Parsing and Question Classification for Question Answering

Ulf Hermjakob
Information Sciences Institute
University of Southern California
ulf@isi.edu

Abstract

This paper describes machine learning based parsing and question classification for question answering. We demonstrate that for this type of application, parse trees have to be semantically richer and structurally more oriented towards semantics than what most treebanks offer. We empirically show how question parsing dramatically improves when augmenting a semantically enriched Penn treebank training corpus with an additional question treebank.

1 Introduction

There has recently been a strong increase in the research of question answering, which identifies and extracts answers from a large collection of text. Unlike information retrieval systems, which return whole documents or larger sections thereof, question answering systems are designed to deliver much more focused answers, e.g.

Q: Where is Ayer's Rock?
A: in central Australia

Q: Who was Gennady Lyachin?
A: captain of the Russian nuclear submarine Kursk

The August 2000 TREC-9 short form Q&A track evaluations, for example, specifically limited answers to 50 bytes.

The Webclopedia project at the USC Information Sciences Institute (Hovy 2000, 2001) pursues a semantics-based approach to answer pinpointing that relies heavily on parsing. Parsing covers both questions as well as numerous answer sentence candidates. After parsing, exact answers are extracted by matching the parse trees of answer sentence candidates against that of the parsed question. This paper describes the critical challenges that a parser faces in Q&A applications and reports on a number of extensions of a deterministic machine-learning based shift-reduce parser, CONTEX (Hermjakob 1997, 2000), which was previously developed for machine translation applications. In particular, section 2 describes how additional treebanking vastly improved parsing accuracy for questions; section 3 describes how the parse tree is extended to include the answer type of a question, a most critical task in question answering; section 4 presents experimental results for question parsing and QA typing; and finally, section 5 describes how the parse trees of potential answer sentences are enhanced semantically for better question-answer matching.

2 Question Treebank

In question answering, it is particularly important to achieve a high accuracy in parsing the questions. There are often several text passages that contain an answer, so if the parser does not produce a sufficiently good parse tree for some of the answer sentences, there's still a good chance that the question can be answered correctly based on other sentences containing the answer. However, when the question is analyzed incorrectly, overall failure is much more likely.

A scenario with a question in multiple variations, as cleverly exploited by the SMU team (Harabagiu, 2000) in TREC9 for maybe about 10% of the 500 original questions, is probably more of an anomaly and can't be assumed to be typical.

Parsing accuracy of trained parsers is known to depend significantly on stylistic similarities between training corpus and application text. In the Penn Treebank, only about half a percent of all sentences from the Wall Street Journal are (full) questions. Many of these are rhetorical, such as *"So what's the catch?"* or *"But what about all those non-duck ducks flapping over Washington?"*. Many types of questions that are common in question answering are however severely underrepresented. For example, there are no questions beginning with the interrogatives *When* or *How much* and there are no para-interrogative imperative

sentences starting with *"Name"*, as in *Name a Gaelic language.*

This finding is of course not really surprising, since newspaper articles focus on reporting and are therefore predominantly declarative. Therefore, we have to expect a lower accuracy for parsing questions than for parsing declarative sentences, if the parser was trained on the Penn treebank only. This was confirmed by preliminary question parsing accuracy tests using a parser trained exclusively on sentences from the Wall Street Journal. Question parsing accuracy rates were significantly lower than for regular newspaper sentences, even though one might have expected them to be higher, given that questions, on average, tend to be only half as long as newspaper sentences.

To remedy this shortcoming, we treebanked additional questions as we would expect them in question answering. At this point, we have treebanked a total of 1153 questions, including

- all 38 prep questions for TREC 8,
- all 200 questions from TREC 8,
- all 693 questions from TREC 9,
- plus 222 questions from a travel guide phrase book and online resources, including *answers.com*.

The online questions cover a wider cross-section of style, including yes-no questions (of which there was only one in the TREC questions set), true-false questions (none in TREC), and questions with wh-determiner phrases[1] (none in TREC). The additionally treebanked questions therefore complement the TREC questions.

The questions were treebanked using the deterministic shift-reduce parser CONTEX. Stepping through a question, the (human) treebanker just hits the return key if the proposed parse action is correct, and types in the correct action otherwise. Given that the parser predicts over 90% of all individual steps correctly, this process is quite fast, most often significantly less than a minute per question, after the parser was trained using the first one hundred treebanked questions.

