Semi-Automatic Construction of a Question Treebank

Karin Müller

University of Amsterdam,
Language and Inference Technology Group,
Nieuwe Achtergracht 166
1018WV Amsterdam, Netherlands
kmueller@science.uva.nl

Abstract

A method for the semi-automatic construction of a question treebank is presented. We exploit linguistic knowledge like grammatical functions, constituent structure and the relatively strict word order of English encoded in the Penn Treebank to generate semi-automatically questions. The outcome is a treebank of questions which might be useful for developing better tagging and parsing models for processing questions. We show that it is feasible to reuse the Penn Treebank. At the current stage our treebank comprises about 7000 questions, which can be easily extended.

1. Introduction

The aim of question answering (QA) systems is to find an appropriate answer to a question asked by a potential user. Answering a question involves a procedure of properly analyzing and classifying the question. The methods chosen by the system builders are either build on pattern-based approaches or involve also deeper syntactic analysis. Intuitively, it is clear that the information encoded in a question should be exploited as much as possible independent of the nature of the information (shallow linguistic information, like part-of-speech tags, n-grams, or deeper linguistic information e.g., subcategorization frames). Thus, some approaches (e.g., (Hermjakob, 2001), (Moldovan et al., 2000)) incorporate both deep and shallow parsers for analyzing questions. However, most state-of-the-art parsers for English (e.g., (Charniak, 1997), (Collins, 1996), (Bod, 2001)) are based on probabilistic grammars induced from treebanks (e.g. the Penn Treebank (Marcus et al., 1993)), which in turn are derived from newspaper texts.

Due to this specific text genre, the Penn Treebank comprises only a half percent full questions. It is often the case that these questions are more rhetorical than real questions. Beside the lack of questions in the Penn Treebank, the average sentence length differs significantly from real questions (20.54 tokens/sentence versus 9.98 tokens/sentence). As a consequence, those syntactic structures which are typical for questions are almost nonexistent. Obviously, this might hurt the performance of taggers and parsers for questions if these tools are solely trained on the Penn Treebank. In Section 2, we indeed find evidence for this assumption.

In our approach, we develop a method which automatically derives syntactically annotated questions from the Penn Treebank. Particularly, we intensively use grammatical functions to generate questions. The advantage of this approach is that we do not need manual annotations and that we additionally use the full annotation effort invested in the treebank. Moreover, we automatically inherit the rich syntactic structures of e.g., part-of-speech information, and deeper syntactic structure like noun phrases, prepositional phrases and subordinated phrases. After investigating the syntactic properties of questions in Section 3, the general idea of transforming statements to questions is presented in Section 4. In particular, we focus on the automatic generation of "who" and "what" questions, as well as locative ("where") and temporal ("when") questions by exploiting the grammatical functions in the Penn Treebank. In the last sections, we discuss our results and point at future work.

2. Pre-Study: Performance of Taggers and Parser on Questions

In this section, we investigate how tagging and parsing systems perform on questions. As we will discuss in the next section, normal newspaper texts seem to differ from questions. Specifically, syntactic structures which are typical for questions are very rare in newspaper texts. We assume that this hurts the performance of taggers and parsers when applied to a specific task like processing questions.

This assumption is partly supported by a study of Hermjakob (2001). There it is claimed that accuracy rates for parsing questions are significantly lower than for regular newspaper sentences. Moreover, Hermjakob showed that his parser, designed to predict the type of a given question, improves on this task if it is trained on the Penn Treebank augmented by a corpus of additional questions. Although it would be very interesting to confirm these results for a probabilistic state-of-the-art parser, this is beyond the scope of the current paper.

In the rest of this section, we investigate whether a tagging model trained on newspaper text performs equally good on the same text genre as on questions. We randomly choose 300 questions from a question collection ((Li and Roth, 2002) and (Hovy et al., 2001)) and automatically tag them with TnT (Brants, 2000) allowing multiple tags. In a second step, we manually correct the part-of-speech tags resulting in a tagged evaluation corpus. Third, we evaluate the English tagging model trained on the Penn Treebank on our manually corrected evaluation corpus. The model provided by TnT yields 92.3% tagging accuracy on questions. When evaluating the tagger on section 23 of the Penn Treebank, the model achieves 96.7%. Our results point out that questions and newspaper texts indeed differ syntactically.

