School of Informatics, University of Edinburgh

Institute for Communicating and Collaborative Systems

Annotating CBC4Kids: A Corpus for Reading Comprehension and Question Answering Evaluation

by

Tiphaine Dalmas, Jochen Leidner, Bonnie Webber, Claire Grover, Johan Bos

Informatics Research Report EDI-INF-RR-0204

School of Informatics **March 2004**
http://www.informatics.ed.ac.uk/

Annotating CBC4Kids: A Corpus for Reading Comprehension and Question Answering Evaluation

Tiphaine Dalmas, Jochen Leidner, Bonnie Webber, Claire Grover, Johan Bos

Informatics Research Report EDI-INF-RR-0204

SCHOOL *of* INFORMATICS
Institute for Communicating and Collaborative Systems

March 2004

Abstract :

Reading comprehension tests are receiving increased attention within the NLP community as a controlled test-bed for developing, evaluating and comparing robust question answering (NLQA) methods.

To support this, we have enriched the MITRE CBC4Kids corpus with multiple XML annotation layers recording the output of various tokenizers, lemmatizers, a stemmer, a semantic tagger, POS taggers and syntactic parsers. To demonstrate its use, we have built a baseline NLQA system for word-overlap based answer retrieval, NLQA evaluation and corpus browsing.

Keywords : , Computer corpora; linguistic XML annotation; natural language question answering (NLQA; QA); news stories for children; natural language understanding (NLU);

School of Informatics, University of Edinburgh

Annotating CBC4Kids: A Corpus for Reading Comprehension and Question Answering Evaluation

by

Tiphaine Dalmas
Jochen L. Leidner
Bonnie Webber
Claire Grover
Johan Bos

Informatics Research Report

School of Informatics **March 2004**
http://www.informatics.ed.ac.uk/

Annotating CBC4Kids: A Corpus for Reading Comprehension and Question Answering Evaluation

Tiphaine Dalmas
Jochen L. Leidner
Bonnie Webber
Claire Grover
Johan Bos

Informatics Research Report

SCHOOL *of* INFORMATICS

March 2004

Abstract

Reading comprehension tests are receiving increased attention within the NLP community as a controlled test-bed for developing, evaluating and comparing robust question answering (NLQA) methods.
To support this, we have enriched the MITRE CBC4Kids corpus with multiple XML annotation layers recording the output of various tokenizers, lemmatizers, a stemmer, a semantic tagger, POS taggers and syntactic parsers. To demonstrate its use, we have built a baseline NLQA system for word-overlap based answer retrieval, NLQA evaluation and corpus browsing.

Keywords : Computer corpora; linguistic XML annotation; natural language question answering (NLQA; Q&A); news stories for children; natural language understanding (NLU); reading comprehension by machine

GUID : 947794e6-bcec-4dee-84d0-f175c1973c58 (Globally Unique Identifier of this document)

Contents

List of Figures

List of Tables

1 Introduction

Linguistic corpora marked up with XML represent the state of the art in language engineering. Recently, reading comprehension tests have received increased attention for testing Question Answering methods. We present our ongoing project to develop a re-usable resource for reading comprehension and Natural Language Question Answering (NLQA) that we hope will be useful as a controlled test-bed for developing and evaluating robust NLQA methods. Starting from MITRE's CBC4Kids corpus collection [Breck et al.2001], we have created a practical multi-layer annotation scheme and added various strata of linguistic annotation automatically using state-of-the-art NLP tools. This paper presents the architecture, the various tool-sets we used and the distributed development scenario we worked in. We also describe how the chosen multi-layer scheme naturally leads to a simple implementation of a baseline question answering system and an evaluation program.

2 Automatic Linguistic Annotation

2.1 Distributed XML development scenario

Our distributed development scenario is shown in Figure 1. A normalization phase of the corpus produces valid XML. After this, development team members each applied the same process to each NLP tool assigned whose output was desired as an annotation layer: a wrapper was created to convert XML into the tool's input format, and another wrapper to convert the tool's output back into a well-formed XML stratum that could be inserted in the XML stream on the fly. This distributed form of collaboration easily scales up to larger development teams, where individual team members are free to choose different implementation languages and glue mechanisms. The final document instance trees were then merged. While a generic XSLT tree union script can be used for this, we instead defined one tree to be the master instance and added all new subtrees present in the second instance. The result was validated against the DTD and transformed further.

2.2 Design principles

Our work is driven by the following observation [Cotton and Bird2002]: "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 original CBC4Kids corpus was developed at MITRE[1], based on a collection of newspaper stories for teenagers written for CBC's Web site[2]. To each article selected for inclusion in the corpus, the MITRE group added a set of 8-10 questions of various degrees of difficulty. The corpus also includes one or more answers for each question, in the form of a disjunction of one or more phrases (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 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 the replication of a baseline system, and 3) quality of the automatic annotation. Because most available tools (with the exception of LT TTT, [Grover et al.2000]) do not output XML, we had to develop a set of converters.

2.3 Linguistic layers

Each sentence has three different representations: 1) the original string, 2) a list of tags labeled `TOKEN` encoding the results from linguistic tools that output lexical information (POS tags, stems, etc.), 3) a list of trees (`PARSE`s) corresponding to analyzes at a non-terminal level, i.e. syntactic or dependency graphs. This is a compromise between minimizing redundancy and maximizing ease of use. In particular, there is no link between token positions and the corresponding occurrences of words in the parse trees/dependency graphs. Any annotation scheme with a tighter coupling would require an alignment step which, in many cases, would have to remain incomplete due to idiosyncrasies of the tools: for instance, a parser that used its own built-in tokenization might yield a different number of tokens from `tokenizer.sed`.[3]

Because various forms of linguistic processing depend on the output of other tools, we wanted to make the 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 2 shows how a `TOKEN` layer 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 4). The word-form token from Figure 2 are then repeated in the `PARSE` trees (`<LEAF type="Scotia"/>`).

