Experiments at the University of Edinburgh for the TREC 2006 QA track

Michael Kaisser, Silke Scheible, Bonnie Webber University of Edinburgh {m.kaisser@sms.ed.ac.uk, s.scheible@sms.ed.ac.uk, bonnie@inf.ed.ac.uk}

Abstract

We describe experiments carried out at the University of Edinburgh for our TREC 2006 QA participation. Our main effort was to develop an approach to OA that is based on frame semantics. Two algorithms were implemented to this end, building on the lexical resources FrameNet, PropBank and VerbNet. The first algorithm uses the resources to generate potential answer templates to a given question, which are then used to pose exact, quoted queries to a web search engine and confirm which of the results contain an actual answer to the question. The second algorithm bases search queries on key words only, but it can recognize answers in a wider range of syntactic variants in its candidate sentence analysis stage. We discuss both approaches when applied to each of the resources and a combination of these.

We also describe how-in a later step-the found answer candidates are mapped to the AQUAINT corpus and how we answered *other* questions.

1 Introduction

This year we concentrated mainly on two approaches that employ lexical resources based on frame semantics like FrameNet (Fillmore and Lowe, 1998), PropBank (Palmer et al., 2005) and VerbNet (Schuler, 2005) for question answering. Compared to WordNet (Miller et al., 1993) which has been used widely in Question Answering, these other lexical resources are relatively new and therefore their usefulness for QA has still to be proven. In this paper we argue that, by employing these tools, we can indeed search for and extract answers in ways that were not or only indirectly possible beforehand. We describe two new and complementary techniques for using these resources and show the improvements to be gained when they are used individually and then together. We also point out problems that must be overcome to achieve these results.

Generally, the mentioned resources offer the following features which can be used to gain a better understanding of questions, sentences containing answer candidates, and the relations between them:

- They all provide verb-argument structures for a large number of lexical entries.
- FrameNet and PropBank contain semantically annotated sentences that exemplify the underlying frame.
- FrameNet contains not only verbs but also lexical entries for other part-of-speeches.
- FrameNet provides inter-frame relations that can be used for more complex paraphrasing to link the question and answer sentences.

By using these features, we are able, for example, to give a complete frame-semantic analysis of the following sentences and to recognize that they all contain an answer to the question "When was Alaska purchased?":

The United States purchased Alaska in 1867. Alaska was bought from Russia in 1867. In 1867, Russia sold Alaska to the United States. The acquisition of Alaska by the United States in 1867 is known as "Seward's Folly.

The first algorithm we present uses the three lexical resources to generate potential answercontaining templates. While the templates contain holes—in particular, for the answer—the parts that are known can be used to create exact quoted search queries. Sentences can then be extracted from the output of the search engine and annotated with respect to the resource being used. From this, an answer candidate (if present) can be extracted. The second algorithm analyzes the dependency structure of the annotated example sentences in FrameNet and PropBank. It then poses rather abstract queries to the web, but can in its candidate sentence analysis stage recognize answers in a wider range of syntactic possibilities. As we will see, the two algorithms are nicely complementary.

We additionally report on two other features of our TREC 2006 system. We describe how we map answer candidates found on the web to supporting AQUAINT documents and also our method for answering *other* questions, which received the best result this year.

2 QuALiM at TREC 2006: An Overview

Our system, QuALiM, uses four different algorithms to answer factoid questions. The pattern matching and the keyword/keyphrase-based algorithm are essentially the same as in our 2004 system (Kaisser and Becker, 2004). New this year are two algorithms based on frame semantics, which we focus on in this paper. The first of these algorithms described in Section 4 was finished at the time of the 2006 evaluation period. The second algorithm (see Section 6) only existed in a rudimentary form and so did not contribute significantly to the result. However, it is now fully functional and produces the results described in Section 7.

For list questions, we used a modified version of our keyword/keyphrase-based algorithm from the 2004 system. Similarly, *other* questions were answered with a modified version of the method used in 2004 (with the exception of run 3). However, because the system architecture changed considerably since 2004, both algorithm were reimplemented. Reimplementation of the *other* question algorithm involved tweaking parameters and paying attention to details etc. because we felt that the algorithm did not perform as well as it could have had in recent years. We did not devote much effort to the list question algorithm. The official TREC results reported in Section 12 mirror this in an interesting manner, especially if compared to our TREC 2004 results.

3 FrameNet, PropBank and VerbNet

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 (lexical units) and contains human-annotated sentences (currently more than 135,000), which exemplify the use of more than 8,900 lexical units organized in more than 625 semantic frames. Frames may also be linked to each other by particular semantic relations. Lexical units in different frames but with similar semantics usually receive identical role labels. For example, the frames for both "invent" and "design" relate the roles (*frame elements* in FrameNet's terminology) *Cognizer* and *Invention*, while those for the verbs "buy" and "sell" relate the frame elements *Buyer* and *Seller*.