The treebanking process includes a "sanity check" after the treebanking proper of a sentence. The sanity check searches the treebanked parse tree for constituents with an uncommon sub-constituent structure and flags them for human inspection. This helps to eliminate most human errors. Here is an example of a (slightly simplified) question parse tree. See section 5 for a discussion of how the trees differ from the Penn Treebank II standard.

[1] How much does one ton of cement cost?
  [SNT,PRES,Qtarget: MONETARY-QUANTITY]
  (QUANT) [2] How much [INTERR-ADV]
    (MOD) [3] How [INTERR-ADV]
    (PRED) [4] much [ADV]
  (SUBJ LOG-SUBJ) [5] one ton of cement [NP]
    (QUANT) [6] one ton [NP,MASS-Q]
      (PRED) [7] one ton [NP-N,MASS-Q]
        (QUANT) [8] one [CARDINAL]
        (PRED) [9] ton [COUNT-NOUN]
    (PRED) [10] of cement [PP]
      (P) [11] of [PREP]
      (PRED) [12] cement [NP]
        (PRED) [13] cement [NOUN]
  (PRED) [14] does cost [VERB,PRES]
    (AUX) [15] does [AUX]
    (PRED) [16] cost [VERB]
  (DUMMY) [17] ? [QUESTION-MARK]

Figure 1: a simplified sample parse tree

3 QA Typing ("Qtargets")

Previous research on question answering, e.g. Srihari and Li (2000), has shown that it is important to classify questions with respect to their answer types. For example, given the question "How tall is Mt. Everest?", it is very useful to identify the answer type as a distance quantity, which allows us to narrow our answer search space considerably. We refer to such answer types as *Qtargets*.

To build a very detailed question taxonomy, Gerber (2001) has categorized 18,000 online questions with respect to their answer type. From this we derived a set of currently 115 elementary Qtargets, such as distance quantity. For some questions, like *"Who is the owner of CNN?"*, the answer might be one of two or more distinct types of elementary Qtargets, such as proper-person or proper-organization for the ownership question. Including such combinations, the number of distinct Qtargets rises to 122.

Here are some more examples:

- **Q1:** How long would it take to get to Mars?
  Qtarget: temporal-quantity
- **Q2:** When did Ferraro run for vice president?
  Qtarget: date, temp-loc-with-year; =temp-loc
- **Q3:** Who made the first airplane?
  Qtarget: proper-person, proper-company; =proper-organization
- **Q4:** Who was George Washington?
  Qtarget: why-famous-person
- **Q5:** Name the second tallest peak in Europe.
  Qtarget: proper-mountain

Question 1 (Q1) illustrates that it is not sufficient to analyze the wh-group of a sentence, since "how

[1] "What country's national anthem does the movie Casablanca close to the strains of?"

long" can also be used for questions targeting a distance-quantity. Question 2 has a complex Qtarget, giving first preference to a date or a temporal location with a year and second preference to a general temporal location, such as "six years after she was first elected to the House of Representatives". The equal sign (=) indicates that sub-concepts of *temp-loc* such as *time* should be excluded from consideration at that preference level. Question 3 & 4 both are *who*-questions, however with very different Qtargets. *Abstract* Qtargets such as the *why-famous-person* of question 4, can have a wide range of answer types, for example a prominent position or occupation, or the fact that they invented or discovered something. Abstract Qtargets have one or more *arguments* that completely describe the question: "*Who was George Washington?*", "*What was George Washington best known for?*", and "*What made George Washington famous?*" all map to **Qtarget** *why-famous-person*, **Qargs** ("George Washington"). Below is a listing of all currently used abstract Qtargets:

Abstract Qtargets

- why-famous (*What is Switzerland known for?* - 3 occurrences in TREC 8&9)
  - why-famous-person (*Who was Lacan?* - 35)
- abbreviation-expansion (*What does NAFTA stand for?* - 16)
- abbreviation (*How do you abbreviate limited partnership?* - 5)
- definition (*What is NAFTA?* - 35)
- synonym (*Aspartame is also known as what?* - 6)
- contrast (*What's the difference between DARPA and NSF?* - 0)

The ten most common **semantic Qtargets** in the TREC8&9 evaluations were

- proper-person (98 questions)
- at-location/proper-place (68)
- proper-person/proper-organization (68)
- date/temp-loc-with-year/date-range/temp-loc (66)
- numerical-quantity (51)
- city (39)
- (other) named entity (20)
- temporal quantity (15)
- distance quantity (14)
- monetary quantity (12)

Some of the Qtargets occurring only once were proper-American-football-sports-team, proper-planet, power-quantity, proper-ocean, season, color, phone-number, proper-hotel and government-agency.