Figure 5 and Figure 6 show the current status of our annotation "pipe tree" for tokens and trees/graphs, respectively, as described below.[4] Figure 3 gives an overview of our current annotation layers.

[1] The contact person for the corpus is Lisa Ferro (`lferro@mitre.org`).

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

[3] Treatment of *doesn't* as *does n't* is but one example.

[4] We call it a "pipetree" because it represents a set of "pipelines" (like UNIX pipes) with common initial sub-steps.

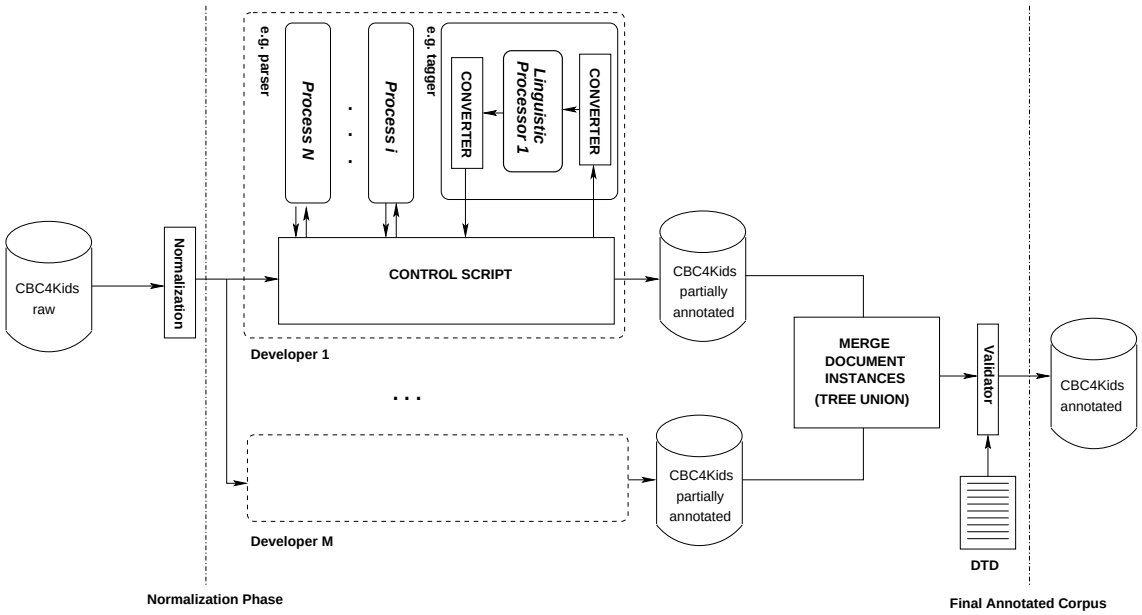


Figure 1: Building a richly annotated corpus in a distributed XML scenario.

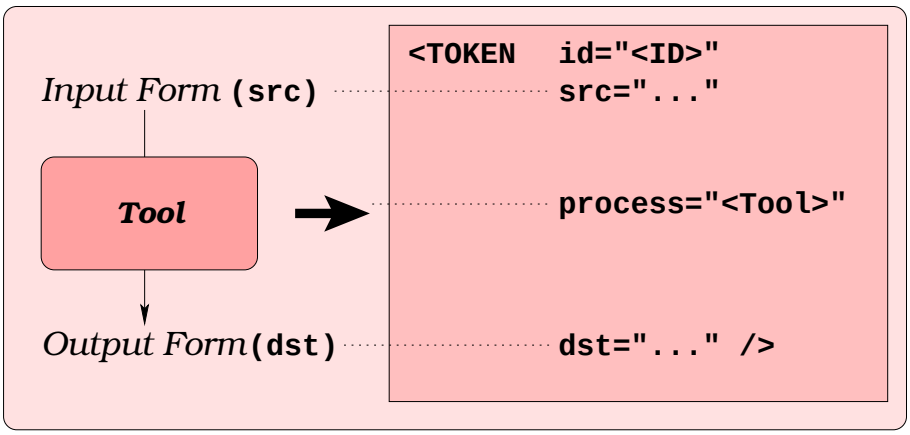


Figure 2: Building a new layer of `TOKEN` tags.

Type	Tool	Process ID	Reference
Sentence Boundaries	MXTERMINATOR	ID	[Ratnaparkhi1996]
Tokenization	Penn tokenizer.sed	ID_TOK1	
	Tree-Tagger (internal)	ID_TOK2	[Schmid1994]
	LT TTT	ID_TOK3	[Grover et al.2000]
Part-of Speech	MXPOST	TOK1_POS2	[Ratnaparkhi1996]
	Tree-Tagger	TOK2_POS1	[Schmid1994]
	LT POS	TOK3_POS3	[Mikheev et al.1999a]
Lemmatization	CASS 'stemmer'	TOK1_LEMMA2	[Abney1996]
	Tree-Tagger	TOK2_LEMMA1	[Schmid1994]
	morpha	POS1_LEMMA3	[Minnen et al.2001]
Stemming	Porter stemmer	LEMMA2_STEM1	[Porter1980]
Stop-Word Filtering	Deep Read	LEMMA2_CLEMMA2	[Hirschman et al.1999]
	Deep Read	STEM1_CSTEM1	[Hirschman et al.1999]
Named Entity Tagging	Deep Read (WordNet)	LEMMA2_SEMCLASS1	[Hirschman et al.1999]
Syntactic Analysis	Apple Pie Parser	POS2_SYN1	[Sekine and Grishman1995]
	Minipar relations	TOK1_SYN2	[Lin1998]
	CASS chunk trees	POS1_SYN3	[Abney1996]
	CASS dependency tuples	POS1_SYN4	[Abney1996]
	Collins parse trees	POS2_SYN5	[Collins1997]

Figure 3: Annotation tools: Layers in Release 1.