PropBank uses the syntactic structure present in the Penn Treebank and adds to it a new layer of semantic annotation. As a result PropBank delivers parsed example sentences where the position of the semantic fillers for a given head verb are specified as nodes in the parse tree. Unlike FrameNet, it uses the argument labels ARG0, ARG1, ARG2 and various versions of ARGM for modifiers. Thus the relations between the arguments of different verbs are not obvious.

Finally, VerbNet does not list annotated example sentences for a given verb, but rather only specifies an abstract argument structure, as in this entry for "purchase":¹

Agent[NP]	VERB	Theme[NP]	
Agent[NP]	VERB	Theme[NP]	Source[PP-from]
Agent[NP]	VERB	Theme[NP]	Asset[PP-for]
Asset[NP]	VERB	Theme[NP]	

VerbNet furthermore aims to provide links to the other two aforementioned resources and to Word-Net. The entry for "purchase", for example, links to the WordNet synset {*buy, purchase*}.

4 Method 1: Question Answering by Natural Language Generation

The first method implemented uses the data available in the resources to generate potential answer templates to the question. While at least one component of such a template (the answer) is yet un-

¹A realization for the fourth structure would be: "\$50 won't even purchase a dress".

known, the remainder of the sentence can be used to query a web search engine. The results can then be analyzed, and if they match the originally-proposed answer template structure, an answer candidate can be extracted. In this section we will give a short explanation of how this method processes the question "Who purchased YouTube?"

First, the incoming question is parsed using Mini-Par (Lin, 1998), and the resulting dependency tree is simplified to the following structure:

```
head: purchased(V)
subj: Who
whn: Who
obj: YouTube
```

head indicates that the head of the question is the verb *purchased*, 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 *YouTube*. Furthermore, our system analyzes the tense as being *Simple Past*.

This provides enough information to look up the head verb in (for example) FrameNet, where one lexical unit (LU) for *purchase.v* can be found. One of the associated annotated sentences is:

The company FE:Buyer		had	PURCHA lexical	ASED unit		
	••	several PDMS terminals FE:Goods				••

As can be seen, parts of the sentences are annotated with frame elements, here *Buyer* and *Goods*. The system will parse and simplify the annotated sentences until we achieve a set of abstract frame structures, similar to those in VerbNet. By doing this we intentionally remove certain levels of information that were present in the original data, i.e. tense, voice, mood and negation. (In a later step we will reintroduce some of it.) For the above example the abstract structure is:

Buyer[Subj,NP] VERB Goods[Obj,NP]

Other structures similarly extracted from FrameNet are:

```
Buyer[Subj,NP] VERB Goods[Obj,NP]
Seller[Dep,PP-from]
Buyer[Subj,NP] VERB Goods[Obj,NP]
Money[Dep,PP-for]
Buyer[Subj,NP] VERB Goods[Obj,NP]
Recipient[Dep,PP-for]
...
```

This shows that usually, in active sentences, the *Buyer* role is realized as an NP at subject position, while *Goods* is an NP at object position. As mentioned earlier, the analysis of the question showed that, in a potential (active) answer sentence, the answer should be in subject relation to the verb "purchase". Furthermore, "YouTube" needs to be in object relation to the verb. Taking this information together, it can be concluded that the filler for the *Goods* frame element is "YouTube", and that the question asks for a *Buyer*.

This and the fact that the question was asked in past tense, enables the system to create the following potential answer templates by alternating all possible past tense forms and the voice:

```
ANSWER[NP] purchased YouTube
ANSWER[NP] (has|have) purchased YouTube
ANSWER[NP] had purchased YouTube
YouTube (was|were) purchased by
ANSWER[NP]
```

The part (or parts) of the templates that are known are quoted and sent to a search engine. For the fourth example, one query would be "YouTube was purchased by". From the snippets returned by the search engine, we extract candidate sentences and match them against the abstract frame structure from which the queries were originally created. In this way, we annotate the candidate sentences and are now able to identify the filler of the answer role. For example, the above query returns "On October 9, 2006, YouTube was purchased by Google for an incredible US\$1.65 billion", from which we can extract "Google", because it is the NP filling the buyer role.

Note that although we described the approach using an example based on FrameNet, we use the data provided in PropBank and VerbNet as well. In the case of PropBank the proceeding is basically the same. When using VerbNet, we can skip one processing step, because VerbNet does not list example sentences, but returns the frame structure directly.

So far, we have only discussed questions whose answer role is an argument of the head verb. However, for some question classes (especially time- or location-questions) this assumption does not hold. Here, the answer to the question is usually realized as an adjunct. This is an important difference for at least three reasons:

- 1. FrameNet and VerbNet do not or only sparsely annotate peripheral adjuncts. (PropBank how-ever does.)
- 2. In English, the position of adjuncts varies much more than those of arguments.
- 3. In English, different kinds of adjuncts can occupy the same position in a sentence, although naturally not at the same time.

The following examples illustrate point 2:

YouTube was purchased by Google on October 9. On October 9, YouTube was purchased by Google. YouTube was purchased on October 9 by Google.