3. Syntactic Properties of Questions

Probabilistic taggers and parsers trained on treebanks are dependent on the syntactic structures comprised in a training corpus. A corpus of newspaper texts and a corpus of questions, however differ in several syntactic features: sentence length, presence of a question word and which constituent (e.g., subject, object etc.) the question asks for.

One obvious syntactic feature is sentence length. The sentence length of questions measured on the collection ((Li and Roth, 2002) and (Hovy et al., 2001)) is significantly shorter than in the Penn Treebank, namely 9.98 words per sentence including punctuation. In contrast to the question collection, the average sentence length of section 02-21 of the Penn Treebank is 20.54 words per sentence (including punctuation, excluding traces).

A second syntactic feature is the existence of an initial question word. When analyzing the above mentioned collection consisting of over 5500 questions, the majority of the questions do comprise initial question words (95.4%). The distribution of the question words within the collection is displayed in Table 1, showing that 78% of the questions are *what*, *who*, *where* and *when* questions.

with question word	frequency	percentage
What ...	3585	60.2%
How ...	796	13.4%
Who ...	606	10.2%
Where ...	299	5%
When ...	157	2.6%
Which ...	111	1.9%
Why ...	107	1.8%
Whose ...	14	0.2%
Whom ...	4	0.007%
without question word	**number**	**percentage**
Name ...	92	1.5%
Define ...	4	0.007%

Table 1: distribution of question words

A third syntactic feature of questions is which part of the sentence is asked for: e.g., subject, object, or temporal, instrumental, locative complements. The word order is sensitive to this feature. So-called subject questions preserve the word order of a statement, whereas yes/no questions, and questions asking for objects and complements require partial inversion of the word order and additionally incorporate a form of the auxiliary verb "to do". In this paper, we want to show that it is feasible to exploit syntactic features to automatically build a question-treebank. In particular, we aim at generating *what* and *who*-questions which ask for the subject of a sentence and which do not need an inversion. We also exploit simple temporal and locative structures to gain *where* and *when* questions.

4. Generating the Question Treebank

In this section, we describe how we construct our question treebank. The type of questions we are generating are *what*, *who*, *when* and *where* questions. Our main idea is to use patterns that match with specific syntactic structures of a tree from the Penn Treebank using tgrep2 (Rhode, 2002). Our patterns exploit the constituent structure and the grammatical functions encoded in the treebank, and they are

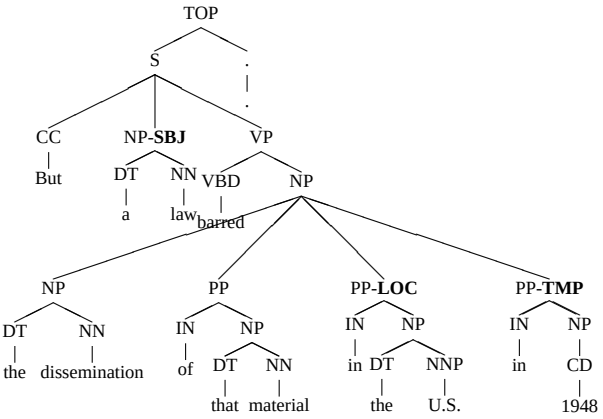


Figure 1: *Original* sentence

used to extract only those constituents which are necessary to transform the original tree to a question. It is also possible that several questions are generated from a single sentence provided several patterns match. We will exemplify our procedure by an example. Figure 1 shows the syntactic annotations from the Penn Treebank of the following sentence:

