Generating Annotated Corpora for Reading Comprehension and Question Answering Evaluation

Generating Annotated Corpora for Reading Comprehension and Question Answering Evaluation

Tiphaine Dalmas Jochen L. Leidner Bonnie Webber Claire Grover Johan Bos
Institute for Communicating and Collaborative Systems (ICCS),
School of Informatics, University of Edinburgh,
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK.
t.dalmas@sms.ed.ac.uk

Abstract

Recently, reading comprehension tests for students and adult language learners have received increased attention within the NLP community as a means to develop and evaluate robust question answering (NLQA) methods. We present our ongoing work on automatically creating richly annotated corpus resources for NLQA and on comparing automatic methods for answering questions against this data set. Starting with the CBC4Kids corpus, we have added XML annotation layers for tokenization, lemmatization, stemming, semantic classes, POS tags and best-ranking syntactic parses to support future experiments with semantic answer retrieval and inference. Using this resource, we have calculated a baseline for word-overlap based answer retrieval (Hirschman et al., 1999) on the CBC4Kids data and found the method performs slightly better than on the REMEDIA corpus. We hope that our richly annotated version of the CBC4Kids corpus will become a standard resource, especially as a controlled environment for evaluating inference-based techniques.

1 Introduction

The goal of computer systems capable of simulating understanding with respect to reading a story and answering questions about it has attracted researchers since the early 1970s. We present our ongoing work on creating richly annotated corpus resources for NLQA that can provide input for a wide range of NLQA techniques and simultaneously support their evaluation and cross comparison.

2 Related Work

The challenge to computer systems of reading a story or article and demonstrating understanding through question answering was first addressed in Charniak's Ph.D. thesis (Charniak, 1972). That work showed the amount and diversity of both logical and common sense reasoning needed to link together what was said explicitly in the story or article and thereby to answer questions about it.

More recent work has stressed the value of reading comprehension exams as a research challenge in terms of (1) their targeting successive skill levels of human performance, and hence their potential to challenge automated systems to successively higher levels of performance (Hirschman et al., 1999), and (2) the existence of independently developed scoring algorithms and human performance measures, as an alternative to the special purpose evaluations developed for TREC Open Domain Question-Answering (Voorhees and Tice, 1999).

The first attempt to systematically determine the feasibility of reading comprehension tasks as a research challenge for automated systems was Deep Read (Hirschman et al., 1999). Deep Read established a baseline on a professionally-developed

remedial reading comprehension test for children in grades 3-6 (ages 8-12), using a simple bag-of-words approach. Scoring essentially by word intersection with the answer key provided by the test designer, Deep Read's simple approach produced sentence-level answers that agreed with sentences supporting the answer key (a metric called **HumSent**, see below) 30% of the time. That was sufficient to establish reading comprehension tests as a tractable research problem for automated systems.[1] This work was followed in 2000 by both an ANLP-NAACL workshop on *Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*[2] and a Summer workshop on technology for reading comprehension QA at the Johns Hopkins University.[3]

3 Automatic Linguistic Annotation

Our work is driven by the following observation (Cotton and Bird, 2002): "With all the annotations expressed in the same data model, it becomes a straightforward matter to investigate the relationships between the various linguistic levels. Modeling the interaction between linguistic levels is a central concern."

The CBC4Kids corpus was developed at MITRE[4], based on a collection of newspaper stories for teenagers written for the CBC's WWW site.[5] To each article selected for inclusion in the corpus, Ferro and her colleagues added a set of 8-10 questions of various degrees of difficulty (Ferro, 2000). The corpus also includes one or more answers for each question in the form of a disjunction of a phrase or a clause (the "answer key").

Due to the wide availability of XML processing tools, we decided to define an XML DTD for the CBC4Kids corpus and to convert various automatically[6] obtained linguistic forms of annotation into XML and integrate them so as to provide a rich knowledge base for our own NLQA experiments and potential re-use by other groups. We selected a set of tools with the guiding principles of 1) public availability, 2) usefulness for our replication a Deep Read-style baseline system, and 3) quality of the automatic annotation. Because most available tools (with the exception of TTT, (Grover et al., 2000)) do not output XML, we had to develop a set of converters.