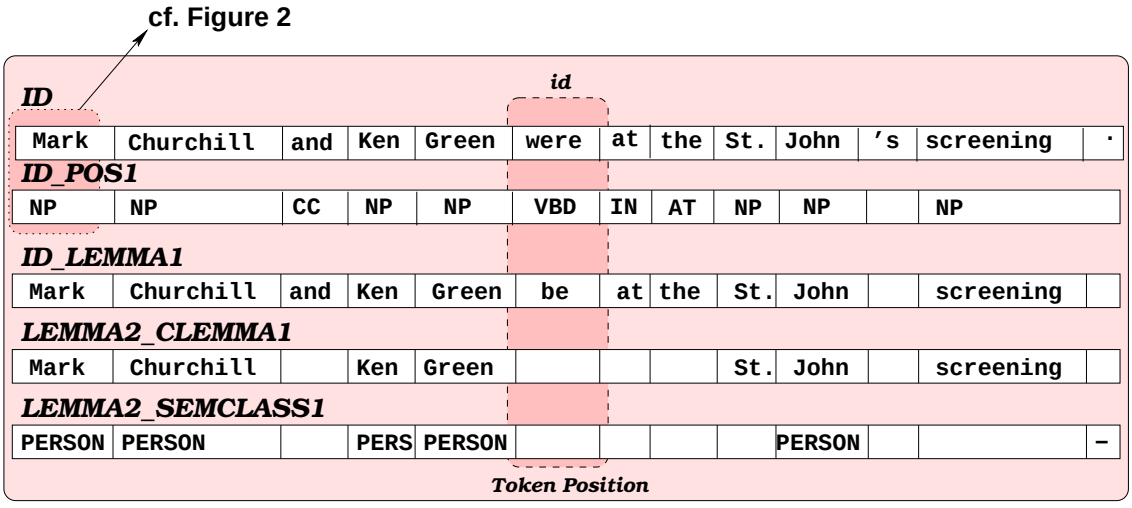


Figure 4: Multiple annotation layers.

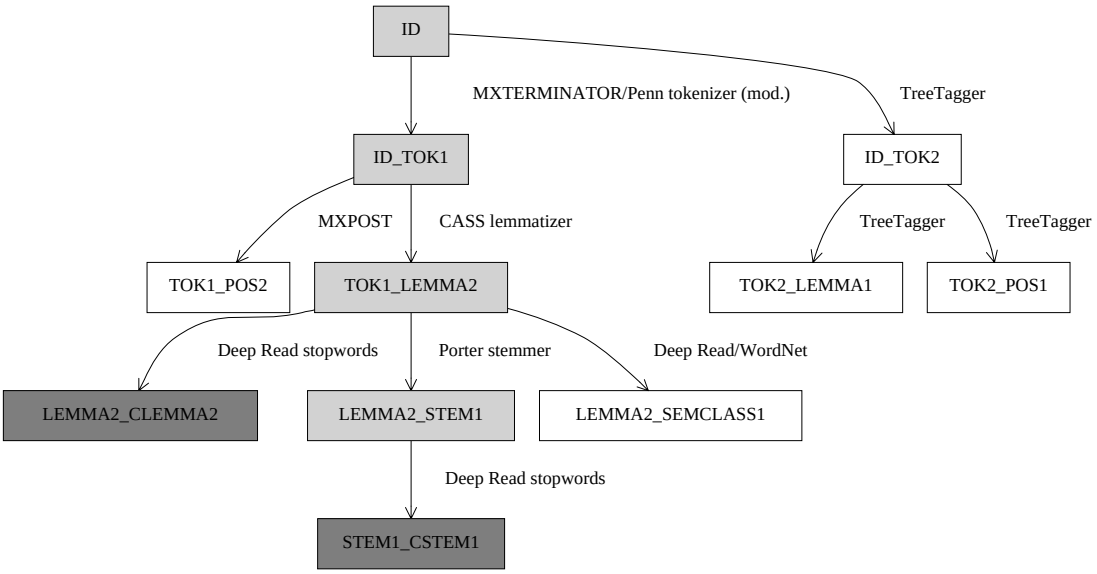


Figure 5: Annotation layers per token. The replicated Deep Read baseline system pipeline is highlighted.