(1) *But a law barred the "dissemination" of that material in the U.S. in 1948.*

The sentence in Figure 1 comprises three constituents where the grammatical functions are annotated: a subject (NP-SBJ), a locative constituent (PP-LOC), and a temporal constituent (PP-TMP). The grammatical functions of these constituents and our patterns, which we specify in the following subsections, allow us to generate the subsequent questions:

(2) *What barred the dissemination of that material in the U.S. in 1948?*
(3) *Where did a law bar the dissemination of that material in 1948?*
(4) *When did a law bar the dissemination of that material in the U.S.?*

Example (2) and its syntactic tree is generated by a pattern that extracts the NP-SBJ and the VP from the original sentence. The subject, *a law*, is transformed to *what* as the subject is inanimated. By contrast, if the subject is animated, *who* questions are created. We use WordNet as a tool to decide if a noun is animated or not (Fellbaum, 1998). In the transformation process, we also make sure that the verb receives the correct verb form utilizing the CELEX dictionary (Baayen et al., 1993). The newly generated tree is displayed in Figure 2.

Sentence (3) and its syntactic tree can be automatically created from the original sentence by using the information that the constituent, *in the U.S.*, is a locative prepositional phrase. The construction of the question tree involves partial inversion of the statement and the transformation of the

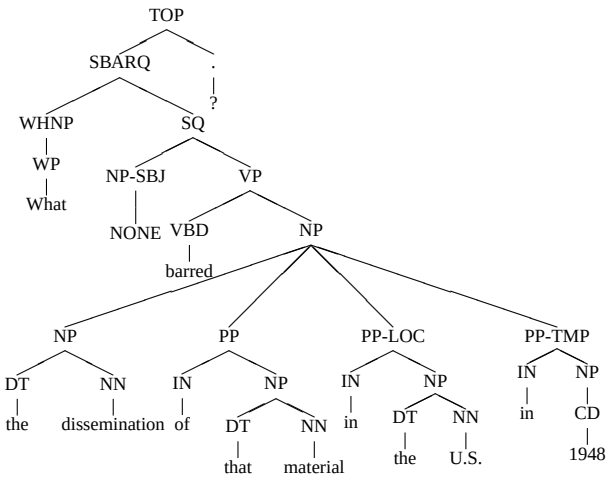


Figure 2: *What* question

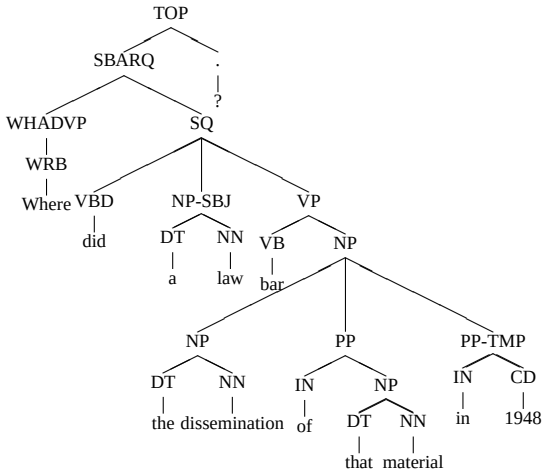


Figure 3: *Where* question

main verb to its base form. The syntactic tree of the locative question is displayed in Figure 3. Finally, the temporal question (4) can be built because the corresponding grammatical function (PP-TMP) is found in the original tree in Figure 1. Similar to the locative question, we have to partially change the word order of the original sentence and to adapt the verb form. Now, the new tree displayed in Figure 4 can be added to the question treebank.