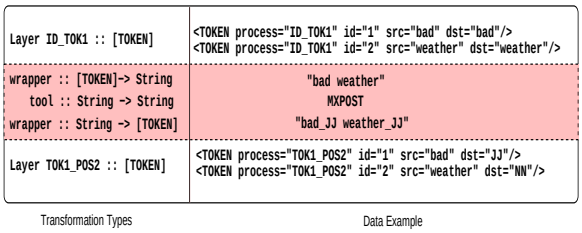


Figure 1: Building a new layer of TOKEN tags.

Each sentence has three different representations: 1) the original string, 2) a list of tags labeled `TOKEN` encoding the results from linguistic tools that give information on words (POS tags, stems, etc.), 3) a list of trees (`PARSE`) corresponding to a non-terminal level, i.e. syntactic or dependency analyses. This is a compromise between redundancy and ease of use.

Because various forms of linguistic processing depend on the output of other tools, we wanted to make this processing history explicit. We devised a multi-layer annotation scheme in which an XML `process` attribute refers to a description of the input (token or tree), the output, and the tool used. Figure 1 shows how a layer of `TOKEN` is built. This annotation allows for easy stacking of mark-up for tokenization, part-of-speech (POS) tags, base forms, named entities, syntactic trees etc. (Figure 3).

Figure 4 and Figure 5 show the current status of our annotation "pipe tree" on the token and sentence levels, respectively, as described below[7].

[1] *Nota bene:* despite the name, the strand of research we report here makes no claims as to the cognitive aspects of human reading comprehension (Levelt and Kelter, 1982).

[2] `http://acl.ldc.upenn.edu/W/W00/` in which results were reported by other groups working on this same corpus

[3] `http://www.clsp.jhu.edu/ws2000/groups/reading/`

[4] The contact person for the corpus is Lisa Ferro (address see Section 7).

[5] `http://www.cbc4kids.ca`

[6] Note that the gold standard for the question answering task are the "gold answers", not perfect linguistic annotations.

[7] We call it a "pipe tree" because it represents a set of "pipe lines" with common initial sub-steps.

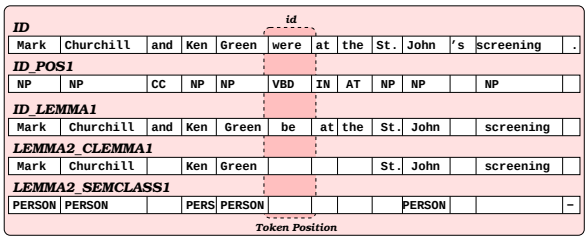


Figure 3: Multiple annotation layers.

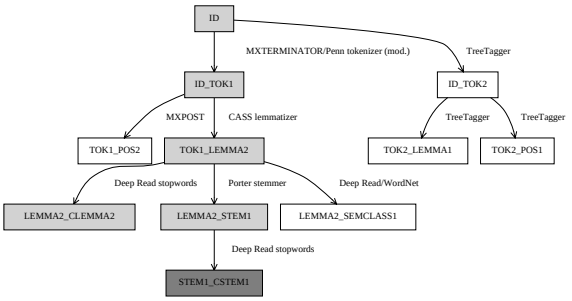


Figure 4: Annotation layers per token. The replicated Deep Read baseline system pipeline is highlighted.

Figure 2 gives an overview of our targeted annotation. A comprehensive description of the tools and structure can be found in the manual (Dalmas et al., 2003) distributed with the corpus.

The layers described here allow detailed comparisons of components' contribution for any question answering method by exploring different paths in the annotation "pipe tree".

We have implemented converters for all the tools listed (except the LTG tools, which output XML and hence do not need conversion) in Perl, and a master script that assembles the individual converters' output into a well-formed and valid XML document instance.

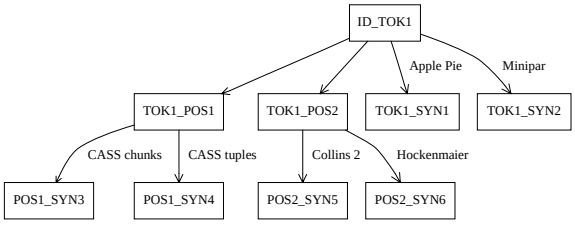


Figure 5: Annotation layers per sentence.

Difficulty	QC	R	P	AutSent	HumSent
Easy	237	0.74	0.18	0.75	0.74
Moderate	177	0.57	0.22	0.55	0.57
Difficult	67	0.49	0.19	0.43	0.43
Average	481	0.63	0.19	0.62	0.63

Table 1: Baseline evaluation using the STEM1_CSTEM1 layer according to question difficulty. QC is the number of questions.

