QuALiM at TREC 2005: Web-Question Answering with FrameNet

Michael Kaisser
Saarland University / DFKI GmbH
(now: School of Informatics, University of Edinburgh)
M.Kaisser@sms.ed.ac.uk

Abstract

In this paper I describe my TREC 2005 participation. The system used was–except from one new module–the same as in TREC 2004. In the following I will describe this new module, which uses the annotated Natural Language data collected in the FrameNet project in order to find paraphrases to answer questions. I will furthermore present and discuss the TREC 2005 results and compare them to those achieved in TREC 2004.

1 Introduction

This is the second time I participated in TREC. The 2005 system is mainly based on the 2004 system as described in [Kai04].

In a nutshell, the TREC 2004 system used two different answer strategies:

- The pattern based rephrasing algorithm.
- The Google fallback mechanism based on key words and key phases.

The rephrasing algorithm accesses patterns that reformulate questions to potential answer sentences. Such a pattern can be seen in figure 1. It accepts for example the question "When was Amtrak founded?" and reformulates it to "Amtrak was founded in ANSWER(NP)" and "In ANSWER(NP) Amtrak was founded". Google is searched for sentences containing the known parts of potential answer sentence, and if proper results are found, the answer is extracted.

The fallback mechanism creates rather simple queries based on key words and key phrases from the question. In the snippets returned from the search engine n-grams are mined and the most frequent one is returned as the answer.

The plan for TREC 2005 was to concentrate on the first algorithm. In 2004, a drawback with it was that the patterns it used to find answers had to be created manually, which of course is a time consuming matter. I thought about different methods to gain more patterns in a automated way. One idea was to use data from lexical resources like FrameNet[FL98] to create more patterns similar to those used in TREC 2004.

Unfortunately–because of other commitments–there was not much time to work on this idea. In late July, when the TREC evaluation took place, a first version of the FrameNet algorithm was implemented. It was able to some return some answers, but only for a small fraction of the test questions. As mentioned the 2004 modules were reused in 2005. Because of that and the fact that the FrameNet algorithm caught only for a few questions, the 2005

```
<pattern name="When+did+NP+Verb+NPorPP" level="5">
  <sequence>
    <word id="1">When</word>
    <word id="2">did</word>
    <parse id="3">NP</parse>
    <morph id="4">V INF</morph>
    <parse id="5">NP|PP</parse>
    <final>?</final>
  </sequence>

  <target name="target1">
    <ref>3</ref>
    <ref morph="V PAST">4</ref>
    <ref>5</ref>
    <word>in</word>
    <answer>NP</answer>
  </target>
  <target name="target2">
    <word>in</word>
    <answer>NP</answer>
    <punctuation optional="true">,<punctuation>
    <ref>3</ref>
    <ref morph="V PAST">4</ref>
    <ref>5</ref>
  </target>

  ... more targets ...

   <answerType phrases="NP|PP">
     <built-in weight="2">dateComplete</built-in>
     <namedEntity weight="4">Date</namedEntity>
     <built-in weight="3">year|in_year</built-in>
     <other ignore="true"/>
   </answerType>
 </pattern>
```

Figure 1: Example pattern as used in the 2004 version of the QuALiM system.

system did not differ that much from the 2004 system.

In the following I will give a few brief remarks about FrameNet in general. I will then explain how I used FrameNet to answer questions. Because the algorithm was not finished for TREC 2005 these remarks will be rather short. (Once I am finished with this work, it will be described in more detail elsewhere.) At the end of this paper I will report on the TREC evaluation results QuALiM received in 2005 and compare them to those of 2004.

2 FrameNet

FrameNet is a lexical database resource based on frame semantics and supported by corpus evidence. It documents the range of semantic and syntactic combinatory possibilities (valences) of target words (so-called lexical units). In order to do this, it contains human-

annotated sentences (currently more than 135,000), which exemplify the use of more than 6,100 lexical units organized into 625 semantic frames. As FrameNet is still in ongoing development, not all lexical units contain annotated sentences yet.