All variations are possible, although they may differ in frequency. PPs conveying other adjuncts could replace all the above temporal PPs, or they could be added at other positions.

This behavior has to be accounted for, both when annotating the question with semantic roles and when creating and processing potential answer sentences. When annotating the answer role in a question which asks for an peripheral adjunct, the syntax of the question is of little help. Instead, we have to consult the answer type of the question.² We match certain answer types to certain roles, e.g. whenever a temporal or location answer type is detected, the answer role becomes, in FrameNet terms, *Place* or *Time*, respectively. We then use an abstract frame structure like the following to create the queries:

```
Buyer[Subj,NP,unknown]
VERB Goods[Obj,NP,"YouTube"]
```

While this lacks a role for the answer, we can still use it to create, for example, the query "has purchased YouTube". When sentences returned from the search engine are then matched against the abstract structure, we extract all PPs directly before the *Buyer* role, between the *Buyer* role and the verb and directly behind the *Goods* role. Then we check all these PPs on their semantic types and keep only those that match the answer type of the question (if any).

5 Making use of FrameNet Frames and Inter-Frame Relations

The methods presented so far can be used with all three resources. But FrameNet goes a step further than just listing verb-argument structures: It organizes all of its lexical entries in frames³, with relations between frames that can be used for a wider paraphrasing and inference. In the following we explain how we use this information to generate additional answer templates.

The *purchase.v* lexical unit, for example, is found in a *Commerce-buy* frame which also contains the lexical units *buy.v* and *purchase.n*. Both of these entries list annotated example sentences which use the same frame elements as *purchase.v*. It is therefore relatively straightforward to produce reformulations based on these related entries:

```
ANSWER[NP] bought YouTube
ANSWER[NP] (has|have) bought YouTube
YouTube (has|have) been bought by
ANSWER[NP]
```

It is also possible to generate target paraphrases with heads which are not verbs like ANSWER[NP-Genitive] purchase of YouTube.

Handling these is usually easier than sentences based on verbs, because no tense/voice information has to be introduced.

Furthermore, frames themselves can stand in different relations. The frame *Commerce_goods-transfer*, for example, stands both to the already mentioned *Commerce_buy* frame and to *Commerce_sell* in an *is_perspectivized_in* relation. The latter contains the lexical entries *retail.v*, *retailer.n*, *sale.n*, *sell.v*, *vend.v* and *vendor.n*. Here is one annotated example sentence listed in *sell.v*:

 the landowner FE:Seller		SOLD lexical unit	
 the land FE:Goods	to	developers FE:Buyer	

As can be seen, the frame elements of these lexical units use the same labels as the frame elements in the *purchase.v* and *buy.v* entries. This enables us to create answer templates like

²See (Kaisser and Becker, 2004) for an explanation of how QuALiM processes answer types.

³Note the different meaning of *frame* in FrameNet and Prop-Bank/VerbNet.

YouTube was sold to ANSWER[NP]. Other templates created from this frame seem odd, e.g. YouTube has been retailed to ANSWER[NP].

This is because the verb "to retail" usually takes mass-products as its object argument and not a company. But FrameNet does not make such fine-grained distinctions. However, we did not come across a single example in our experiments where such a phenomenon caused an overall wrong answer. Sentences like the one above will most likely not be found on the web (just because they are in a narrow semantic sense not well-formed). Yet even if we would get a hit, it probably would be a legitimate to count the odd sentence "YouTube had been retailed to Google" as evidence for the fact that Google bought YouTube.

6 Method 2: Combining Frame Semantics and Dependency Paths

The second method we have implemented compares the dependency structure of example sentences found in PropBank with the dependency structure of candidate sentences. (VerbNet does not list example sentences for lexical entries, so could not be used here.)

In a preprocessing step, all example sentences in PropBank are analyzed and the dependency paths from the head to each of the frame elements are stored. For example, in the sentence "The Soviet Union has purchased roughly eight million tons of grain this month", "purchased" is recognized as the head, "The Soviet Union" as *ARG0*, "roughly eight million tons of grain" as *ARG1*, and "this month" as an adjunct of type *TMP*. The stored paths to each are as follows:

headPath = /irole = ARG0, paths = {./s, ./subj,} role = ARG1, paths = {./obj} role = TMP, paths = {./mod}

This says that the head is at the root, ARG0 is at both surface subject (*s*) and deep subject (*subj*) position⁴, ARG1 is the deep object (*obj*), and TMP is a direct adjunct (*mod*) of the head.

Questions are annotated as described in Section 4. Sentences that potentially contain answer candidates are then retrieved by posing an rather abstract query consisting of key words from the question. Once we have obtained a set of candidate-containing sentences, we ask the following questions of their dependency structures compared with those of the example sentences from PropBank:

- 1a Does the candidate-containing sentence share the same head verb as the example sentence?
- 1b Do the candidate sentence and the example sentence share the same path to the head?
- 2a In the candidate sentence, do we find one or more of the example's paths to the answer role?
- 2b In the candidate sentence, do we find all of the example's paths to the answer role?
- 3a Can some of the paths for the other roles be found in the candidate sentence?
- 3b Can all of the paths for the other roles be found in the candidate sentence?
- 4a Do the surface strings of the other roles partially match those of the question?
- 4b Do the surface strings of the other roles completely match those of the question?

