A Semi-Automatic, Iterative Method for Creating a Domain-Specific Treebank

Corina Dima **Erhard Hinrichs**
Department of Linguistics
University of Tübingen
corina.dima, erhard.hinrichs@uni-tuebingen.de

Abstract

In this paper we present the development process of NLP-QT, a question treebank that will be used for data-driven parsing in the context of a domain-specific QA system for querying NLP resource metadata. We motivate the need to build NLP-QT as a resource in its own right, by comparing the Penn Treebank-style annotation scheme used for QuestionBank (Judge et al., 2006) with the modified NP annotation for the Penn Treebank introduced by Vadas and Curran (2007). We argue that this modified annotation scheme provides a better interface representation for semantic interpretation and show how it can be incorporated into the NLP-QT resource, without significant loss in parser performance.

The parsing experiments reported in the paper confirm the feasibility of an iterative, semi-automatic construction of the NLP-QT resource similar to the approach taken for QuestionBank. At the same time, we propose to improve the iterative refinement technique used for QuestionBank by adopting Hwa (2001)'s heuristics for selecting additional material to be hand-corrected and added to the data set at each iteration.

1 Introduction

Question-Answering (QA) systems have a long history in the field of natural language processing. In the 1970s and 1980s QA systems focused on natural language interfaces to domain-specific data bases or expert systems. Such systems typically used hand-crafted, rule-based front ends for parsing and semantic interpretation. With the increased availability of large-scale textual resources, QA systems more recently have focused on domain-independent broad-coverage information retrieval applications that typically employ more shallow processing techniques for question analysis and answer matching.

The intended application for the research reported in the present paper is more in the tradition of the earlier, domain-specific QA systems in that it aims to provide a natural language front-end to large repositories of metadata about language tools and resources that are made available by the CLARIN[1] project. However, instead of relying on a parser with hand-crafted grammar rules, it employs a robust data-driven parser that requires annotated training data in the form of a treebank.

Since the natural language front end for the intended QA system is English, the simplest solution would be to use a statistical parser such as the Berkeley (Petrov and Klein, 2007) or Stanford (Klein and Manning, 2003) parser with an existing language model obtained from the Penn Treebank (Marcus et al., 1993). However, it is well known that parser performance drops when analyzing text from domains other than that represented in the training data (Sekine, 1997; Gildea, 2001). In particular, Judge et al. (2006) have shown that language models obtained from the Penn Treebank perform far worse on questions than on their original test data. The Bikel (2004) parser they employ has an F-Score of 82.97 when tested on Section 23 of the Penn-II Treebank and an F-Score of 78.77 when tested on the 4000 questions in QuestionBank. Judge et al. (2006) attribute this loss of per-

[1] CLARIN project - http://www.clarin.eu

Proceedings of Recent Advances in Natural Language Processing, pages 413–419,
Hissar, Bulgaria, 12-14 September 2011.

formance to two factors: (i) in the genre of newspaper texts, which the Penn Treebank is based on, questions are not a high frequency syntactic construction, and (ii) if wh-type constructions occur at all in the Penn Treebank, they predominantly involve relative clause constructions or indirect questions, but not unembedded questions. Therefore, a parser trained on Penn Treebank data, routinely misanalyses unembedded questions as these other two construction types. In fact, it was this poor parser performance that led Judge et al. to create QuestionBank, a special-purpose treebank based on SemEval data sets for Question Answering (QA).The data include the SemEval QA data from 1999-2001, part of the 2003 set (2000 questions), and another 2000 questions provided by the Cognitive Computation Group at the University of Illinois, which were also test data for developing QA systems. Training a statistical parser on QuestionBank data, possibly in combination with Penn Treebank data, therefore seems to be an attractive alternative. In fact, this is precisely how Judge et al. train their parser. However, for reasons explained in more detail in sections 2 and 3, we will adopt annotation guidelines for questions that differ from the Penn Treebank-style annotation used in QuestionBank. Rather, we will follow a more hierarchical annotation style for NPs that has been proposed by Vadas and Curran (2007) and that provides an easier interface for semantic interpretation. Section 3 will introduce the Vadas and Curran (2007) annotation style and will motivate why it is appropriate for the QA system envisaged here. Section 4 will present a set of parsing experiments for the Berkeley parser trained on different combinations of treebank data discussed in sections 2 and 3. The final section summarizes the main results of this paper and discusses directions for future research.