In FrameNet, words with similar semantics receive descriptions with identical role labels. The lexical units for "invent" and "design" for example describe the relation between the roles (*frame elements* in FrameNet's terminology) *Cognizer* and *Invention*. Similarly, the frames for the verbs "buy" and "sell" both list the frame elements *Buyer* and *Seller*. The annotated sentences show that the position of these frame elements differ in the the syntactic realizations of the different verbs.

The purpose of FrameNet is to create a sample selection of how Natural Language works. Many applications using FrameNet try to use the information the annotated sentences provide and apply it to sentences from other sources. Thus, FrameNet can help in getting closer to understanding Natural Language sentences or even texts. In the 2005 version of QuALiM I used FrameNet to

1. understand questions;
2. create a set of exact search engine queries;
3. analyze the sentences in the snippets returned by the search engine in order to find exact answers to the question.

3 Answering Questions with FrameNet

In the following, I will give a short explanation of how the system answers the question "When was the telegraph invented?"

First, the incoming questions is parsed using MiniPar[Lin98], and the resulting dependency tree simplified to the following structure:

```
head: invented(V)
subj: Who
whn:  Who
obj:  the telegraph
```

`head` indicates that the head of the question is the verb *invented*, `subj` indicates that the deep subject is *who* (which `whn` marks as also being a question word) and `obj` indicates that the deep object is *the telegraph*.

This provides enough information to look up the head verb in the FrameNet dictionary, where two lexical units for *invent.v* can be found.[1] One of the entries contains annotated sentences including the following:

Du Pont FE:Cognizer	in the USA	had	INVENTED lexical unit	nylon FE:Invention	in the late 1930s	...

Parts of the sentences are annotated with frame elements. The system parses all such annotated sentences with MiniPar to find out which semantic roles are assigned to which syntactic roles. It shows that usually the *Cognizer* role is realized as an NP at subject position, while *Invention* is an NP at object position.

[1] The system does not perform any form of word-sense disambiguation, yet.

As mentioned earlier, the analysis of the question shows that, in a potential answer sentence, the answer should be in subject relation to the verb "invent". Furthermore "the telegraph" needs to be in object relation to the verb. From this it can be concluded that the filler for the *Invention* frame element is "the telegraph", and that the question asks for a *Cognizer*.

The system can now give a pseudo-semantic formula for the question

```
invent_272(Cognizer=X, Invention="the telegraph")
```

and replace the frame elements *Cognizer* and *Invention* in each annotated sentence with their values from the question. For the above sentence the outcome would be:

```
ANSWER(NP) in the USA had invented the telegraph, in the late 1930s...
```

The PPs "in the USA" and "in the late 1930s" are recognized as additional information, most likely specific to the topic of the initial annotated sentence, but not transferable to the new domain, so they are–at least in the current version of the system–simply removed:

```
ANSWER(NP) had invented the telegraph
```

What we have so far can straightforward be translated into a pattern as used in the TREC 2004 system (see figure 2).

```
<pattern>
  <sequence>
    <word id="1">Who</word>
    <word id="2">invented</word>
    <parse id="3">NP</parse>
    <final>?</final>
  </sequence>

  <target name="FN_target1">
    <answer>NP</answer>
    <word>had</word>
    <ref>2</ref>
    <ref>3</ref>
  </target>
</pattern>
```

Figure 2: Pattern generated by the FrameNet algorithm.

From here the strategy is the same as in TREC 2004:

1. The system generates Google queries from the patterns, in this case: `"had invented the telegraph"`.
2. It extract sentences from the Google snippets.
3. It parse these sentences and checks whether they have the required syntactic structure.
4. If a sentence has the correct syntax, the potential answer can be extracted, because the system knows from the FrameNet data where in that sentence the answer is located. For example in "By 1832 Samuel FB Morse had invented the telegraph." it must be the NP preceding "had", thus: "Samuel FB Morse".