Tests 1a and 2a of the above are required criteria: If the candidate sentence does not share the same head verb or if we can find no path to the answer role, we exclude it from further processing.

Each sentence that passes steps 1a and 2a is assigned a weight of 1. For each of the remaining tests that succeeds, we multiply that weight by 2. Hence an candidate sentence that passes all the tests is assigned a weight 64 times higher than a candidate that only passes tests 1a and 2a. We take this as reasonable, as the evidence for having found a correct answer is indeed very weak if only tests 1a and 2a succeeded and very high if all tests succeed. Whenever condition 2a holds, we can extract an answer candidate from the sentence: It is the phrase that the answer role-path points to. All extracted answers are stored together with their weights. If we retrieve the same answer more than once, we simple add the new weight to the old ones. After all candidate sentences have been compared with all pre-extracted structures, the ones that do not show the correct semantic type are removed. This

⁴MiniPar allows more than one path between nodes due, for example, to traces. The given example is MiniPar's way of indicating that this is a sentence in active voice.

is especially important for answers that are realized as adjuncts (see Section 4). We choose the answer candidate with the highest score as the final answer.

We now illustrate this method with respect to our question "Who purchased YouTube?" The roles assignment process produces this result: "YouTube" is *ARG1* and the answer is *ARG0*. From the web we retrieve *inter alia* the following sentence: "Their aim is to compete with YouTube, which Google recently purchased for more than \$1 billion." The dependency analysis of relevant phrases is:

```
headPath = /i/pred/i/mod/pcomp-n/rel/i
phrase = "Google", paths = {./s, ./subj,}
phrase = "which", paths = {./obj}
phrase = "YouTube", paths = {./..}
phrase = "for more than $1 billion", paths = {./mod}
```

If we annotate this sentence by using the analysis from the above example sentence ("The Soviet Union has purchased ...") we get the following (partially correct) frame assignment: "Google" is *ARG0*, "which" is *ARG1*, "for more than \$1 billion" is *TMP*.

The following table shows the results of the 8 tests described above:



Test 1a and 2a succeeded, so this sentence is assigned an initial weight of 1. However, only three other tests succeed as well, so its final weight is 8. This rather low weight for a positive candidate sentence is due to the fact that we compared it against a dependency structure which it only partially matched. However, it might very well be the case that another of the annotated sentences shows a perfect fit. In such a case this comparison would result in a weight of 64. If these were the only two sentences that produce a weight of 1 or greater, the final weight for this answer candidate would be 8 + 64 = 72.

7 Evaluation of the Frame Semantics Approaches

We do not have a separate evaluation of what the above described frame semantics algorithms contributed to our TREC 2006 entry. Rather, we chose to evaluate them on the TREC 2002 QA test set because test sets from 2004 and beyond contain question series that pose problems that are separate from the research described in this paper. While the anaphora-resolution component we used for question series performed quite well, we feel that if one wants to evaluate a particular method, adding an additional module, unrelated to the actual problem, can distort the results. The evaluation also reflects searching for answers on the web rather than in the AQUAINT corpus, and so does not distinguish between supported and unsupported judgments.

Of the 500 questions in the TREC 2002 test set, 236 have *be* as their head verb. As the work described here essentially concerns *verb* semantics, such questions fall outside its scope. Evaluation has thus been carried out on only the remaining 264 questions.

For the first method (cf. Section 4), we evaluated system accuracy separately for each of the three resources, and then together, obtaining the following values:

FrameNet	PropBank	VerbNet	combined
0.181	0.227	0.223	0.261

For the combined run we looked up the verb in all three resources simultaneously and all entries from every resource were used. As can be seen, Prop-Bank and VerbNet perform similarly well, while FrameNet's performance is lower. These differences are due to coverage issues: FrameNet is still in development, and further versions with a higher coverage will be released. However, a closer look shows that coverage is a problem for all of the resources. The following table shows the percentage of the head verbs that were looked up during the above experiments based on the 2002 question set, that could not be found (not found). It also lists the percentage of lexical entries that contain no annotated sentences (s = 0), five or fewer $(s \le 5)$, ten or fewer $(s \le 10)$, or more than 50 (s > 50). Finally, the table provides the average (mean) number of lexical entries found per head verb (avg senses) and the average (mean) number of annotated sentences found per lexical entry (avg sent).⁵

⁵As VerbNet contains no annotated sentences, it is not listed. Note also, that these figures are not based on the resources in total, but on the head verbs we looked up for our evaluation.