This annotation is work in progress insofar as we are planning to include further layers featuring analyses of LT TTT, LT POS, LT CHUNK, named entity annotation using MITRE's Alembic (cf. Deep Read), the LTG MUC-7 system, as well as anaphora resolution software.

4 Baseline Results

This section describes our experiment replicating the baseline that was previously computed by Deep Read on the REMEDIA corpus, but here on the CBC4Kids data.

We began exploiting the `STEM1_CSTEM1` layer of our XML annotation scheme to get a baseline using stemmed lemmata of content words. The shaded path in Figure 4 shows these final layers we used and their ancestors in the linguistic pipeline, from token through lemma, stemming, stop-word removal, as in the Deep Read experiments.

We have implemented a batch QA system as a set of filters in the functional programming language Haskell.[8] The XML encoding of linguistic information greatly simplified the implementation part: the QA system was reduced to a program filtering a tree (the XML document containing story and questions) and computing intersection (overlap) on lists of tokens. Table 1 shows the results for the baseline using the `STEM1_CSTEM1` filter. The answers of the system are added to the XML file as a separate layer.

The evaluation metrics in Table 1 are the same as described in (Hirschman et al., 1999), namely **Recall**, **Precision**, **AutSent** and **HumSent**:[9]

[8] `http://www.haskell.org`

[9] *Cave lector:* The definitions for P and R in (Hirschman et al., 1999) appear to have been swapped.

Type	Tool	Process ID	Reference
Sentence Boundaries	MXTERMINATOR	ID	
Tokenization	Penn tokenizer.sed	ID_TOK1	
	Tree-Tagger (internal)	ID_TOK2	(Schmid, 1994)
	LT TTT	ID_TOK3	(Grover et al., 2000)
Part-of Speech	MXPOST	TOK1_POS2	(Ratnaparkhi, 1996)
	Tree-Tagger	TOK2_POS1	(Schmid, 1994)
	LT POS	TOK3_POS3	(Mikheev et al., 1999)
Lemmatization	CASS "stemmer"	TOK1_LEMMA2	(Abney, 1996)
	Tree-Tagger	TOK2_LEMMA1	(Schmid, 1994)
	morpha	POS1_LEMMA3	(Minnen et al., 2001)
Stemming	Porter stemmer	LEMMA2_STEM1	(Porter, 1980)
Stop-Word Filtering	Deep Read	LEMMA2_CLEMMA2	(Hirschman et al., 1999)
	Deep Read	STEM1_CSTEM1	(Hirschman et al., 1999)
Syntactic Analysis	Apple Pie Parser	POS2_SYN1	(Sekine and Grishman, 1995)
	Minipar relations	TOK1_SYN2	(Lin, 1998)
	CASS chunk trees	POS1_SYN3	(Abney, 1996)
	CASS dependency tuples	POS1_SYN4	(Abney, 1997)
	Collins parse trees	POS2_SYN5	(Collins, 1997)
	CCG parse trees	POS2_SYN6	(Hockenmaier and Steedman, 2002)
	LT CHUNK	POS3_SYN7	(Mikheev et al., 1999)
Named Entity Tagging	Deep Read (WordNet)	LEMMA2_SEMCLASS1	(Hirschman et al., 1999)
	MITRE Alembic	TOK1_NE1	(Aberdeen et al., 1995)
	LTG MUC-7	SYN7_NE2	(Mikheev et al., 1998)
Anaphora Resolution	*N.N.*	SYN5_AR1	*N.N.*

Figure 2: Annotation tools: Targeted list of layers.

Question Type	R	P	AutSent	HumSent
when	0.71	0.15	0.76	0.76
who/-se/-m	0.68	0.16	0.67	0.71
how	0.71	0.21	0.70	0.70
how many/much	0.62	0.08	0.63	0.67
what	0.66	0.26	0.63	0.65
which_np	0.70	0.08	0.60	0.60
where	0.58	0.14	0.56	0.56
how_att	0.56	0.15	0.56	0.56
what_np	0.59	0.18	0.56	0.56
why	0.57	0.23	0.52	0.51

Table 2: Baseline evaluation (STEM1_CSTEM1) according to question type.