4.1. *What/Who* questions

In this section, we describe the transformation of statements to *what* and *who* questions in more detail. The first step is to select all sentences which consist of an "S"-node dominating in turn a subject-NP and a VP-sister. We output (i) the right-most daughter of the subject-NP in order to be able to determine in a post-processing step if the subject is animated or inanimated, (ii) the verb and (iii) the daughters of the VP. The following code which was written in Tgrep2 exemplifies one of our pattern files (using: tgrep2 -c Penn-Treebank -m "%=n=\n%=vp=\nXXX\n" pattern-file)

```
/^S*/=s < /^NP-SBJ/=sb : =sb <<' /NNP|NNS|NN|NNPS/=n \
: =sb $ /VP/=vp : =vp < /VBP|VBD|VBZ/ \
: =vp < /^NP$/=np : =np < !/-NONE-/
```

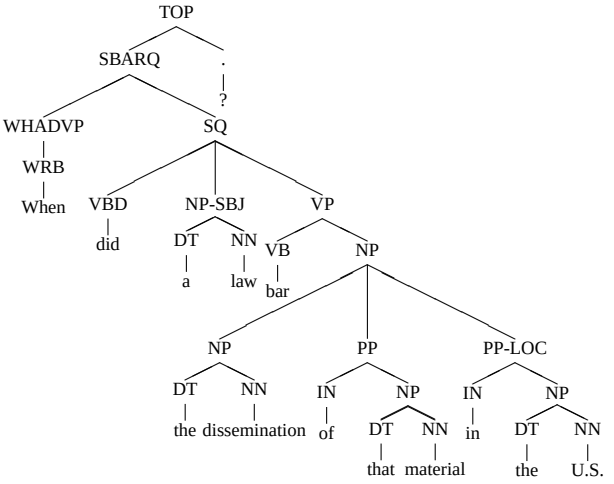
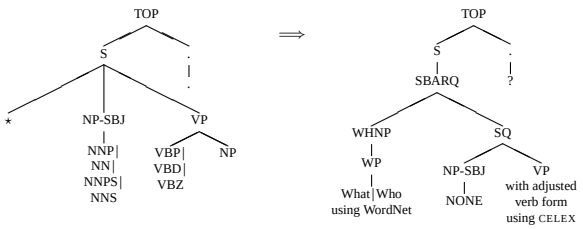


Figure 4: *When* question

In a second step, we post-process the output by determining the question word of the question using the POS-tag information of the head noun of the subject-NP. "NNP" and "NNPS" point at a person/group, which is then transformed to "who". The POS-tags "NN" and "NNS" are a mixed class. They comprise animated (*spokeswoman, 22 researchers, workers*) and inanimated nouns (*events, computers, production*). We determine the question word by the following simple procedure: The nouns are automatically looked-up in WordNet. If the synset of the word delivers a synset "person" or "group", the question word *who* is selected otherwise *what* is selected.

Furthermore, the verb has to be adjusted to the question word. If the verb is in present tense (VBP POS-tag), the verb is looked-up in the full form lexicon CELEX and is replaced by the third person verb form.

The last step is the construction of the syntactic structure of the question. The following picture shows an abstraction of the whole transformation process. The left-hand side describes the search pattern and the right-hand side shows the frame of the newly created question. Note, that the nodes which are represented by a star on the left-hand side are omitted, whereas the explicitly specified nodes and their daughter nodes are used for the construction of a question.

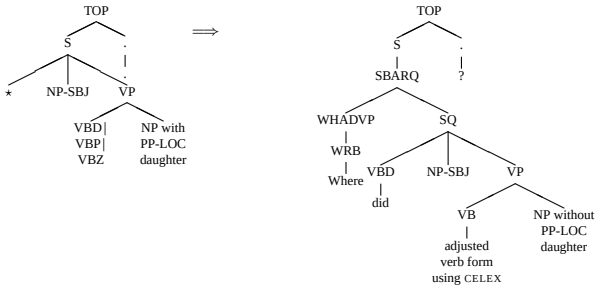


A similar transformation is applied to sentences comprising a verbal phrase with a second embedded verbal phrase (so-called verbal complexes).

