Question Answering based on Semantic Roles

Michael Kaisser **Bonnie Webber**
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
Scotland
m.kaisser@sms.ed.ac.uk, bonnie@inf.ed.ac.uk

Abstract

This paper discusses how lexical resources based on semantic roles (i.e. FrameNet, PropBank, VerbNet) can be used for Question Answering, especially Web Question Answering. Two algorithms have been implemented to this end, with quite different characteristics. We discuss both approaches when applied to each of the resources and a combination of these and give an evaluation. We argue that employing semantic roles can indeed be highly beneficial for a QA system.

1 Introduction

A large part of the work done in NLP deals with exploring how different tools and resources can be used to improve performance on a task. The quality and usefulness of the resource certainly is a major factor for the success of the research, but equally so is the creativity with which these tools or resources are used. There usually is more than one way to employ these, and the approach chosen largely determines the outcome of the work.

This paper illustrates the above claims with respect to three lexical resources – FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005) and VerbNet (Schuler, 2005) – that convey information about lexical predicates and their arguments. 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.

Compared with WordNet (Miller et al., 1993)–which has been used widely in QA–FrameNet, PropBank and VerbNet are still relatively new, and therefore their usefulness for QA has still to be proven. They 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.

In this paper we describe two methods that use these resources to annotate both questions and sentences containing answer candidates with semantic roles. If these annotations can successfully be matched, an answer candidate can be extracted. 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.

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 41–48,
Ann Arbor, June 2005.

The first algorithm we present uses the three lexical resources to generate potential answer-containing 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 deal with a wider range of syntactic possibilities. As we will see, the two algorithms are nicely complementary.

2 Method 1: Question Answering by Natural Language Generation

The first method implemented uses the data available in the resources to generate potential answer sentences to the question. While at least one component of such a sentence (the answer) is yet unknown, 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 sentence structure, an answer candidate can be extracted.

The first step is to annotate the question with its semantic roles. For this task we use a classical semantic role labeler combined with a rule-based approach. Keep in mind that our task is to annotate questions, not declarative sentences. This is important for several reasons:

1. The role labeler we use is trained on FrameNet and PropBank data, i.e. mostly on declarative sentences, whose syntax often differs considerably from the sytax of questions. Aa a result, the training and test set differ substantially in nature.
2. Questions tend to be shorter and simpler syntactically than declarative sentences–especially those occurring in news corpora.
3. Questions contain one semantic role that has to be annotated but which is not or is only implicitly (through the question word) mentioned – the answer.

Because of these reasons and especially because many questions tend to be gramatically simple, we found that a few simple rules can help the question annotation process dramatically. We rely on MiniPar (Lin, 1998) to find the question's head verb, e.g. “purchase” for “Who purchased YouTube?” (In the following we will often refer to this question to illustrate our approach.) We then look up all entries in one of the resources, and for FrameNet and PropBank we 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.) Here is what we find in FrameNet for “purchase”:

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

A syntactic analysis of the question (also obtained from MiniPar) shows that “Who” is the (deep) subject and “YouTube”, the (deep) object. The first of the above frames fits this analysis best, because it lists only the two roles with the desired grammatical functions. By mapping the question analysis to this frame, we can assign the roles *Goods* to “YouTube” and *Buyer* to “Who”. From this we can conclude that the question asks for the *Buyer* role.

An additional point suitable to illustrate why a few simple rules can achieve in many cases more that a statistical classifier, are *When*- and *Where*-questions. Here, the hint that leads to the correct detection of the answer role lies in the question word, which is of course not present in the answer sentence. Furthermore, the answer role in an answer sentence will usually be realized as a PP with a totally different dependency path than the one of question's question word. In contrast, a rule that states that whenever a temporal or location question is detected the answer role becomes, in FrameNet terms, *Place* or *Time*, respectively, is very helpful here.

Once the role assignment is complete, we use all abstract frames which contain the roles found in the question to generate potential answer templates.