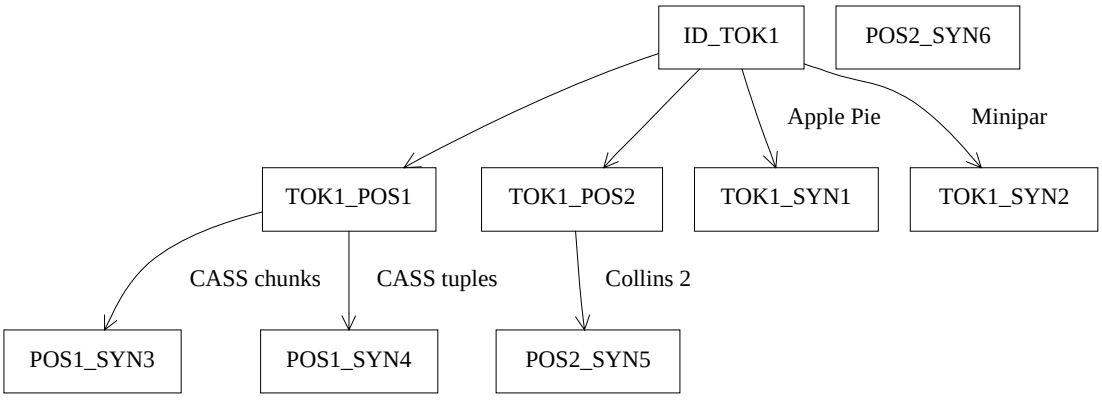


Figure 6: Annotation layers per sentence.

2.4 Normalization and annotation of the corpus

The original CBC4Kids corpus from MITRE comes marked up with XML-like tags. After authoring the Document Type Definition (DTD), processing comprised two steps (Figure 1): First, we normalized the corpus so as to make sure the data consistently fits the form described by our DTD. Because the minimal scheme requires sentence boundary detection (cf. Chapter 2.5) and the original CBC4Kids corpus only contained markup for paragraphs, normalization also involved splitting each paragraph into a list of sentences. Secondly, we enriched the corpus with linguistic annotation layers. Annotation layers are optional in our DTD. Each annotation layer is added by a program taking an XML file as input and outputting another XML file containing the additional layer; Figure 2 shows the internal process.

2.5 Description of the layers

Sentence boundary detection. We used MXTERMINATOR [Ratnaparkhi1996] to split each paragraph into sentences.[5]. Questions and human answers are already demarcated in the source CBC corpus released by MITRE.

Tokenization. Each sentence was tokenized using the Penn Treebank tokenizer, a **sed(1)** script by Robert MacIntyre (University of Pennsylvania)[6]. We modified it slightly before running it on the corpus so as to recognize number separators, URLs, and intra-sentential quotations which were characteristic of the CBC corpus. The resulting token sequence was defined as process `ID_TOK1`.

POS Tagging. We recorded the results of two POS taggers for comparison purpose: TreeTagger and MXPOST. TreeTagger [Schmid1994] is a POS tagger based on decision tree induction. Trained and tested on a Penn Treebank sample, it has a reported accuracy of 96.34% (trigram maximal window size). The POS tags of TreeTagger define our layer `TOK2_POS1` (TreeTagger comes with a built-in tokenizer). MXPOST is a POS tagger based on the Maximum Entropy framework that has a reported accuracy of 96.6% on Wall Street Journal text [Ratnaparkhi1996]. We have added a token layer `TOK1_POS2` based on MXPOST.

Lemmatization. `TOK1_LEMMA1`, our first token layers of lemmata, is provided by TreeTagger. Additionally, we obtained a second base-form layer `TOK1_LEMMA2` using the rule-based program `stemmer` from the CASS software distribution [Abney1996].[7]

Stop-Word Filtering. We used the same stop-words set as described in the Deep Read baseline reported in [Hirschman et al.1999].

Stemming. Porter describes a simple stemmer for English [Porter1980]. We applied it to our texts and questions and made the output available as layer `LEMMA2_STEM1`.

Semantic Classes. Deep Read [Hirschman et al.1999] uses WordNet to check words for subsumption of the synsets PERSON and/or LOCATION. We have integrated the result of this lookup as layer named `LEMMA2_SEMCLASS1`.

Parsing. `PARSE` tags record the output of several different parsers that we have included in our pipe trees.

The Apple Pie Parser (APP) is a statistical parser trained on the Wall Street Journal subset of the Penn Treebank [Sekine and Grishman1995]. It comprises only non-terminals for NP and S and parses by recombination of NP/S-fragments, memo-izing the complete training set. On unseen WSJ material, it has been reported to achieve a labeled precision[8] of 72.61% and 83% for sentences up to 15 words). APP returns a single best tree, which we have incorporated as process `POS2_SYN1`.

[5] `http://www.cis.upenn.edu/ãdwait/statnlp.html`

[6] `http://www.cis.upenn.edu/treebank/tokenization.html`

[7] Despite the name, `stemmer` is a lemmatizer rather than a stemmer.

[8] The reported numbers for parse tree evaluation refer to the PARSEVAL.

Minipar [Lin1998] is a rule-based parser that implements a procedural model of Government & Binding (GB) realized as message passing; it is derived from Principar [Lin1995], incorporates knowledge about named entities and comprises a lexicon ≈90k lemmata. Its output consists of dependency relations over word token positions. It has a reported labeled dependency precision of 88.54%. This is the process `TOK1_SYN2`.

Shallow processing techniques have emerged as an efficient way to deal with large quantities of text. 'Chunking' – partial parsing by iterative bottom-up bracketing using multi-layer deterministic finite-state transducers for non-recursive noun/verb groups ('chunks') – has been described by Abney [Abney1996] and implemented in his CASS. The chunker outputs either trees or dependency relation tuples.[9] We define a layer `POS1_SYN3` with trees of CASS chunks and `POS1_SYN4` with the dependency tuples.

[Collins1997] presents three statistical, lexicalized parsing models. We chose his model 2 (which models left and right dependents), for integration as layer `POS2_SYN5`. The POS2 layer is used as input because Collins' parser uses MXPOST POS tags for handling unknown words. Collins reports 88.35% labelled precision for this model on sentences with less than 40 words. The average sentence length in CBC4Kids is 18 words (maximum 57).