2 Data Collection for Querying NLP Resource Metadata

One of the main reasons to create a new data set of questions and not use some already existing set has to do with the specific subject domain of the QA system to be developed. All the questions should concern particular pieces of information associated with language resources or with different application domains of natural language processing. In order to obtain a realistic data set of this sort, we harvested the questions from mailing lists like LinguistList[2] and Corpora List[3], as well as from the Stack Overflow[4] questions tagged with ”nlp”.

The mailing lists have a history of 20 years and have a lot of extra content other than user queries. Therefore, all the posts had to be browsed through in order to manually extract only the relevant questions from the whole post. For example, information about the person asking the question was deleted from the original posts, since such information is not relevant for a QA system. Spelling and grammar errors were then removed from the extracted questions. A number of 2500 questions were harvested until the moment of writing, but the goal is to gather a 10.000 questions corpus that should provide enough training and testing data when converted into a treebank.

The data below provide some typical examples that have been collected from the three sources:

(1) Where can I find a corpus of German newspapers from the 17th century until the 1950s?

(2) What good introductory books on the subject of natural language processing, parsing and tagging are there?

(3) Where can I find the Orleans corpus of spoken French (created by Michel Blanc and Patricia Biggs)?

(4) Where can I find a parallel corpus of translations in English, French, German and Italian, ideally containing news stories?

(5) Where can I find a free or available English tagger other than Brill’s tagger?

Apart from the more restricted subject domain, the NLP Resource Metadata Questions significantly differ from the SemEval data used in QuestionBank in at least two other respects:

- The average length of the SemEval questions in QuestionBank is 47.58 characters and 9.45 words, whereas the NLP Questions average 81.17 characters and 12.88 words.

- Moreover, the distribution of questions types is quite different in the two cases. The SemEval data set used for QuestionBank is intended to query encyclopedic knowledge from sources such as Wikipedia. This means that the questions essentially include all possible question words such as *who*, *what*, *which*, *where*, *when*, *why*, *how*, etc. When

[2] LinguistList - http://linguistlist.org/

[3] Corpora List - http://www.hit.uib.no/corpora/

[4] Stack Overflow - http://stackoverflow.com/

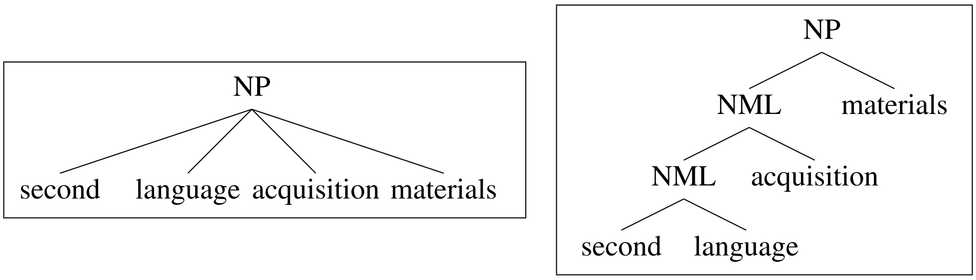


Figure 1: Comparing annotations for the compound noun *second language acquisition materials*: Penn-style annotation on the left, Vadas and Curran (2007) style annotation on the right

Question Word	% in QB	% in NLP-QT
Are there	0	5.45
At what	0.075	0
For what	0.025	0
How	3.75	0.5
How *	8.2	0
In what	0.825	0
In which	0.15	0
Is there	0	16.81
On what	0.075	0
On which	0.075	0
What	57.35	0.98
When	5	0.05
Where	6.075	75.07
Which	1.925	0.09
Who	11.375	0.1
Why	1.2	0
Other	3.822	0.93

Table 1: Distribution of question types in the two datasources; *How ** stands for questions like *how many, how much, how far, how long* etc.

querying NLP resource metadata, the emphasis is to a large extent on *where* and *is there* questions; the percentage for each type of question in the two datasources is showed in Table 1.