This is also the point where we reintroduce tense and voice information:[1] If the question was asked in the a past tense, we will now create from each abstract frame, all surface realizations in all past tenses, both in active and passive voice. If we had used the annotated data directly without the detour over the abstract frames, we would have difficulty sorting out negated sentences, those in undesired moods and those in unsuitable tenses. In contrast our approach guarantees that all possible tenses in both voices are generated, and no meaning-altering information like mood and negation is present. For the example given above we would create *inter alia* the following answer templates:

```
ANSWER[NP] purchased YouTube
YouTube was purchased by ANSWER[NP]
ANSWER[NP] had purchased YouTube
...
```

The part (or parts) of the templates that are known are quoted and sent to a search engine. For the second example, this 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.

So far, we have mostly discussed questions whose answer role is an argument of the head verb. However, for questions like “When was YouTube purchased?” this assumption does not hold. Here, the question asks for 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 however 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.

[1] While we strip off mood and negation during the creation of the abstract frames, we have not yet reintroduced them.

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 peripheral adjuncts (e.g. “for $1.65 billion”) could replace all the above temporals PPs, or they could be added at other positions.

The special behavior of these types of questions has not only to be accounted for when annotating the question with semantic roles, but also and when creating and processing potential answer sentences. We 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 can extract all PPs directly before the Buyer role, between the Buyer role and the verb and directly behind the Goods role. Then we can check all these PPs on their semantic types and keep only those that match the answer type of the question (if any).

3 Making use of FrameNet Frames and Inter-Frame Relations

The method 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 frames[2], with relations between frames that can be used for a wider paraphrasing and inference. This section will explain how we make use of these relations.

The *purchase.v* entry is organized in a frame called *Commerce_buy* which also contains the entries for *buy.v* and *purchase_((act)).n*. Both these entries are annotated with the same frame elements as *purchase.v*. This makes it possible to formulate alternative answer templates, for example: `YouTube was bought by ANSWER[NP]` and

[2] Note the different meaning of *frame* in FrameNet and PropBank/VerbNet respectively.

`ANSWER[NP-Genitive] purchase of YouTube.`
The latter example illustrates that we can also generate target paraphrases with heads which are not verbs. 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, relates 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*. Again, the frame elements used are the same as for *purchase.v*. Thus we can now create answer templates like `YouTube was sold to ANSWER[NP]`. Other templates created from this frame seem odd, e.g. `YouTube has been retailed to ANSWER[NP].` 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. Interestingly, 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.

4 Method 2: Combining Semantic Roles and Dependency Paths

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

In a pre-processing step, all example sentences in PropBank and FrameNet 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" (found in PropBank), "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 = $\downarrow i$
role = *ARG0*, paths = $\{\downarrow s, \downarrow subj\}$
role = *ARG1*, paths = $\{\downarrow obj\}$
role = *TMP*, paths = $\{\downarrow mod\}$

This says that the head is at the root, ARG0 is at both surface subject (*s*) and deep subject (*subj*) position[3], ARG1 is the deep object (*obj*), and TMP is a direct adjunct (*mod*) of the head.

Questions are annotated as described in Section 2. Sentences that potentially contain answer candidates are then retrieved by posing a 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[4]:

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.

[3] MiniPar 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.

[4] Note that our proceeding is not too different from what a classical role labeler would do: Both approaches are primarily based on comparing dependency paths. However, a standard role labeler would not take tests 3a, 3b, 4a and 4b into account.

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 a 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 is especially important for answers that are realized as adjuncts, see Section 2. 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 the relevant phrases is:

headPath = $\downarrow i \downarrow i \downarrow pred \downarrow i \downarrow mod \downarrow pcom\text{-}n \downarrow rel \downarrow i$
phrase = "Google", paths = $\{\downarrow s, \downarrow subj\}$
phrase = "which", paths = $\{\downarrow obj\}$
phrase = "YouTube", paths = $\{\uparrow i \uparrow rel\}$
phrase = "for more than $1 billion", paths = $\{\downarrow mod\}$

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) role 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:

1a	OK	2a	OK	3a	OK	4a	–
1b	–	2b	OK	3b	OK	4b	–

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 $8 + 64 = 72$.