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

We have implemented converters for all the tools listed in Perl, and a master script that assembles the individual converters' output into a well-formed and valid XML document instance (implemented in Haskell).

The annotation should be seen as a continuing process: other groups working with the corpus are encouraged to add their own layers and submit their enhancements to Lisa Ferro at MITRE.

[9] Since the latter output format is based not on token-position but on the surface string of the region, there is a potential for ambiguity if a surface string occurs multiple times in the same sentence.

3 Building NLQA Systems as Set of XML Filters

This section describes the architecture of a simple question answering system we built on top of the annotated CBC4Kids corpus. We built it to transfer the baseline results from the word overlap method used by the Deep Read system in connection with the REMEDIA corpus [Hirschman et al.1999] to the annotated CBC4Kids data and to support our investigation of more sophisticated methods.

3.1 Baseline System

We exploited our XML annotation scheme using the `STEM1_CSTEM1` and `LEMMA2_CLEMMA2` layers for a baseline based on content stems and content lemmata, respectively. The layer is a parameter, so any–even a user-defined–layer may be used with our existing implementation. Figure 5 shows these final layers we used and their ancestors in the linguistic pipeline. We have implemented a batch NLQA system as a set of filters in the functional programming language Haskell.[10]

The XML markup of linguistic information greatly simplified the implementation part: the NLQA system was reduced to a function filtering a tree (the XML document containing story and questions) and computing intersection (overlap) on lists of tokens. Figure 7 shows the root of the XML tree structure of a CBC4Kids document. A document (`DOC`) instance comprises the story and the associated set of questions and answers. Question Answering is reduced to selecting a desired layer and intersecting the bags of tokens associated with questions and answers, respectively. Sentences without any overlap are filtered out. Sentences which do overlap with the question are sorted according to the size of the overlap. These answers are added to the XML file as a separate layer.

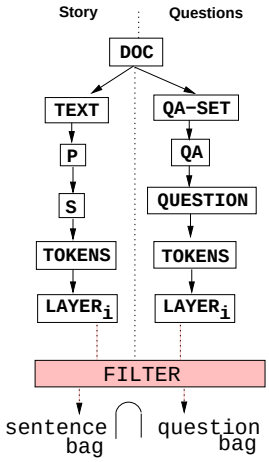


Figure 7: QA as intersection of analyzed question and analyzed text. White boxes represent XML layers.

LAYER$_i$ are the annotation layers (STEM1_CSTEM1, LEMMA2_CLEMMA2 etc).

3.2 Baseline Results

The evaluation metrics in Table 1 are the same as described in [Hirschman et al.1999], namely **Recall**, **Precision**, **AutSent** and **HumSent**:[11]

[10] For an introduction, see [Thompson1999] or `http://www.haskell.org/`.

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

Recall	=	$\|cw_{sa} \cap cw_{ha}\| \; / \; \|cw_{ha}\|$
Precision	=	$\|cw_{sa} \cap cw_{ha}\| \; / \; \|cw_{sa}\|$
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).

We automatically computed **AutSent** as defined by Hirschman et al. and added it as a layer in the corpus. The **AutSent** layer is the set of sentences that overlap with the answer key.

We manually annotated the corpus with **HumSent** that correspond to the answer key. (Answer keys are not always extracted from a sentence but may rephrase part of it).

Both **HumSent** and **AutSent** information were added as XML layers for evaluation. Figure 8 is an HTML view of a question with the evaluation layers and the results from the baseline. The figure shows two answers, derived from two different processes, one using stemming, whereas the other one relies on lemmatization instead. The score (recall) indicated is 0.4 for both cases, because two words from the question were matched in the answers (*Madonna* and *music* (stemmed form of *musical*) in the first process, *music* and *business* in the second process).