3 Comparing the Annotation of Base NPs

There is yet another property of both QuestionBank and the Penn Treebank that limits its usefulness for the QA application considered here. This concerns the flat-structure annotation style for noun phrases adopted in both resources. For example, in the question *Where can I find a German corpus containing second language acquisition materials?* the compound noun *second language acquisition materials* would be annotated in these resources as a single flat NP, as shown in the left column of Figure 1. Such a flat annotation does not provide sufficient information about the scope of each member of the compound. It is precisely this type of shortcoming that led Vadas and Curran (2007) to revise the Penn Treebank annotation style for NPs along the following lines:

- If the intended scope of a base NP leads to a strictly right-branching structure, then the Penn Treebank annotation remains unchanged.
- If the intended scope is partially or completely left-branching, then an extra node is introduced into the tree for each left-branching structure. The label of this node is either NML or JJP, depending on the lexical head of the local tree (noun or adjective, respectively).

The resulting annotation for the compound noun *second language acquisition materials* is shown in the right column of Figure 1.

From the point of view of semantic interpretation, the more contoured Vadas and Curran (2007) annotation style is to be preferred since it reflects the type of answer that is required, namely *materials for second language acquisition*, but not for example *acquisition materials for second language*, or *the second (batch) of language acquisition materials*.

It is precisely for this reason that we adopt the annotation style of Vadas and Curran (2007) for the NLP Resource Metadata Questions Treebank (henceforth abbreviated as NLP-QT).

4 Experimental Results

This section summarizes the set of experiments that we have conducted with the Vadas and Curran (2007) annotation style for NPs and in particular with the NLP-QT data set. We discuss two types of experiments:

- comparing the performance of the parser using different annotation styles for base NPs,
- experiments for optimizing the language model of a statistical parser in order to assist with the semi-automatic creation of the treebank.

All the experiments were performed with the Berkeley parser. The results are summarized in Table 2 and Table 3.

4.1 Parsing Results for Different Annotation Styles

Using Bikel (2004)'s parser, Vadas and Curran (2007) report that the parsing results slightly decrease when the parser is trained on the Penn Treebank with the modified annotation style for NPs. As Table 2 shows, we obtain a similar result when testing on section 23 of the Penn Treebank, using the Berkeley parser trained on sections 02-21 of the same treebank: there is minor drop in F-score from 90.43 to 89.96. We also confirm Gildea's finding that testing a parser on test sets from a different domain than the training sets results in a significant loss of performance: when using the same models that we used for the Penn Treebank experiments, the average F-score for test data from the Question Bank in a 10-fold cross-validation experiment is 79.944 for the model trained on the original Penn Treebank and 77.607 for the model trained on the modified Penn Treebank.

The above experiments were designed as a baseline for comparing the performance of the parser trained only on Penn Treebank data. But since our primary interest is in parsing questions as accurately as possible, we conducted a second set of experiments, summarized in the lower half of Table 2. Here additional training data from the Question Bank was added to both the original and the modified Penn Treebank training data. The decrease in performance caused by adding the QuestionBank training data together with the modified NP annotation on section 23 is comparable to the one caused by adding the modified NP annotation alone (a decrease from 90.263 to 90.04, whereas for the original Penn Treebank data the F-score decreased from 90.43 to 89.96), but this slight decrease is more than offset by the increase in semantic information obtained from the Vadas and Curran (2007) annotation for complex base NPs. Even more noteworthy is the big jump in F-score from 77.607 to 92.658 when adding the QuestionBank data to the training data.

4.2 Semi-automatic Creation of NLP-QT

The creation of a treebank is a time-consuming and expensive task if all the annotation has to be performed manually. It is therefore useful to investigate whether at least parts of the annotation can be performed automatically or by a combination of automatic analysis and manual post editing. To this end, we performed a set of parsing experiments, again using the Berkeley parser, where the test data are taken both from the QuestionBank and a seed set of 500 manually annotated questions from the NLP-QT. The results are shown in Table 3.