	FrameNet	PropBank
not found	11.3%	8.3%
s = 0	41.6%	7.1%
$s \ll 5$	48.4%	35.2%
s <= 10	57.5%	45.4%
s > 50	8.7%	23.4%
avg senses	2.8	4.4
avg sent.	16.4	115.0

The problem with lexical entries only containing a small number of annotated sentences is that these sentences often do not exemplify common argument structures, but rather rare ones. As a solution to this coverage problem, we experimented with a cautious technique for expanding coverage. Any head verb, we assumed displays the following three patterns:

intransitive:	[ARG0] VERB
transitive:	[ARG0] VERB [ARG1]
ditransitive:	[ARG0] VERB [ARG1] [ARG2]

During processing, we then determined whether the question used the head verb in a standard intransitive, transitive or ditransitive way. If it did, and that pattern for the head verb was not contained in the resources, we temporarily added this abstract frame to the list of abstract frames the system used. This method rarely adds erroneous data, because the question shows that such a verb argument structure exists for the verb in question. By applying this technique, the combined performance increased from 0.261 to 0.284.

In Section 4 we reported on experiments that make use of FrameNet's inter-frame relations. The next table lists the results we get when (a) using only the question head verb for the reformulations, (b) using the other entries in the same frame as well, (c) using all entries in all frames to which the starting frame is related via the *Inheritance*, *Perspective_on* and *Using* relations (by using only those frames which show the same frame elements).

(a)	only question head verb	0.181
(b)	all entries in frame	0.204
(c)	all entries in related frames (with same frame elements)	0.215

Our second method described in Section 6, can only be used with FrameNet and PropBank, because VerbNet does not give annotated example sentences. Here are the results:

FrameNet	PropBank	
0.030	0.159	

Analysis shows that PropBank dramatically outperforms FrameNet for three reasons:

- 1. PropBank's lexicon contains more entries.
- PropBank provides many more example sentences for each entry.
- 3. FrameNet does not annotate peripheral adjuncts, and so does not apply to When- or Where-questions, which are common question types in TREC evaluation sets.

Of interest is the relatively small overlap found between correct answers returned by method 1 and by method 2. For PropBank, only 3.0% of all questions were answered correctly by both algorithms simultaneously. When the two methods are combined we obtain a accuracy of 0.306.

The methods we have described above are complementary. When we combine them (i.e. using method 1 with all three resources and our cautious coverage-extension strategy, with all additional reformulations that FrameNet can produce, and method 2), we achieved an accuracy of 0.367 on the mentioned 264 questions of the TREC 2002 corpus.

8 Frame Semantics: Related Work

So far, there has been little work at the intersection of QA and frame semantics. Fliedner (2004) describes the functionality of a planned system based on the German version of FrameNet, SALSA (Erk et al., 2003), but no so far no paper describing the completed system has been published.

Novischi and Moldovan (2006) use a technique that builds on a combination of lexical chains and verb argument structures extracted from VerbNet to re-rank answer candidates. The authors' aim is to recognize changing syntactic roles in cases where an answer sentence shows a head verb different from the question (similar to work described here in Section 4). However, since VerbNet is based on *thematic* rather than *semantic* roles, there are problems in using it for this purpose, illustrated by the following VerbNet pattern for *buy* and *sell*: [Agent] buy [Theme] from [Source] [Agent] sell [Recipient] [Theme]

Starting with the sentence "Peter bought a guitar from Johnny", and mapping the above roles for *buy* to those for *sell*, the resulting paraphrase in terms of *sell* would be "Peter sold UNKNOWN a guitar". That is, there is nothing blocking the Agent role of *buy* being mapped to the Agent role of *sell*, nor anything linking the Source role of *buy* to any role in *sell*. There is also a coverage problem: The authors report that their approach only applies to 15 of 230 TREC 2004 questions. They report a performance gain of 2.4% (MMR for the top 50 answers), but it does not become clear whether that is for these 15 questions or for the complete question set.

The way in which we uses the web in our first method is somewhat similar to (Dumais et al., 2002). However, our approach allows control of verb argument structures, tense and voice and thus we can create a much larger set of reformulations.

Regarding our second method, two papers describe related ideas: Firstly, in (Bouma et al., 2005) the authors describe a Dutch QA system which makes extensive use of dependency relations. In a preprocessing step they parsed and stored the full text collection for the Dutch CLEF QA-task. When their system is asked a question, they match the dependency structure of the question against the dependency structures of potential answer candidates. Additionally, a set of 13 equivalence rules allows transformations of the kind "the coach of Norway, Egil Olsen" \Leftrightarrow "Egil Olsen, the coach of Norway".

Secondly, Shen and Klakow (2006) use dependency relation paths to rank answer candidates. In their work, a candidate sentence supports an answer if relations between certain phrases in the candidate sentence are similar to the corresponding ones in the question.

Our work complements that described in both these papers, based as it is on a huge collection of semantically annotated example sentences: We do not require the structure of the answer candidate to match the question's structure: Instead we require it to match one of the annotated example sentences. This allows us to deal with a much wider range of syntactic possibilities, as the resources we use do not only document verb argument structures, but also the many ways they can be syntactically realized.