For the given example, the system was able to find the correct, exact answer and the open proposition shown above can now be completed:

```
invent_272(Cognizer="Samuel FB Morse", Invention="the telegraph")
```

* * *

As mentioned, at the time of TREC 2005, only an alpha version of the described algorithm existed. It was able to answer 35 of the 362 factoid questions asked, 25 of them correct. The main problem was that the mapping from parts of the question to a) a lexical unit and b) frame elements failed in most cases. Whenever this happened the question was simply not processed by the FrameNet module.

To sum up, in 2005 the following modules were used:

1. The 2004 rephrasing algorithm
2. The 2004 fallback mechanism
3. The new FrameNet algorithm

4 TREC Evaluation Results

TREC 2005	run 1	run 2
Factoid	0.207	**0.235**
List	0.029	**0.032**
Other	**0.147**	0.123
Combined	0.150	**0.158**

TREC 2004	run 1	run 2	run 3
Factoid	**0.343**	0.339	**0.343**
List	0.096	0.111	**0.125**
Other	0.145	0.181	**0.211**
Combined	0.232	0.242	**0.256**

Table 1: TREC 2005 results, compared to the 2004 results. The runs differed in parameter settings: The threshold to answer a question with NIL for factoid questions, the number of answers returned when answering a list question, the length of the answers in characters for other questions.)

Table 1 shows the results obtained in TREC 2005 compared to those of 2004. The 2005 results for all type of questions are worse than the 2004 results. This is somewhat surprising as the system was roughly the same as in TREC 2004 and evaluations show that the new FrameNet module improves the system performance rather than deteriorating it.

One reason for this is that the questions in 2005 seemed harder to answer than those of 2004 and that the assessors were more strict. The median accuracy scores for factoid questions in 2005 were lower than those in 2004 (0.170 compared to 0.152) and the best participant achieved worse results as well (0.770 compared to 0.713, see [Vor05] and [VD06]). Note that the scores dropped, although it should be the case that the systems participating in 2005 are better than those in 2004.

Another reason is that QuALiM could not deal with the event series introduced in 2005. QuALiM needs a complete question to start with. In 2004, where one could rely on the fact that all targets would be NPs, this meant to modify the series' target and insert into the question at the appropriate position. This was not the case in 2005. Targets like "Russian submarine Kursk sinks" or "Plane clips cable wires in Italian resort" are obviously not NPs. Thus, the question construction mechanism failed. Table 2 shows QuALiMs 2005 results

	All series		Event series		Non-event series	
W	223	0.616	61	0.663	162	0.600
U	30	0.083	10	0.107	20	0.074
X	24	0.066	4	0.043	20	0.074
R	85	0.235	17	0.182	68	0.252

Table 2: TREC results with respect to different types of the series' target.

(run 2) for different types of series. The leftmost table shows the results for all series. The next table shows only the results for those question series that had events as targets. The table to the right shows the results for all non-event series. As can be seen, QuALiM performs significantly better for the non-event series.

The slightly worse results for other questions can be explained by the fact that the targets in 2005 were more complicated than those in 2004. The bad 2005 results for list questions are due to a bug in the module designed to answer them.

References

[FL98] Colin F. Baker Charles J. Fillmore and John B. Lowe. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*, 1998.

[Kai04] Michael Kaisser. Question Answering by Searching Large Corpora with Linguistic Methods. In *The Proceedings of the 2004 Edition of the Text REtrieval Conference, TREC 2004*, 2004.

[Lin98] Dekang Lin. Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*, 1998.

[VD06] Ellen M. Vorheese and Hoa Trang Dang. Overview of the TREC 2005 Question Answering Track. In *The Proceedings of the 2005 Edition of the Text REtrieval Conference, TREC 2005*, 2006.

[Vor05] Ellen M. Vorheese. Overview of the TREC 2004 Question Answering Track. In *The Proceedings of the 2004 Edition of the Text REtrieval Conference, TREC 2004*, 2005.