The following Qtarget examples show the hierarchical structure of Qtargets:

Quantity

- energy-quantity (1)
- mass-quantity (6)
- monetary-quantity (12)
- numerical-quantity (51)
- power-quantity (1)
- spatial-quantity
  - distance-quantity (14)
  - area-quantity (3)
  - volume-quantity (0)
- speed-quantity (2)
- temperature-quantity (2)
- temporal-quantity (15)

Besides the abstract and semantic (ontology-based) Qtargets, there are two further types.

1. **Qtargets referring to semantic role**
   Q: Why can't ostriches fly?
   Qtarget: (ROLE REASON)
   This type of Qtarget recommends constituents that have a particular semantic role with respect to their parent constituent.

2. **Qtargets referring to marked-up constituents**
   Q: Name a film in which Jude Law acted.
   Qtarget: (SLOT TITLE-P TRUE)
   This type of Qtarget recommends constituents with slots that the parser can mark up. For example, the parser marks constituents that are quoted and consist of mostly and markedly capitalized content words as potential titles.

The 122 Qtargets are computed based on a list of 276 hand-written rules.[2] One reason why there are relatively few rules per Qtarget is that, given a semantic parse tree, the rules can be formulated at a high level of abstraction. For example, parse trees offer an abstraction from surface word order and CONTEX's semantic ontology, which has super-concepts such as *monetarily-quantifiable-abstract* and sub-concepts such as *income*, *surplus* and *tax*, allows to keep many tests relatively simple and general.

For 10% of the TREC 8&9 evaluation questions, there is no proper Qtarget in our current Qtarget hierarchy. Some of those questions could be covered by further enlarging and refining the Qtarget hierarchy, while others are hard to capture with a semantic super-category that would narrow the search space in a meaningful way:

- What does the Peugeot company manufacture?
- What do you call a group of geese?
- What is the English meaning of caliente?

[2]These numbers for Qtargets and rules are up by a factor of about 2 from the time of the TREC9 evaluation.

# of Penn sentences	# of add. Q. sentences	Labeled Precision	Labeled Recall	Tagging Accuracy	Cr. Brackets per sent.	Qtarget acc. (strict)	Qtarget acc. (lenient)
2000	0	83.47%	82.49%	94.65%	0.34	63.0%	65.5%
3000	0	84.74%	84.16%	94.51%	0.35	65.3%	67.4%
2000	38	91.20%	89.37%	97.63%	0.26	85.9%	87.2%
3000	38	91.52%	90.09%	97.29%	0.26	86.4%	87.8%
2000	238	94.16%	93.39%	98.46%	0.21	91.9%	93.1%
2000	975	95.71%	95.45%	98.83%	0.17	96.1%	97.3%

Table 1: Parse tree accuracies for varying amounts and types of training data.
Total number of test questions per experiment: 1153

4 Experiments

In the first two test runs, the system was trained on 2000 and 3000 Wall Street Journal sentences (enriched Penn Treebank). In runs three and four, we trained the parser with the same Wall Street Journal sentences, augmented by the 38 treebanked pre-TREC8 questions. For the fifth run, we further added the 200 TREC8 questions as training sentences when testing TREC9 questions, and the first 200 TREC9 questions as training sentences when testing TREC8 questions.

For the final run, we divided the 893 TREC-8 and TREC-9 questions into 5 test subsets of about 179 for a five-fold cross validation experiment, in which the system was trained on 2000 WSJ sentences plus about 975 questions (all 1153 questions minus the approximately 179 test sentences held back for testing). In each of the 5 subtests, the system was then evaluated on the test sentences that were held back, yielding a total of 893 test question sentences.

The Wall Street Journal sentences contain a few questions, often from quotes, but not enough and not representative enough to result in an acceptable level of question parsing accuracy. While questions are typically shorter than newspaper sentences (making parsing easier), the word order is often markedly different, and constructions like preposition stranding ("What university was Woodrow Wilson President of?") are much more common. The results in figure 1 show how crucial it is to include additional questions when training a parser, particularly with respect to Qtarget accuracy.[3] With an additional 1153 treebanked questions as training input, parsing accuracy levels improve considerably for questions.

5 Answer Candidate Parsing

A thorough question analysis is however only one part of question answering. In order to do meaningful matching of questions and answer candidates, the analysis of the answer candidate must reflect the depth of analysis of the question.