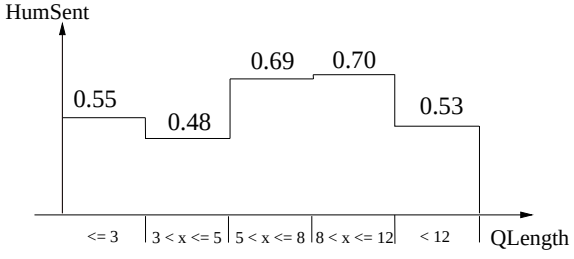


Figure 6: HumSent accuracy by length of question bag (STEM1_CSTEM1). The average bag length for QLength $\geq$ 12 is 14 words, with a maximum of 19 words.

R	=	$\lvert cw_{sa} \cap cw_{ha} \rvert \;/\; \lvert cw_{ha} \rvert$
P	=	$\lvert cw_{sa} \cap cw_{ha} \rvert \;/\; \lvert cw_{sa} \rvert$
AutSent	=	$\#[\; sentence \mid \mathrm{R}(sentence) > 0 \;]$
HumSent	=	*list of sentences considered as answers by a human annotator*
cw	:	content words
sa	:	system answer
ha	:	human answer (a phrase).

Sentences containing the answer picked by human and machine, respectively, are also marked up in XML. We have developed an automated evaluation program that can currently take into account three parameters: the difficulty of the answer (as annotated in the original CBC4Kids release, see below), the question type (based on the WH-word) and the length of the question bag. Table 2 and Figure 6 show some of the results.

5 Discussion

As already noted, the questions constructed for the CBC4Kids corpus are rated as to their difficulty (Ferro, 2000):

> "Easy: Uses exact wording from the text and/or the question and answer are

close to each other in the text. [...] Moderate: Some paraphrasing from the text and/or the question and answer aren't close to each other in the text. [...] Difficult: Very or entirely different words are used in question; lots of other tempting but incorrect answers are in the story; subtle knowledge is required to answer the question."

Table 1 shows the performance of the baseline system, broken down by difficulty class. For all scoring metrics other than Precision (P), the table shows a strong correlation between the retrieval score and the class assigned according to Ferro's guidelines for Q&A writing. As for Precision, it is not really significant because human answers are phrases and our system outputs a sentence as answer. However, Precision allows us to see from Table 2 that very short answers are expected for HOW_MANY, HOW_MUCH and WHICH_NP questions. This is not surprising for HOW_MANY or HOW_MUCH questions, for which expected answers are very short named entities (*How many people?* $\rightarrow$ *twenty-five*). But for WHICH_NP questions, they are in fact expecting a named entity and especially a proper name (*In which city / Which two African leaders / Which U.S. states*). The length of the expected answer is not so obvious for other questions that expect named entities, such as WHEN questions. The main reason for this is that the corpus itself asks for a story comprehension and not for general answers as in the TREC evaluation. For example, the following WHEN question *When did Wilson climb onto the second-floor balcony?* expects a long answer: *when he heard the cries of Westley, Hughes, and their children*.

As already noted by Hirschman and co-workers for Deep Read, the Recall (R) and HumSent metrics behave in a similar manner. But here for WHY and WHICH_NP questions, we notice a significant difference: generally these questions contain one or two words repeated all along the story (name of the main character for instance) and therefore the possibility of a tie between possible answers becomes more important. This is particulary true when the question bag is either short (between 3 and 5 words) or very long (more than 12 words, see Figure 6).

Since an answer occurs generally only once in a story, we cannot rely on techniques using redundancy. But the advantage of a short text is also that deeper NLP techniques can be used appropriately.

We obtain significantly higher Recall scores for CBC4Kids compared to Deep Read's performance on the REMEDIA corpus, although the language used in the latter is targeted at a much younger age group. Independent experiments at MITRE have also yielded higher performance scores for CBC4Kids.[10]

One possible explanation for the overall higher scores is that the CBC4Kids questions were composed with a NLQA system in mind: for instance, question authors were told to avoid anaphoric references in the questions (Ferro, 2000), which are quite frequent in the REMEDIA questions. Another possible explanation is that the shorter sentence length due to the younger audience fragments information across sentences, thus decreasing term overlap at the given sentence granularity.[11] It remains to be investigated how much the purpose of text production impacts reading comprehension simulation results, as the REMEDIA text and questions were not authored with an informative purpose in mind. In the CBC4Kids case, the text was pre-existing and created with informative intent, but the questions were created a posteriori; hence both methods are artificial, but in different ways.