4.2. Locative questions

The second class of questions we are generating are locative questions. We exploit the grammatical functions for locations, PP-LOC, to produce *where*-questions. For

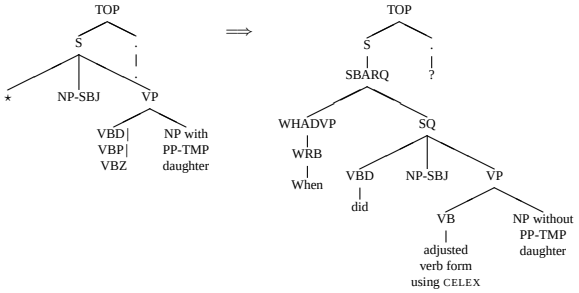
this purpose, we extract from all sentences which consist of a subject and a sister-VP with an embedded locational preposition phrase, the subject, the verbal phrase and the nodes which are embedded in the VP except for the PP-LOC. After extracting those sentences, we construct the trees by adding the question frame, Additionally, we post-process the subject to change upper case to lower case. Thus, we receive a question with a partially inverted word order:



We also use a pattern that extracts from a sentence with a verbal complex and an embedded PP-LOC *where* questions.

4.3. Temporal questions

When questions are generated by searching for syntactic structures that comprise temporal prepositional phrases. Our pattern extracts a non-empty subject, and a sister node VP with an embedded temporal PP. The temporal PP is transformed to *when*. The process of rebuilding a temporal question from the extracted constituents is similar to that described in section 4.2.. The subsequent figure displays the transformation process.



Additional patterns allow that a temporal constituent is a sister node of the subject. Moreover sentences with verbal complexes and an embedded temporal constituent are also matched.

5. Results

Our generation process delivers with our current version of basic search patterns almost 7000 questions comprising 2734 *what*-questions, 3301 *who*-questions, 300 *when*, and 145 *where*-questions. The average sentence length on our newly created treebank is slightly shorter than the sentence length of the Penn Treebank, namely 16 tokens per sentence.

6. Conclusion and Future Work

We presented a method for automatically constructing a question treebank by using the Penn Treebank. We exploit the grammatical functions to construct questions. The method shows that it is possible to generate a corpus of complex questions and their syntactic trees with minimal manual effort. We exemplified our approach with *who*, *what*, *where* and *when*-questions resulting in about 7000 questions from the Penn Treebank. Future work aims at generalizing this approach to other question types like *how*, *which* and *whom* questions. We also work on the creation of new patterns. The next step is to use our question treebank for the development of better tagging and parsing systems for processing questions. Another possible application for our question treebank is its use for training of (components for) question answering systems as we can not only generate the questions but also the corresponding answers. Moreover, it would be interesting to study the suitability of this approach for generating question treebanks for languages other than English.

7. References

Baayen, Harald R., Richard Piepenbrock, and H. van Rijn, 1993. The CELEX lexical database—Dutch, English, German. (Release 1)[CD-ROM]. Philadelphia, PA: Linguistic Data Consortium, Univ. Pennsylvania.

Bod, Rens, 2001. What is the Minimal Set of Fragments that Achieves Maximal Parse Accuracy? In *Proceedings ACL-2001*. Toulouse, France.

Brants, Thorsten, 2000. TnT - A Statistical Part- of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*.

Charniak, Eugene, 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

Collins, Michael, 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the ACL*. Santa Cruz.

Fellbaum, Christiane (ed.), 1998. *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Hermjakob, Ulf, 2001. Parsing and Question Classification for Question Answering. In *Proceedings of the Workshop on Open-Domain Question Answering at ACL-2001*.

Hovy, Eduard, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran, 2001. Toward Semantics-Based Answer Pinpointing. In *Proceedings of the DARPA Human Language Technology Conference (HLT)*. San Diego, CA.

Li, Xin and Dan Roth, 2002. Learning Question Classifiers. In *Proceedings of the COLING 2002*.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz, 1993. Building a large annotated corpus of English: the Penn Treebank. In *Proceedings of the ARPA Human Language Technology Workshop.*.

Moldovan, Dan, Sanda Harabagiu, Marius Pasca, Rada Milhalcea, Roxana Girju, Richard Goodrum, and Vasile Rus, 2000. The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL-2000)*. Hong Kong.

Rhode, Douglas L.T., 2002. *Tgrep2, User Manual, Version 1.06*. MIT, Cambridge, MA.