. Ng. 2001. Data-Intensive Question Answering. . *Proceedings of the TREC-10 Conference*. NIST, Gaithersburg, MD (183–189).

Harabagiu, S., M. Pasca, and S. Maiorano. 2000. Experiments with Open Domain Textual Question Answering. *Proceedings of the 18th COLING Conference.* Saarbrücken, Germany (292–298).

Hermjakob, U. 2001. Parsing and Question Classification for Question Answering. In *Proceedings of the Workshop on Question Answering at the Conference ACL-2001*. Toulouse, France.

Hermjakob, U. 2002. In prep.

Hovy, E.H., L. Gerber, U. Hermjakob, M. Junk, and C.-Y. Lin. 2000. Question Answering in Webclopedia. *Proceedings of the TREC-9 Conference*. NIST. Gaithersburg, MD.

Hovy, E.H., U. Hermjakob, and C.-Y. Lin. 2001. The Use of External Knowledge in Factoid QA. *Proceedings of the TREC-10 Conference*. NIST, Gaithersburg, MD (166–174).

Oh, JH., KS. Lee, DS. Chang, CW. Seo, KS. Choi. 2001. TREC-10 Experiments at KAIST: Batch Filtering and Question Answering. *Proceedings of the TREC-10 Conference*. NIST, Gaithersburg, MD (354–361).

Soubbotin, M.M. and S.M. Soubbotin. 2001. Patterns of Potential Answer Expressions as Clues to the Right Answer. *Proceedings of the TREC-10 Conference*. NIST, Gaithersburg, MD. (175–182)