5.1 Semantic Parse Tree Enhancements

This means, for example, that when the question analyzer finds that the question "How long does it take to fly from Washington to Hongkong?" looks for a temporal quantity as a target, the answer candidate analysis should identify any temporal quantities as such. Similarly, when the question targets the name of an airline, such as in "Which airlines offer flights from Washington to Hongkong?", it helps to have the parser identify proper airlines as such in an answer candidate sentence.

For this we use an in-house preprocessor to identify constituents like the 13 types of quantities in section 3 and for the various types of temporal locations. Our named entity tagger uses BBN's IdentiFinder(TM) (Kubala, 1998; Bikel, 1999), augmented by a named entity refinement module. For named entities (NEs), IdentiFinder provides three types of classes, *location*, *organization* and *person*. For better matching to our question categories, we need a finer granularity for location and organization in particular.

- **Location** $\rightarrow$ proper-city, proper-country, proper-mountain, proper-island, proper-star-constellation, ...
- **Organization** $\rightarrow$ government-agency, proper-company, proper-airline, proper-university, proper-sports-team, proper-american-football-sports-team, ...

For this refinement, we use heuristics that rely both on lexical clues, which for example works quite well for colleges, which often use "College" or "University" as their lexical heads, and lists of proper entities, which works particularly well for more limited classes of named entities like countries and government agencies. For many classes like mountains, lexical clues ("Mount Whitney", "Humphreys Peak", "Sassafras Mountain") and lists of well-known entities ("Kilimanjaro", "Fujiyama", "Matterhorn") complement each other well. When no heuristic or back-

[3] At the time of the TREC9 evaluation in August 2000, only about 200 questions had been treebanked, including about half of the TREC8 questions (and obviously none of the TREC9 questions).

ground knowledge applies, the entity keeps its coarse level designation ("location").

For other Qtargets, such as "Which animals are the most common pets?", we rely on the SENSUS ontology[4] (Knight and Luk, 1994), which for example includes a hierarchy of animals. The ontology allows us to conclude that the "dog" in an answer sentence candidate matches the Qtarget *animal* (while "pizza" doesn't).

5.2 Semantically Motivated Trees

The syntactic and semantic structure of a sentence often differ. When parsing sentences into parse trees or building treebanks, we therefore have to decide whether to represent a sentence primarily in terms of its syntactic structure, its semantic structure, something in between, or even both.

We believe that an important criterion for this decision is what application the parse trees might be used for. As the following example illustrates, a semantic representation is much more suitable for question answering, where questions and answer candidates have to be matched. What counts in question answering is that question and answer match semantically. In previous research, we found that the semantic representation is also more suitable for machine translation applications, where syntactic properties of a sentence are often very language specific and therefore don't map well to another language.

Parse trees [1] and [12] are examples of our system's structure, whereas [18] and [30] represent the same question/answer pair in the more syntactically oriented structure of the Penn treebank[5] (Marcus 1993).

Question and answer in CONTEX format:

[1] When was the Berlin Wall opened?
  [SNT,PAST,PASSIVE,WH-QUESTION,
  Qtarget: DATE-WITH-YEAR,DATE,
  TEMP-LOC-WITH-YEAR,TEMP-LOC]
  (TIME) [2] When [INTERR-ADV]
  (SUBJ LOG-OBJ) [3] the Berlin Wall [NP]
    (DET) [4] the [DEF-ART]
    (PRED) [5] Berlin Wall [PROPER-NAME]
      (MOD) [6] Berlin [PROPER-NAME]
      (PRED) [7] Wall [COUNT-NOUN]
  (PRED) [8] was opened [VERB,PAST,PASSIVE]
    (AUX) [9] was [VERB]
    (PRED) [10] opened [VERB]
  (DUMMY) [11] ? [QUESTION-MARK]

[12] On November 11, 1989, East Germany
  opened the Berlin Wall. [SNT,PAST]
  (TIME) [13] On November 11, 1989,
  [PP,DATE-WITH-YEAR]
  (SUBJ LOG-SUBJ) [14] East Germany
  [NP,PROPER-COUNTRY]
  (PRED) [15] opened [VERB,PAST]
  (OBJ LOG-OBJ) [16] the Berlin Wall [NP]
  (DUMMY) [17] . [PERIOD]

Same question and answer in PENN TREEBANK format:

[18] When was the Berlin Wall opened? [SBARQ]
  [19] When [WHADVP-1]
  [20] was the Berlin Wall opened [SQ]
    [21] was [VBD]
    [22] the Berlin Wall [NP-SBJ-2]
    [23] opened [VP]
      [24] opened [VBN]
      [25] -NONE- [NP]
        [26] -NONE- [*-2]
      [27] -NONE- [ADVP-TMP]
        [28] -NONE- [*T*-1]
  [29] ? [.]