It is quite easy to carry out an error analysis once the results of the system have been encoded in XML. A simple XSL stylesheet can be sufficient for extracting questions and answers we want to analyse (Figures 7 and 8).

6 Future Work

This section describes some of the experiments we have planned for the future. These are likely to require adding further layers with linguistic annotation.

6.1 Towards Predicate/Argument Structure

Surface overlap metrics are intrinsically limited, since they cannot, for instance, distinguish be-

[10] Ben Wellner, personal communication.

[11] Lisa Ferro, personal communication.

tween *man bites dog* and *dog bites man*—they are a-semantic in nature. To overcome this, we are planning to utilize the various syntactic representations (cf. Figure 2) to obtain predicate-argument structures (`bite(man, dog)` versus `bite(dog, man)`), which allow for higher precision. One path of investigation is to induce the grammar underlying the corpus, to filter the top-*n* most likely productions and to subsequently add semantic composition rules manually. Another path worthwhile exploring is learning the mappings from chunks to predicate-argument structures in a supervised regime. Once we have more robust methods of predicate-argument structures, we will be able to explore shallow inferences for NLQA (Webber et al., 2002) in the controlled environment that CBC4Kids provides.

One path of investigation is to induce the grammar underlying the corpus, to filter the top-n most likely productions and to subsequently add semantic composition rules by hand. Another path worthwhile exploring is learning the mappings from chunks to Quasi-Logical Forms (QLF) in a supervised regime.

Once we have more robust methods of QLF extraction, we will be able to explore shallow inferences for NLQA (Webber et al., 2002) in the controlled environment that CBC4Kids provides.

6.2 Comparing Answers

Each question in the CBC4Kids corpus receives at least one answer determined by a human annotator. We would like to use this rich annotation to begin a study on detecting multiple answer cases which is part of the current roadmap for research in NLQA in the TREC community (Burger et al., 2001).

Few have so far proposed to consider the evaluation of NLQA systems retrieving complex answers, but recently (Buchholz and Daelemans, 2001) and (Webber et al., 2002) have suggested different classification sets for comparing answers. This would allow NLQA systems to provide multiple answers linked together by labels expressing their relationship, such as "P implies Q", "P and Q are equivalent", "P and Q are alternative answers" (exclusiveness), "P and Q provide a collective answer" (complementarity), and others (Webber et al., 2002).

One goal of this thread of research is to build a practical framework for evaluation multiple answers that allows answer comparison.

7 Conclusions

We have described the process of creating rich annotation of the CBC4Kids corpus of news for children. The chosen XML annotation architecture is a compromise that allows for multilayer annotation whilst simplifying the integration of added linguistic knowledge from heterogeneous toolsets. The architecture reduces many applications to a sequence of selections and functional mappings over the annotation layers. The application of such a scheme is by no means restricted to the corpus under consideration; we intend to reuse it, notably for textual resources from the biomedical domain.

On the basis of the resulting dataset, CBC4Kids, we have replicated an evaluation performed by (Hirschman et al., 1999), but on the CBC4Kids corpus. This will serve as a basis for our future experiments involving robust semantic construction and inference for question answering.

We do not know of any other corpus that has been automatically annotated with comparably rich strata of linguistic knowledge and believe that the corpus can be a valuable resource also for other NLQA research groups.

The corpus is distributed by MITRE, with layers as given above, including answers given by our system for the Deep Read baseline. Please contact Lisa Ferro directly for a copy.[12]

Acknowledgments. We are grateful to Lynette Hirschman and Lisa Ferro at MITRE, who provided us with the initial CBC4Kids corpus. We would also like to thank the authors of all the tools mentioned and used in this paper for making them available to the academic community. Thanks to Julia Hockenmaier, Maria Lapata, Dekang Lin, Katja Markert, Satoshi Sekine and Bill Wellner and three anonymous reviewers for helpful advice and feedback.
We would like to acknowledge the financial support of the German Academic Exchange Service (DAAD) under grant D/02/01831, of Linguit

[12] `lferro@mitre.org`

GmbH (research contract UK-2002/2), and of the School of Informatics, University of Edinburgh.