As in the experiments shown in the previous subsection, the performance with a model trained purely on Penn Treebank data (with NPs annotated in the Vadas and Curran (2007) style) serves as a baseline (the model is called *np-wsj* in the table). This model is then enriched by first adding annotated data from Question Bank and then by adding the manually annotated questions from the NLP-QT. We refer to these models as *np-wsjqb* and *np-wsjqblq_500*, respectively. The results are very encouraging on several dimensions:

1. overall parsing performance on the test data for both the *np-wsjqb* and the *np-wsjqblq_500* models is very good
2. adding questions from the NLP-QT yields a desired increase in performance
3. almost two-thirds of all questions from the test data yield a completely correct parse.

These three findings together make a semi-automatic construction of the NLP-QT entirely feasible. In fact, we are currently constructing the NLP-QT treebank in this semi-automatic fashion, using the same iterative approach to treebank construction adopted for the QuestionBank data by Judge et al. This approach involves iterations of manual post correction of automatically generated

Models	Section 23 of Penn Treebank			QuestionBank test section		
	Prec.	Recall	F-score	Prec.	Recall	F-score
Orig. PTB	90.480	90.390	90.430	79.285	80.617	79.944
PTB w/ NPs	90.000	89.920	89.960	77.546	77.670	77.607
Orig. PTB + QB	90.317	90.211	90.263	93.618	92.801	93.207
PTB w/ NPs + QB	90.095	89.985	90.040	92.592	92.725	92.658

Table 2: Comparison of parser performance when trained on different data sources with different annotation styles

Models	Questions test set			
	Prec.	Recall	F-score	Exact match
np-wsj	78.525	78.780	78.651	30.231
np-wsjqb	91.256	91.499	91.375	63.111
np-wsjqblq_500	92.128	92.186	92.157	64.801

Table 3: Parser performance increases when adding hand-corrected question data to the training set

	% of total	Avg. char. length	Avg. word length	Avg. const. no
Correct	48.59	61.55	11.41	20.94
Incorrect	51.41	100.96	17.85	31.96

Table 4: Average length and constituent count for the correctly/incorrectly parsed questions

parses, adding this post-corrected data set to the previously used training material and then retraining the parser with the enlarged data set.

One question that was not addressed in the approach by Judge et al. concerns the selection of the additional trees that will be manually corrected and then added to the training and test material in the next iteration. As Hwa (2001) has pointed out, this selection process can be critical in minimizing the amount of data that needs to be hand-corrected during grammar induction. She suggests several simple heuristics for ranking the candidate trees, two of which will be considered here. One heuristic is based on the often observed fact that, on average, longer sentences are harder to parse correctly than shorter ones. A second, related and somewhat more fine-grained variant of the first heuristic is based on the number of constituents obtained by the automatic parse of a sentence. Since the automatic parse is often at least partially incorrect, the constituent count of the parser will typically be just an estimate of the actual constituent count and related complexity of the sentence. Hwa suggests that when trees are added, the selected trees should match the average constituent count and length profile of the trees that were incorrectly parsed in the previous iteration.

We adopt Hwa's approach in the construction of the NLP-QT treebank. In order to use it effectively, it is necessary to inspect the results of the parser and in particular create an automatic profile of the completely correct versus partially incorrect parses. This type of error analysis is the subject of the next section.

4.3 Error Analysis

Table 4 summarizes the profiling of the 500 questions from the NLP-QT used in the 10-fold validation experiment. On average, 48.59 % of all sentences received an entirely correct parse. The average length in characters and in words as well as the average number of constituents of the correctly parsed sentences differ significantly from the questions where the parse is only partially correct.

These results provide a sound basis for applying Hwa's selection method: in the next iteration of optimizing the statistical model for the parser, sampling should focus on questions that match as closely as possible the character, word, and constituent count of the partially incorrect parse trees.

In order to get an impression of the kinds of mistakes that are made by the Berkeley parser, we are presenting two partially incorrect parse trees for the sentences in 6 and 7.

(6) Is there any freely available text corpus for Croatian, no smaller than 20k words?

(7) Where can I find information on chunking French and German texts?