5 Evaluation

We choose to evaluate our experiments with 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 we participated in TREC 2004, 2005 and 2006, with an anaphora-resolution component that 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. Additionally, because we are searching for answers on the web rather than in the AQUAINT corpus, we do 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 2), 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, PropBank and VerbNet perform equally well, while FrameNet's performance is significantly 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 ($s <= 5$), ten or fewer ($s <= 10$), or more than 50 ($s > 50$). Furthermore, the table lists the average number of lexical entries found per head verb (*avg senses*) and the average number of annotated sentences found per lexical entry (*avg sent*). [5]

	FrameNet	PropBank
not found	11%	8%
$s = 0$	41%	7%
$s <= 5$	48%	35%
$s <= 10$	57%	45%
$s > 50$	8%	23%
avg senses	2.8	4.4
avg sent.	16.4	115.0

The problem with lexical entires only containing a small number of annotated sentences is that these sentences often do not exemplify common argument structures, but rather seldom 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 2 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 4, 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.
2. 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.

The methods we have described above are complementary. When they are combined so that when method 1 returns an answer it is always chosen as the final one, and only if method 1 did not return an answer were the results from method 2 used, we obtain a combined accuracy of 0.306 when only using PropBank. When using method 1 with all three resources and our cautious coverage-extension strategy, with all additional reformulations that FrameNet can produce and method 2, using PropBank and FrameNet, we achieve an accuracy of 0.367.

We also evaluated how much increase the described approaches based on semantic roles bring to our existing QA system. This system is completly web-based and employs two answer finding strategies. The first is based on syntactic reformulation rules, which are similar to what we described in section 2. However, in contrast to the work described in this paper, these rules are manually created. The second strategy uses key words from the question as queries, and looks for frequently occuring n-grams in the snippets returned by the search engine. The system received the fourth best result for factoids in TREC 2004 (Kaisser and Becker, 2004) (where both

[5] As 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.

just mentioned approaches are described in more detail) and TREC 2006 (Kaisser et al., 2006), so it in itself is a state-of-the-art, high performing QA system. We observe an increase in performance by 21% over the mentioned baseline system. (Without the components based on semantic roles 130 out of 264 questions are answered correct, with these components 157.)

6 Related Work

So far, there has been little work at the intersection of QA and semantic roles. Fliedner (2004) describes the functionality of a planned system based on the German version of FrameNet, SALSA, 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 2). 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 use the web in our first method is somewhat similar to (Dumais et al., 2002). However, our system 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 pre-processing 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" ⇔ "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 large collection of semantically annotated example sentences: We only require a candidate sentence 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.

7 Discussion

Both methods presented in this paper employ semantic roles but with different aims in mind: The first method focuses on creating obvious answer-containing 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 overall accuracy we achieved suggests that employing semantic roles for question answering is indeed useful. Our results compare nicely to recent TREC evaluation results. This is an especially strong point, because virtually all high performing TREC systems combine miscellaneous strategies, which are already know to perform well. Because the research question driving this work was to determine how semantic roles can benefit QA, we deliberately designed our system to *only* build on semantic roles. We did not chose to extend an already existing system, using other methods with a few features based on semantic roles.

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.
- 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.

8 Conclusion

This paper explores whether lexical resources like FrameNet, PropBank and VerbNet are beneficial for QA and describes two different methods in which they can be used. One method uses the data in these resources to generate potential answer-containing sentences that are searched for on the web by using exact, quoted search queries. The second method uses only a keyword-based search, but it can annotate a larger set of candidate sentences. Both methods perform well solemnly and they nicely complement each other. Our methods based on semantic roles alone achieves an accuracy of 0.39. Furthermore adding the described features to our already existing system boosted accuracy by 21%.

Acknowledgments

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

References

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

Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg 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*.

Gerhard Fliedner. 2004. Towards Using FrameNet for Question Answering. In *Proceedings of the LREC 2004 Workshop on Building Lexical Resources from Semantically Annotated Corpora*.

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*.

Michael Kaisser, Silke Scheible, and Bonnie Webber. 2006. Experiments at the University of Edinburgh for the TREC 2006 QA track. In *The Proceedings of the 2006 Edition of the Text REtrieval Conference, TREC 2006*.

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.

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*.