References

J. Aberdeen, D. Day, L. Hirschman, P. Robinson, and M. Vilain. 1995. MITRE: Description of the Alembic system used for MUC-6. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 141–155.

S. Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344.

S. Abney. 1997. Part-of-speech tagging and partial parsing. *Corpus-Based Methods in Language and Speech Processing*.

S. Buchholz and W. Daelemans. 2001. Complex answers: A case study using a WWW question answering system. *Journal of Natural Language Engineering*, 7:301–323.

J. Burger, C. Cardie, V. Chaudhri, S. Harabagiu, D. Israel, Chr. Jacquemin, C.-Y. Lin, S. Mariorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees, and R. Weischedel. 2001. Issues, tasks and program structures to roadmap research in question and answering. *NIST*.

E. Charniak. 1972. *Toward a Model of Children's Story Comprehension*. Ph.D. thesis, Massachusetts Institute of Technology.

M. J. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL*, Madrid. Association for Computational Linguistics.

S. Cotton and S. Bird. 2002. An integrated framework for treebanks and multilayer annotations. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

T. Dalmas, J. L. Leidner, B. Webber, C. Grover, and J. Bos. 2003. *Annotating CBC4Kids: A Corpus for Reading Comprehension and Question Answering Evaluation. (Technical Report)*. School of Informatics, University of Edinburgh.

L. Ferro. 2000. *Reading Comprehension Tests: Guidelines for Question and Answer Writing. (Unpublished Technical Report)*. The MITRE Corporation.

C. Grover, C. Matheson, A. Mikheev, and M. Moens. 2000. LT TTT–a flexible tokenisation tool. In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC 2000)*.

L. Hirschman, M. Light, E. Breck, and J. D. Burger. 1999. Deep Read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

J. Hockenmaier and M. Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*.

W. J. M. Levelt and S. Kelter. 1982. Surface form and memory in question answering. *Cognitive Psychology*, 14:78–106.

D. Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*.

A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*.

A. Mikheev, C. Grover, and M. Moens. 1999. XML tools and architecture for named entity recognition. *Journal of Markup Languages: Theory and Practice*, 3:89–113.

G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing of English. *Journal of Natural Language Engineering*, 7(3):207–223.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. *International Conference on New Methods in Language Processing*.

S. Sekine and R. Grishman. 1995. A corpus-based probabilistic grammar with only two non-terminals. In *Proceedings Fourth International Workshop on Parsing Technologies*.

E. Voorhees and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In E. M. Voorhees and D. K. Harman, editors, *The Eighth Text REtrieval Conference (TREC 8)*, NIST Special Publication 500–246, pages 1–24, Gaithersburg, Maryland, November 17-19, 1999. National Institute of Standards and Technology.

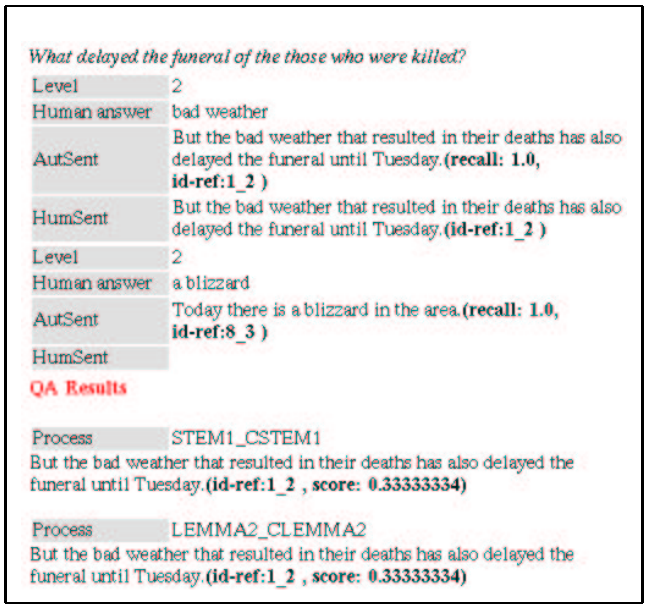
B. L. Webber, C. Gardent, and J. Bos. 2002. Position statement: Inference in question answering. In *Proceedings of the LREC Workshop on Question Answering: Strategy and Resources*, Las Palmas, Gran Canaria, Spain.