The trees obtained by the Berkeley parser for these two sentences are shown in Figures 2 and 3, respectively. They exhibit the following typical attachment mistakes and misgroupings of conjuncts in a coordination structure:

The parse tree generated by the Berkeley parser for sentence 6 (Figure 2) contains several errors: two attachment errors (the PP *for Croatian* is not attached as a post-head modifier to the nominal head *text corpus*, but rather attached high as a sister of the preceding NP. Likewise, the modifier starting with *no smaller ...* is treated as an ADJP rather than an NP and is attached as well as a sister of the preceding NP and PP rather than to the complex NP *any ... for Croatian* in the gold parse. Moreover, the JJP *freely available* is incorrectly labelled as an ADJP.

The parse tree for sentence 7 (Figure 3) fails on the correct grouping and labelling of the coordinate structure *French and German texts*. The tagger treats the lexical token *chunking* as a noun (NN), rather than a gerund (VBG), and the lexical token *French* as a plural noun (NNS) rather than as an adjective (JJ). The parser then combines these two items into an NP, which is then coordinated with the NP *German texts*.

By hand correcting parse trees similar to the ones just discussed and by including them in the data set for retraining the parsing model in the next iteration, the performance of the parser on the types of constructions in question will improve and thereby minimize the amount of manual post editing as much as possible.

5 Conclusion and Future Work

In this paper we have presented the development process of the NLP-QT resource that will be used for data-driven parsing in the context of a domain-specific QA system for querying NLP resource metadata. We have motivated the need to build NLP-QT as a resource in its own right by comparing the Penn Treebank-style annotation scheme used for QuestionBank with the modified NP annotation for the Penn Treebank introduced by Vadas and Curran (2007). We have argued that this modified annotation scheme provides a better interface representation for semantic interpretation and have shown how it can be incorporated into the NLP-QT resource, without significant loss in parser performance.

The parsing experiments reported in the paper confirm the feasibility of an iterative, semi-automatic construction of the NLP-QT resource similar to the approach taken for QuestionBank. At the same time, we propose to improve the iterative refinement technique used for QuestionBank by adopting Hwa's heuristics for selecting additional material to be hand-corrected and added to the data set at each iteration.

Another important aspect in the creation of a treebank how to ensure a consistent and correct annotation of the linguistic material. Automatic error detection techniques that can be used to test the accuracy of the annotation have already been described in works like Květoň and Oliva (2002), for the part of speech annotation level, and Dickinson and Meurers (2005), for the syntactic annotation level. In future work on the NLP-QT, we plan to employ such methods in order to identify and to correct inconsistencies in the annotation.

References

Dan Bikel. 2004. *A distributional analysis of a lexicalized statistical parsing model.* Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004).

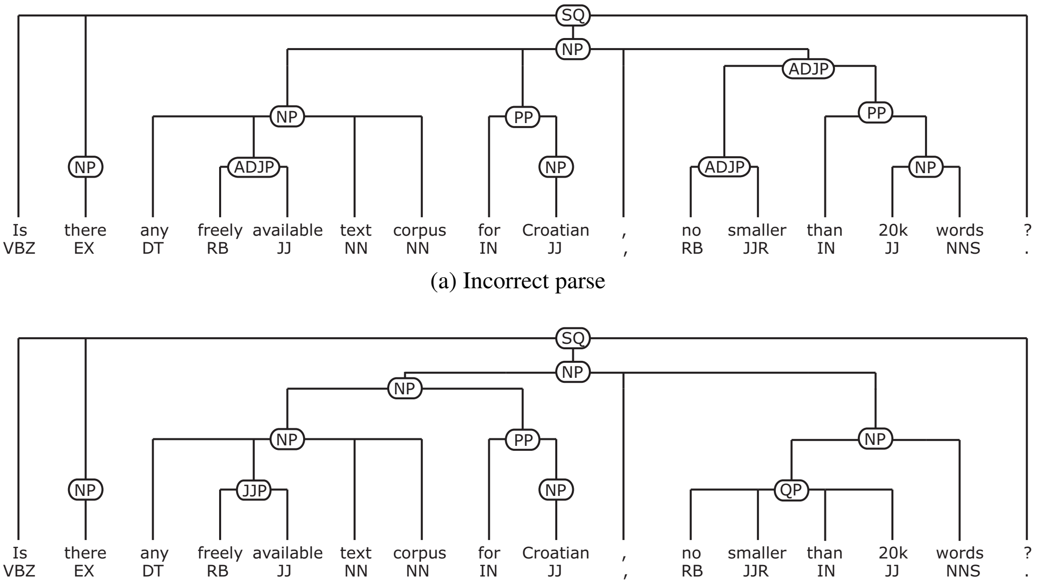
Markus Dickinson and W. Detmar Meurers. 2005. *Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters.* Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT 2005).