9 Frame Semantics: Discussion

Both methods presented in this paper build on frame semantics but with different aims in mind: The first method focuses on creating obvious answercontaining sentences. Because in these sentences, the head and the semantic roles are usually adjacent, it is possible to create exact search queries that will lead to answer candidates of a high quality. Our second method can deal with a wider range of syntactic variations but here the link to the answer sentences' surface structure is not obvious, thus no exact queries can be posed.

The first method performs better than the second one. This is largely due to the fact that the web contains such a huge collection of natural-language texts. It is important to obtain a set of high quality answer candidates, something that the first method was purposely build for. The second method, however, is very useful to fill the gaps of the first.

The overall accuracy we achieved suggest that employing frame semantics for question answering is indeed useful. The evaluation results presented in the last section compare nicely to TREC evaluation results.⁶ This is an especially strong point, because many high performing TREC systems combine miscellaneous strategies, most of them already know to perform well. Because the research question driving this work was to determine how frame semantics can benefit QA, we deliberately designed the approaches to *only* build on frame semantics. We did not chose to extend an already existing approach with a few frame semantic features.

Our results are convincing qualitatively as well as quantitavely: Detecting paraphrases and drawing inferences is a key challenge in question answering, which our methods achieve in various ways:

- They both recognize different verb-argument structures of the same verb.
- Method 1 controls for tense and voice: Our system will not take a future perfect sentence for an answer to a present perfect question.
- For method 1, no answer candidates altered by mood or negation are accepted.

⁶However, note the differences in the evaluation setup.

- Method 1 can create and recognize answer sentences, whose head is synonymous or related in meaning to the answers head. In such transformations, we are also aware of potential changes in the argument structure.
- The annotated sentences in the resources enables method 2 to deal with a wide range of syntactic phenomena.

10 Finding Supporting AQUAINT Documents

All of QuALiM's answer finding approaches use the web as their only source of information. We initially do not search the AQUAINT corpus. However, TREC requires us to return not only an answer but also a supporting AQUAINT document. Hence, in order to be able to participate in TREC, a module to map answer candidates found on the web to an AQUAINT documents is needed. The module we used this year is a new development based on Lucene (Hatcher and Gospodnetić, 2004) and replaces the module we used in 2004 and 2005.

For every question we take into account the top 5 answer candidates. Question 181.4 "Who is the captain of Manchester United?", for example, yields the following list:

```
1.00: "Gary Neville"
0.74: "Roy Keane"
0.53: "Steve Bruce"
0.44: "David Beckham"
0.43: "Ruud van Nistelrooy"
```

The numbers represent the relative confidence values that the system assigns to each candidate. Note that "Gary Neville" is the actual correct answer as of November 2006, though that was not the case at the time the AQUAINT corpus was written. "Roy Keane", "Steve Bruce" and "Ruud van Nistelrooy" were former captains of the club, while "David Beckham" played for Manchester, but was never the team's captain. (He was captain of the English national team until recently.)

For each of the answer candidates, we create a query that consists of keywords from the question and the answer. For the above example, such queries would be "Gary Neville" captain "Manchester United" or "Roy Keane" captain "Manchester United"

We then employ Lucene to find promising documents that contain all or many parts of the query. In the 100 best scoring documents we search for sentences containing the answer. We then assign a score to each of the sentences that reflects how many of the question keywords or keyphrases occur in it. Finally, we combine all this evidence using this formula:

$$final_score = x * web_score \\ + y * document_score \\ + z * sentence_score$$

Prior to this year's evaluation period we conducted an experiment on the TREC 2002 and TREC 2004 corpora in order to find the best assignments for the variables x, y and z. In order to do this, we ran the system several times with all possible assignments for x, y and z ranging from 0 to 10. We then evaluated each of the runs automatically. The assignments for the run that produced the best result was

$$x = 0, y = 10, z = 7.$$

Surprising is the assignment of 0 to x. This implies, that the amount of evidence on the web for a particular answer candidate should not be taken into account when searching for the final answer in the corpus. Does this suggest that the web is not a reliable source for question answering? We do not think so.

Rather, the web is a heterogeneous source of information. Lots of different views are expressed, and the information found has an emphasis on the here and now. (Despite the globalization, the "here" still largely depends on where the user lives, as this will affect the choice of web sites he/she decides to read, e.g. CNN.com vs. BBC.co.uk. Also, Google.com and Google.co.uk do not return the same results.) In contrast, the AQUAINT corpus is a rather homogeneous information source. Its articles have been compiled by mostly American (and partly Chinese) journalists in a period of a few years, which now lies six to ten years in the past. Information regarded as true from that view

and at that time period can still be found on the web: At the time of writing, "Bill Clinton is the President of the United States" gets 72 hits on Google, but "George W. Bush is the President of the United States" gets 404. (Consider also the results we obtained for the question "Who is the captain of Manchester United?"). But it is very hard to determine to which view or time period a certain fact found on the web belongs.