A Sample Story from CBC4Kids

Tragedy Strikes a Northern Village
January 4, 1998

The six hundred mostly Inuit residents of the northern Quebec village of Kangiqsualujjuaq had planned to bury the bodies of nine of their friends and children in a funeral this afternoon. But the bad weather that resulted in their deaths has also delayed the funeral until Tuesday.
Kangiqsualujjuaq* is about 1,500 kilometres north of Montreal, at the mouth of the George River on Ungava Bay. This region is known as Nunavik.
An avalanche hit the town's Satuumavik school gymnasium in the Northern Quebec community early Friday morning.
[...]
Principal Jean Leduc said an inquiry commissioned by the local school board after the earlier avalanche had recommended that fences be built. The fences were never built. Speculation on the cause of the avalanche centered on a ceremonial gun salute at midnight, 90 minutes before the snow crashed in. Villagers wondered if the shots set in motion vibrations that eventually caused the avalanche, while others wondered if music from the dance had played a role.
Police and avalanche experts will travel to the village to investigate the tragedy. Quebec Premier Lucien Bouchard announced there will be a full public inquiry into the disaster.

Questions
How far is Kangiqsualujjuaq from Montreal?
When did the avalanche hit the school?
Where was Mary Baron when the avalanche hit?
How many people were seriously injured by the avalanche?
What delayed the funeral of the those who were killed?
What could have possibly prevented the tragedy?
Who will investigate the tragedy?


What delayed the funeral of the those who were killed?

Level	2
Human answer	bad weather
AutSent	But the bad weather that resulted in their deaths has also delayed the funeral until Tuesday.**(recall: 1.0, id-ref:1_2)**
HumSent	But the bad weather that resulted in their deaths has also delayed the funeral until Tuesday.**(id-ref:1_2)**
Level	2
Human answer	a blizzard
AutSent	Today there is a blizzard in the area.**(recall: 1.0, id-ref:8_3)**
HumSent	

QA Results

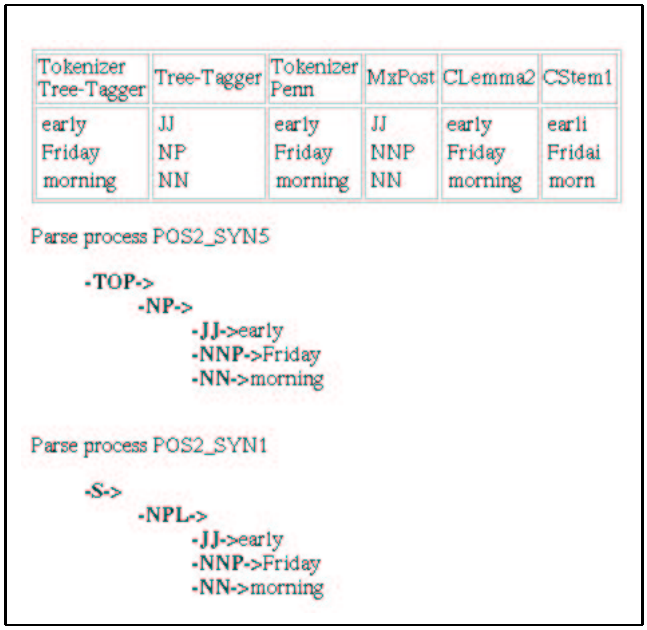
Process	STEM1_CSTEM1

But the bad weather that resulted in their deaths has also delayed the funeral until Tuesday.**(id-ref:1_2 , score: 0.33333334)**

Process	LEMMA2_CLEMMA2

But the bad weather that resulted in their deaths has also delayed the funeral until Tuesday.**(id-ref:1_2 , score: 0.33333334)**

Figure 7: HTML view for a question. The *score* given for each answer corresponds to the overlap between a candidate answer sentence and the question (WdAnsRecall).


Tokenizer Tree-Tagger	Tree-Tagger	Tokenizer Penn	MxPost	CLemma2	CStem1
early Friday morning	JJ NP NN	early Friday morning	JJ NNP NN	early Friday morning	earli Fridai morn

Parse process POS2_SYN5

-TOP->
-NP->
-JJ->early
-NNP->Friday
-NN->morning

Parse process POS2_SYN1

-S->
-NPL->
-JJ->early
-NNP->Friday
-NN->morning

Figure 8: Partial HTML view of linguistic layers for a human answer.