[30] On November 11, 1989, East Germany
  opened the Berlin Wall. [S]
  [31] On November 11, 1989, [PP-TMP]
  [32] East Germany [NP-SBJ]
  [33] opened the Berlin Wall [VP]
    [34] opened [VBD]
    [35] the Berlin Wall [NP]
  [36] . [.]

The "semantic" trees ([1] and [12]) have explicit roles for all constituents, a flatter structure at the sentence level, use traces more sparingly, separate syntactic categories from information such as tense, and group semantically related words, even if they are non-contiguous at the surface level (e.g. verb complex [8]). In trees [1] and [12], semantic roles match at the top level, whereas in [18] and [30], the semantic roles are distributed over several layers.

Another example for differences between syntactic and semantic structures are the choice of the head in a prepositional phrase (PP). For all PPs, such as *on Nov. 11, 1989*, *capital of Albania* and [composed] *by Chopin*, we always choose the noun phrase as the head, while syntactically, it is clearly the preposition that heads a PP.

We restructured and enriched the Penn treebank into such a more semantically oriented representation, and also treebanked the 1153 additional questions in this format.

[4] SENSUS was developed at ISI and is an extension and rearrangement of WordNet.

[5] All trees are partially simplified; however, a little bit more detail is given for tree [1]. UPenn is in the process of developing a new treebank format, which is more semantically oriented than their old one, and is closer to the CONTEX format described here.

6 Conclusion

We showed that question parsing dramatically improves when complementing the Penn treebank training corpus with an additional treebank of 1153 questions. We described the different answer types ("Qtargets") that questions are classified as and presented how we semantically enriched parse trees to facilitate question-answer matching.

Even though we started our Webclopedia project only five months before the TREC9 evaluation, our Q&A system received an overall Mean Reciprocal Rank of 0.318, which put Webclopedia in essentially tied second place with two others. (The best system far outperformed those in second place.) During the TREC9 evaluation, our deterministic (and therefore time-linear) CONTEX parser robustly parsed approximately 250,000 sentences, successfully producing a full parse tree for each one of them.

Since then we scaled up question treebank from 250 to 1153; roughly doubled the number of Qtarget types and rules; added more features to the machine-learning based parser; did some more treebank cleaning; and added more background knowledge to our ontology.

In the future, we plan to refine the Qtarget hierarchy even further and hope to acquire Qtarget rules through learning.

We plan to make the question treebank publicly available.

References

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

Laurie Gerber. 2001. A QA Typology for Webclopedia. In prep.

Sanda Harabagiu, Marius Pasca and Steven Maiorano 2000. Experiments with Open-Domain Textual Question Answering In *Proceedings of COLING-2000*, Saarbrücken.

Ulf Hermjakob and R. J. Mooney. 1997. Learning Parse and Translation Decisions From Examples With Rich Context. In *35th Proceedings of the ACL*, pages 482-489.
file://ftp.cs.utexas.edu/pub/mooney/papers/con texacl-97.ps.gz

Ulf Hermjakob. 2000. Rapid Parser Development: A Machine Learning Approach for Korean. In *Proceedings of the North American chapter of the Association for Computational Linguis tics (NA-ACL-2000)*
http://www.isi.edu/~ulf/papers/kor_naacl00.ps.gz

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

Ed Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, D. Ravichandran 2001. Towards Semantics-Based Answer Pinpointing In *Proceedings of the HLT 2001 Conference*, San Diego

K. Knight, S. Luc, et al. 1994. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the American Association of Artificial Intelligence* AAAI-94. Seattle, WA.

Francis Kubala, Richard Schwartz, Rebecca Stone, Ralph Weischedel (BBN). 1998. Named Entity Extraction from Speech. In *1998 DARPA Broadcast News Transcription and Understanding Workshop*
http://www.nist.gov/speech/publications/darpa 98/html/lm50/lm50.htm

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics 19(2), pages 313–330.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track evaluation. In *E. M. Voorhees and D. K. Harman, editors, Proceedings of the Eighth Text REtrieval Conference* (TREC-8). http://trec.nist.gov/pubs.html

R. Srihari, C. Niu, and W. Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of the conference on Applied Natural Language Processing* (ANLP 2000), Seattle.