For the above example, our system finally changed its mind and returned "Roy Keane" as the final answer, because the corresponding query received a higher score than the one for "Garry Neville". Indeed, Roy Keane was Manchester captain from 1998 to 2005, so with respect to the AQUAINT corpus, this is the correct answer.

Hence, strictly speaking, we do not search the web for answers, but for answer candidates. The one that fits the AQUAINT corpus best becomes the final answer. While several researchers use the term *web* validation to describe their techniques, we might use the term *corpus validation* for ours.

11 Other Questions

QuALiM's approach to answering *other* questions in TREC 2006 is very similar to the one we used in TREC 2004 (Kaisser and Becker, 2004). However we reimplemented the algorithm for this year's contribution and-because we felt that our method potentially could perform better-spend considerable time on parameter tweaking etc. In 2004 we achieved the 10th best result for *other* question (Voorhees, 2005), in 2006 our method performed best.

In a first step, we create search queries from the targets. For simple targets (e.g. person names), we use the target as it is and an additional quoted version. (For the target "Warren Moon" our queries are: Warren Moon and "Warren Moon".) If the target is an event, we extract its main NP, so that for "John William King convicted of murder" the queries John William King convicted of murder are: and "John William King". If the target ends with a PP we add an additional, quoted query without that PP, so the queries for the target "Great Wall of China" are Great Wall of China, "Great Wall of China" and "Great Wall". In total, we developed eight such rules.

The queries are sent to a search engine and we count the occurrence of all non-stop words from the snippets returned in the top 50 results. This is the list of words we obtain for the target "Warren Moon":

```
148: "moon"
145: "warren"
30: "football"
27: "nfl"
20: "houston" "oilers"
18: "autographed"
11: "quarterback", "jerseys"
10: "hall", "time", "18",
    "throwback"
9: "player", "born", "only",
    "1956", "pro", "november"
    "sports"
8: "jersey", "1"
7: "los", "angeles", "team",
    "authentic", "career",
    "fame", "free"
```

We then search for sentences in the AQUAINT corpus that contain the target (or parts of it). If a sentence is regarded as too long (more than 100 non-white space characters) we cut it into smaller parts at punctuation marks it contains, if possible. Every sentence is then assigned a score that is the sum of all weights of all web words occurring in it divided by the length of the sentence in non-white space characters.

We now select the highest scoring sentence as our first answer sentence. This sentence is then removed from the list and the weight of every word that occurs in it is divided by 5 (with the exception of words from the target – here, "moon" and "warren" – whose weight is divided by 2). Again, all sentences are scored and the one with the highest weight is selected. This is repeated until the length of all answer sentences exceed a certain threshold.

The following table lists the length threshold of each run in non-whitespace characters and the results they received.

	run 1	run 2	run 3
length	1400	850	650
result	0.250	0.229	0.203

Run 1 and run 2 used the method described above. But for run 1 the best scoring sentences up to a limit of 200 non-whitespace characters were omitted. This was done because TREC explicitly wants participants to avoid repeating answers to previous factoid and list questions in the series as answers to an *other* question. While we do not explicitly exclude sentences that contain answers to previous questions, we observed in recent years that the best scoring sentences often contain the answers to these questions.

Run 3 reranked the results of the method described above based on importance indicators, such as superlative constructions and a number of other expressions. The assumption that these are useful with respect to answering other questions is based on a recent study of judgment consistency by Lin and Demner-Fushman (2006), where relevant other question nuggets were judged as either vital or okay by 10 different judges rather than the single assessor standardly used in TREC. Our investigation of the nuggets in Lin and Demner-Fushman's study yielded some interesting results. First of all, we created a word frequency list from all nuggets judged as vital in TREC 2004 and 2005 (where nuggets judged as vital by more than one assessor were counted as many times as judged vital). Our aim was to identify frequently occurring words that could serve as importance indicators in answering other questions. The following table shows the first 20 words of the frequency list:

rank	word	freq.]	rank	word	freq.
1	of	1.262	1	11	from	215
2	in	1.255		12	first	198
3	the	895		13	largest	187
4	to	755		14	million	181
5	and	498		15	at	175
6	for	37		16	on	174
7	a	411		17	with	164
8	is	341		18	as	157
9	was	265		19	has	153
10	by	254		20	most	139

While the top 20 of frequent words mainly include prepositions and forms of the verb *to be* or *to have*, ranks 12, 13, 14, and 20 are occupied by the words *first*, *largest*, *million* and *most*, respectively. Interestingly, all but one of the instances of "most" judged as *vital* are part of a superlative construction (i.e., "most" + adjective/adverb). Given that "largest" is also a superlative form, these results suggest that superlatives might be useful indicators of nugget importance. A closer investigation of the nuggets containing superlatives further shows that, on average, at least half of the TREC topics have at least one superlative nugget. Furthermore, of a total of 69 superlative nuggets, 32 (i.e. almost half) are judged *vital* by more than 9 assessors.