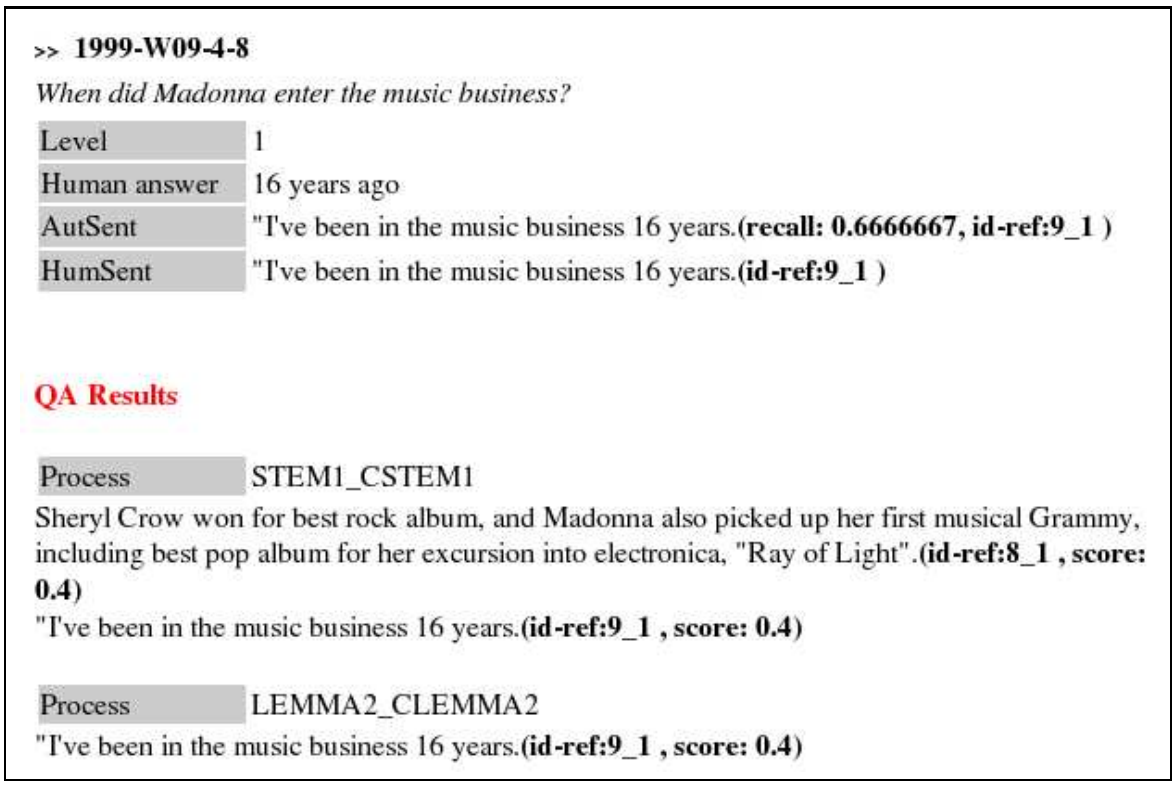


Figure 8: Evaluation and result layers for a question.

We 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 9 show some of the results.

Difficulty	# questions	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.

As already noted, the questions constructed for the CBC4Kids corpus are rated as to their difficulty [Ferro2000]:

> "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. Precision is not really significant because human answers are phrases and the 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*.

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.

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 particularly true when the question bag is either short (between 3 and 5 words) or very long (more than 12 words, see Figure 9).

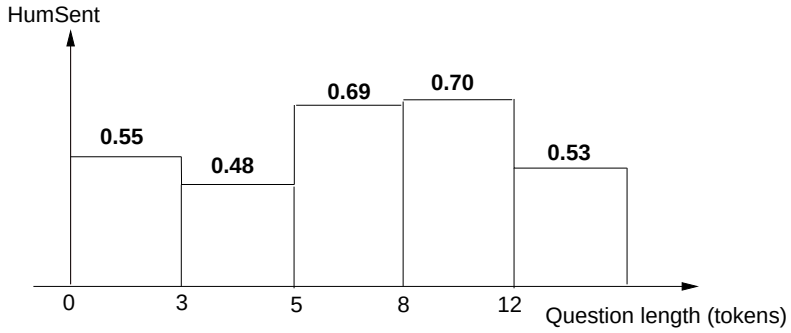


Figure 9: HumSent accuracy by length of question bag (STEM1_CSTEM1). The average bag length for QLength $\geq$ 12 is 14 words, with a maximum of 19 words.

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.[12]

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 [Ferro2000], 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.[13] 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.

[12] Ben Wellner, personal communication.
[13] Lisa Ferro, personal communication.

4 Related Work

Corpus annotation. Some other corpora have multiple annotation layers. For example, [Grover et al.submitted] use five linguistic layers for their DISP corpus of biomedical abstracts in order to compare the relative utility of shallow versus deep parsing in analyzing nominal compounds.

XML alternatives. In the TIPSTER project [NISTonline], a different architecture for text processing was proposed that does not make use of XML: a TIPSTER-compliant application annotates text by maintaining character offset pairs indicating the beginning and end of the zone together with the type of token, phrase etc. (see Figure 3 for a comparison). However, the purpose of CBC4Kids is question answering evaluation, whereas the latter study used the DISP corpus to compare the relative utility of shallow versus deep parsing in analyzing nominal compounds.

Corpus	DISP	CBC4Kids
Domain	biology	open-domain (news; but geographic bias towards Canada)
Targeted layers	5	>20
Specific challenge	compounds	questions
Stand-off XML	yes	no
Derivation process explicit	no	yes
Visualization	HTML	HTML

Table 3: A comparison between the CBC4Kids and DISP annotation projects.

Work on pipelines. An XML pipeline can also be described in a special glue language for streams such as the XML stream processor STnG [Krupnikov2003]. The main advantage would be to formulate the processing pipeline in a language that allows any kind of executable to be called without using a combination of XML parser and a programming language (such as LTG `xmlperl` plus Perl).

```
<st:case xpath="p/text()"/>
  <st:chain>
    <st:contentFilter
        system="MxTerminator"
        --(calling external tool)--/>
    <st:tee>
      <s>
        <string>
         <st:passThrough/>
        </string>
        <st:contentFilter
            system="PennTokenizer"/>
        ...
```

XML-aware languages. Finally, new special-purpose programming languages are already being designed, which—like CDuce [Benzaken et al.2002]—treat DTDs and their elements as first-order objects and allow direct manipulation of DTDs and XML document instances within the functional paradigm; these are expected to simplify XML processing further.

5 Lessons Learned and Future Work

XML Pervasiveness. The NLP community has now widely adopted the use of SGML or XML for computer corpus annotation. XML-aware software such as input/output application programming interfaces (APIs), search and transformation tools are now also available. However, the linguistic community has not generally adopted XML as the standard output format for parsers, taggers etc., so that it is still necessary to invest significant time to develop converters. The collaborative development scenario we have used here has proved effective in supporting this in a distributed fashion. Because there are is a large number of tools available for XML processing and it is programming language independent, XML is the ideal corpus exchange meta-format within and between groups.