Daniel Gildea. 2001. *Corpus Variation and Parser Performance.* Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP 2001), 167–202.

Rebecca Hwa. 2001. *On Minimizing Training Corpus for Parser Acquisition.* Proceedings of the 2001 workshop on Computational Natural Language Learning - Volume 7.

John Judge, Aoife Cahill, and Josef van Genabith. 2006. *QuestionBank: Creating a Corpus of Parse-Annotated Questions.* Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (Coling-ACL 2006).

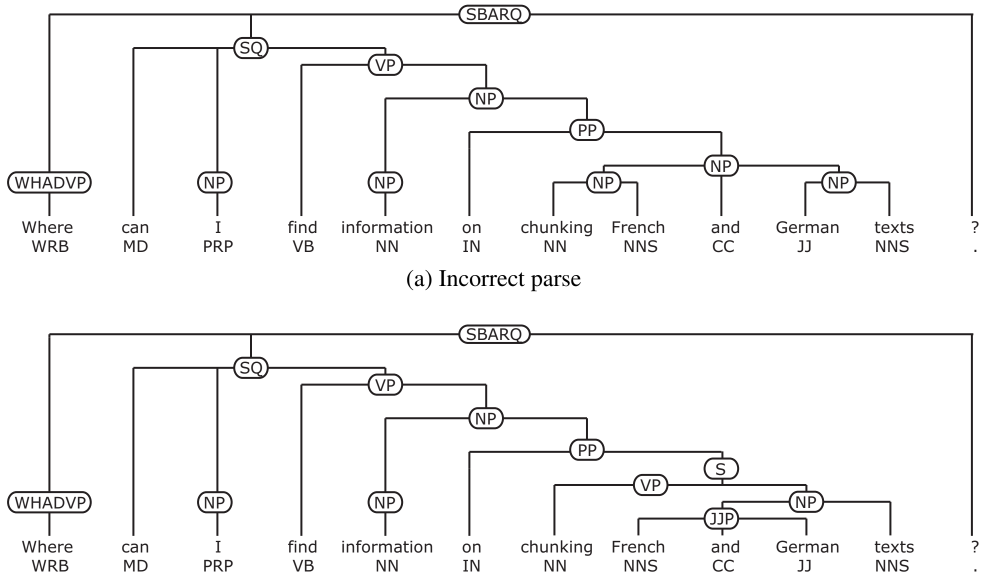
Pavel Květoň and Karel Oliva. 2002. *Achieving an Almost Correct PoS-Tagged Corpus.* Proceedings of the 5th International Conference Text, speech and dialogue (TSD 2002).



(a) Incorrect parse

(b) Correct parse

Figure 2: Incorrect (top) and correct (bottom) parse for example 6



(a) Incorrect parse

(b) Correct parse

Figure 3: Incorrect (top) and correct (bottom) parse for example 7

Dan Klein and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing.* Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL 2003), 423–430.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. *Building a Large Annotated Corpus of English: The Penn Treebank.* Computational Linguistics 19, 313–330.

Slav Petrov and Dan Klein. 2007. *Improved Inference for Unlexicalized Parsing.* Proceedings of NAACL HLT 2007.

Satoshi Sekine. 1997. *The Domain Dependence of Parsing.* Proceedings of the Fifth Conference on Applied Natural Language Processing, 96–102.

David Vadas and James R. Curran. 2007. *Adding Noun Phrase Structure to the Penn Treebank.* Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07), 240–247.