On the basis of these results, a scoring mechanism was devised for run 3 that reranked the snippets used for run 1 and 2 according to the presence of superlatives. Other potential importance indicators such as the words "first", "the only", and "unique", and a number of frequently occurring content words (such as "million", "oil", and "countries") were also considered, but given less weight.

As the results above show, run 3 performed worse than the other two runs. This may be due to either or both of the following. First, the evaluation metric employed by TREC seems to favor longer submissions. Secondly, a subsequent analysis of the nuggets containing superlatives (Scheible, 2007) showed that they can be distinguished by how the question target (T1) relates to the superlative target (T2): T1 and T2 may coincide (class S1 in table below), or T2 may be part of, closely related to or part of the comparison set of T1 (class S2 below). Finally, T1 may be unrelated or only distantly related to T2 (class S3 in table below):

	T1	nugget (T2 in italics)
S 1	AARP	Largest seniors organization
\$2	Florence	Nightingale Medal
32	Nightingale	highest international nurses award
S 3	Kurds	Irbil largest city controlled by Kurds

Of the 69 nuggets containing superlatives in TREC 2004 and 2005, 46 fall into subclass S1, 15 into subclass S2 and 8 into subclass S3. While we noted earlier that 32/69 (46%) of superlative-containing nuggets were judged vital by more than 9 assessors, these judgments are not equally distributed over the subclasses: the following table shows that 87% of S1 nuggets are judged as *vital*, while only 38% of S3 nuggets are.

	number	% of	% of
	of	vital	okay
	instances	judgments	judgments
S 1	46	87%	13%
S2	15	59%	40%
S 3	8	38%	60%

In the TREC 2006 evaluation of *other* questions, 60 superlative-containing nuggets were judged *vital*

or *okay* by the single TREC assessor. But the judgments show a similar distribution, with 91% of S1 superlatives judged as *vital*, but only 20% of class S3 superlatives:

	number	% of	% of	
	of	vital	okay	
	instances	judgments	judgments	
S1	32	91%	9%	
S2	13	54%	46%	
S 3	15	20%	80%	

These results strongly suggest that while superlatives alone are not a reliable indicator of nugget importance, S1 membership may be. This calls for a more sophisticated approach, where class S1 superlatives can be distinguished from other types of superlatives. We are currently at work on such a method, which can then be employed in the "other" questions task of TREC 2007.

12 Official TREC results

This are our TREC 2006 results as determined by NIST:

	run 1	run 2	run 3	median	best
factoid	0.323	0.303	0.293	0.186	0.578
list	0.051	0.053	0.054	0.087	0.433
other	0.250	0.229	0.203	0.125	0.250
comb.	0.207	0.192	0.181	0.134	0.394

The difference between our factoid runs lies in the number of NIL answers returned (run 3 > run 2 > run 1). The runs for list questions differed in the number of instances returned for each question (run 3 > run 2 > run 1). The difference between the *other* question runs is described in section 11.

Acknowledgments

This work was supported in part by Microsoft Research through the European PhD Scholarship Programme.

References

- Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jrg Tiedemann. 2005. Question Answering for Dutch using Dependency Relations. In *Proceedings of the CLEF 2005 Workshop*.
- Susan Dumais, Michele Bankom, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web Question Answering: Is More Always Better? *Proceedings of UAI 2003*.

- Katrin Erk, Andrea Kowalski, Sebastian Pado, and Manfred Pinkal. 2003. Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of ACL-2003*.
- Colin F. Baker Charles J. Fillmore and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING-ACL*.
- Gerhard Fliedner. 2004. Towards Using FrameNet for Question Answering. In Proceedings of the LREC 2004 Workshop on Building Lexical Resources from Semantically Annotated Corpora.
- Erik Hatcher and Otis Gospodnetić. 2004. Lucene in Action. Manning Publications Co.
- Michael Kaisser and Tilman Becker. 2004. Question Answering by Searching Large Corpora with Linguistic Methods. In *The Proceedings of the 2004 Edition of the Text REtrieval Conference, TREC 2004.*
- Jimmy Lin and Dina Demner-Fushman. 2006. Will Pyramids Built of Nuggets Topple Over? In *Proceedings of the HLT/NAACL*.
- Dekang Lin. 1998. Dependency-based Evaluation of MINIPAR. In Workshop on the Evaluation of Parsing Systems.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1993. Introduction to WordNet: An On-Line Lexical Database.
- Adrian Novischi and Dan Moldovan. 2006. Question Answering with Lexical Chains Propagating Verb Arguments. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.
- Silke Scheible. 2007. A Computational Treatment of Superlatives. PhD Thesis Proposal. University of Ed-inburgh.
- Karin Kipper Schuler. 2005. VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. Ph.D. thesis, University of Pennsylvania.
- Dan Shen and Dietrich Klakow. 2006. Exploring Correlation of Dependency Relation Paths for Answer Extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL.*
- Ellen M. Voorhees. 2005. Overview of the TREC 2004 Question Answering Track. In *The Proceedings of the* 2004 Edition of the Text REtrieval Conference, TREC 2004.