DTD. Tokenization is currently considered a standard word-based process. In fact, it should be encoded as a non-terminal layer because 1) its original input is a sentence not a word, 2) depending on the tokenizer, the number of distinct tokens may vary, thus children should appear in different trees, a a different view of the original sentence and 3) a tokenizer may produce alternative tokenizations of the same sentence.[14]

Additional layers. We would like to add further CBC4Kids layers so that for each major processing step assumed in modern NLP systems' pipelines there is a corresponding corpus annotation layer that can be used off the shelf. With such layers added no tool idiosyncrasies such as use of different parameter settings can spoil replication experiments: layer selection in XML replaces tool integration.

Primary candidates for extension are chunking (C&C chunk [Curran and Clark2003a], LT CHUNK etc.), Named Entity Recognition and Classification (e.g. using MITRE's Alembic [Aberdeen et al.1995], LTG's MUC-7 system [Grover et al.2000], and/or C&C ner [Curran and Clark2003b]), and higher-level layers such as quasi-logical form layers (see below) and co-reference layers.

Within each linguistic stratum, there are alternative theories with their own representations. [Hockenmaier and Steedman2002] present a lexicalized statistical parser for Combinatory Categorial Grammar (especially suited for long distance dependencies, which often occur in questions). This yields a labeled precision of 81.60% and recovers 89.7% or more word-word dependencies on Wall Street Journal text. We are considering the integration of a CCG parse layer in a future release of the corpus.[15]

Limitations. The single major drawback of the Annotated CBC4Kids is that because it was developed outside of any official projects, time constraints forced us to adopt a less integrated markup scheme that would otherwise have been possible: all our markup is inline, not stand-off XML [Carletta et al.2003], and we were forced to allow for some redundancy in the representation. For example, the nodes in the parse trees are usually words (XML content elements), but there are no explicit (XML-)links between the tree nodes connecting them to the token positions in the other representation layers yet. Providing such linking information would have been far from trivial, as different tools usually rely on their own pipelines and make their own assumptions about tokenization etc. Hence our basic level of alignment between linguistic layers is the sentential unit (s-unit).

Applications. CBC4Kids can be used for information fusion experiments, since it contains multiple gold-standard answers as beneficial for automatic answer comparison methods [Dalmas and Webber2003]. It can be used to compare reading comprehension metrics, or assess the quality of existing question answering strategies [Leidner et al.2003] in a controlled test-bed.

Towards Predicate/Argument Structure Surface overlap metrics are intrinsically limited, since they cannot, for instance, distinguish between *man bites dog* and *dog bites man*—they are a-semantic in nature. To overcome this, the various syntactic representations (cf. Figure 3) can be utilized to obtain predicate-argument structures (`bite(man, dog)` versus `bite(dog, man)`), which should allow for higher precision.

[14] This interesting possibility is generally not followed up for efficienc y reasons.

[15] The label `POS2_SYN6` shall be reserved for this layer

Re-use. 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 corpus can be a valuable resource also for other NLQA research groups. The annotated corpus is being distributed by MITRE with layers as given above, including answers given by our system for the Deep Read baseline, including a version in conveniently brows-able HTML. Please contact Dr. Lisa Ferro directly for a copy (at the time of writing she can be reached at `lferro@mitre.org`).

6 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 tool-sets. Our central paradigm is that the whole process of corpus preparation, linguistic annotation, question answering and result visualization can be carried out in an XML framework by a set of filters. We have enriched the corpus with many layers of linguistic information so that the final process of question answering is reduced to a simple sequence of layer selection, iteration, and metric (here: term overlap) computation. 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, and we intend to use it for preparing other resources. On the basis of the result, the annotated CBC4Kids corpus, 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.

Acknowledgments. Lynette Hirschman and Lisa Ferro at MITRE provided us with the initial CBC4Kids corpus and feedback. The Canadian Broadcasting Corporation granted the use of their texts for research purposes. Lisa Ferro is managing the re-distribution of the Annotated CBC4Kids corpus. We would like to thank them as well as the authors of all the tools we have used in this project for making them available to the academic community. Thanks to Maria Lapata, Dekang Lin, Katja Markert, Satoshi Sekine and Bill Wellner for practical advice and feedback. We would like to acknowledge the financial support of the German Academic Exchange Service (DAAD) under grant D/02/01831, Linguit GmbH (research contract UK-2002/2), and of the School of Informatics, University of Edinburgh.

References

[Aberdeen et al.1995] 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.

[Abney1996] S. Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344.

[Abney1997] S. Abney. 1997. Part-of-speech tagging and partial parsing. *Corpus-Based Methods in Language and Speech Processing*.

[Arampatzis et al.2000] A. Arampatzis, Th.P. van der Weide, C.H.A. Koster, and P. van Bommel. 2000. Linguistically-motivated Information Retrieval. In Allen Kent, editor, *Encyclopedia of Library and Information Science*, volume 69, pages 201–222. Marcel Dekker, Inc., New York, Basel.

[Benzaken et al.2002] V. Benzaken, G. Castagna, and A. Frisch. 2002. CDuce: A white paper. In *PLAN-X: Workshop on Programming Language Technologies for XML*.

[Breck et al.2001] E. Breck, M. Light, G. Mann, E. Riloff, B. Brown, P. Anand, M. Rooth, and M. Thelen. 2001. Looking under the hood: tools for diagnosing your question answering engine. In *ACL-2001 Workshop on Open-Domain Question Answering*.

[Buchholz and Daelemans2001] S. Buchholz and W. Daelemans. 2001. Complex answers: A case study using a WWW question answering system. *Natural Language Engineering*.

[Burger et al.2001] J. Burger, C. Cardie, V. Chaudhri, S. Harabagiu andD. 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. Weishedel. 2001. Issues, tasks and program structures to roadmap research in question and answering. *NIST*.

[Carletta et al.2003] J. Carletta, J. Kilgour, T. O'Donnell, S. Evert, and H. Voormann. 2003. The NITE object model library for handling structured linguistic annotation on multimodal data sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web. Third Workshop on NLP and XML (NLPXML-2003)*.

[Charniak1972] E. Charniak. 1972. *Toward a Model of Children's Story Comprehension*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

[Charniak1973] Eugene Charniak. 1973. Jack and janet in search of a theory of knowledge. In *3rd IJCAI-73; reprinted in Natural Language Processing (Grosz,et.al)*.

[Collins1996] M. J. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, San Francisco. Association for Computational Linguistics, Morgan Kaufmann Publishers.

[Collins1997] 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.

[Cotton and Bird2002] 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*.

[Curran and Clark2003a] James R. Curran and Stephen Clark. 2003a. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 91–98, Budapest, Hungary.

[Curran and Clark2003b] James R. Curran and Stephen Clark. 2003b. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada.

[Dalmas and Webber2003] Tiphaine Dalmas and Bonnie Webber. 2003. Information fusion for answering factoid questions. *2nd CoLogNET-ElsNET Symposium. Questions and Answers: Theoretical and Applied Perspectives*.

[Dalmas et al.2003] T. Dalmas, J. L. Leidner, B. Webber, C. Grover, and J. Bos. 2003. Generating annotated corpora for reading comprehension and question answering evaluation. In *EACL-2003 Workshop on Question Answering*.

[Davie1992] A. J. T. Davie. 1992. *An Introduction to Functional Programming Using Haskell*. Cambridge University Press, Cambridge, UK.

[Ferro2000] L. Ferro. 2000. *Reading Comprehension Tests: Guidelines for Question and Answer Writing. (Unpublished Technical Report)*. The MITRE Corporation.

[Grover et al.2000] 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)*.

[Grover et al.submitted] C. Grover, M. Lapata, and A. Lascarides. submitted. A comparison of parsing technology for the biomedical domain. In *Natural Language Engineering*.

[Hirschman et al.1999] 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*.

[Hockenmaier and Steedman2002] 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*.

[Krupnikov2003] K. A. Krupnikov. 2003. STnG: a streaming transformation and glue engine for XML. In *Extreme Markup Languages 2003: Proceedings*, Montréal, Quebéc.

[Lehnert1978] W. Lehnert. 1978. *The Process of Question Answering*. Lawrence Erlbaum Publishers.

[Leidner et al.2003] Jochen L. Leidner, Johan Bos, Tiphaine Dalmas, James R. Curran, Stephen Clark, Colin J. Bannard, Mark Steedman, and Bonnie Webber. 2003. The QED open-domain answer retrieval system for TREC 2003. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, pages 595–599, Gaithersburg, MD.

[Levelt and Kelter1982] W. J. M. Levelt and S. Kelter. 1982. Surface form and memory in question answering. *Cognitive Psychology*, 14:78–106.

[Light et al.2001] M. Light, G. Mann, E. Riloff, and E. Breck. 2001. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering*, 7(4):325–342.

[Lin1995] D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*.

[Lin1998] D. Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*.

[Mikheev et al.1998] 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)*.

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

[Mikheev et al.1999b] A. Mikheev, M. Moens, and C. Grover. 1999b. Named entity recognition without gazetteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 1–8.

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

[NISTonline] NIST, editor. online. *TIPSTER Text Architecture Concept*. http://www-nlpir.nist.gov/related_projects/tipster/.

[Porter1980] M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

[Ratnaparkhi1996] A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*.

[Riloff and Thelen2000] E. Riloff and M. Thelen. 2000. A rule-based question answering syatem for reading comprehension tests. *ANLP/NAACL-2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

[Schmid1994] H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. *International Conference on New Methods in Language Processing*.

[Sekine and Grishman1995] S. Sekine and R. Grishman. 1995. A corpus-based probabilistic grammar with only two non-terminals. In *Proceedings Fourth International Workshop on Parsing Technologies*.

[Thompson1999] S. Thompson. 1999. *Haskell: The Craft of Functional Programming*. Addison-Wesley, Reading, MA, 2nd edition.

[Voorhees1999] E. Voorhees. 1999. The TREC-8 question answering track report. 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.

[Webber et al.2002] 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 XML Document Type Definition (DTD)

```
<!DOCTYPE DOC [
  <!-- ====================================================================
       "CBC4Kids.dtd"

       Document Type Definition for the CBC4KIDS Corpus - CBC News for Kids

       Purpose: Automatic question answering of reading comprehension
                questions

       Author:  Jochen Leidner <leidner@acm.org>
       Modified: Tiphaine Dalmas <t.dalmas@sms.ed.ac.uk>

       $Id: cbc4kids.dtd,v 1.1 2002/10/13 14:39:51 s0239229 Exp $
       ==================================================================== -->

  <!-- PUBLIC "http://www.ltg.ed.ac.uk/xml/dtd/cbc.dtd"
       SYSTEM "cbc.dtd" -->

  <!-- DOC is the root element that encapsulates ONE news story
       and its associated metadata and questions/answers -->
  <!ELEMENT DOC                (INFORMATION, TEXT, QA-SET)>

  <!-- Metadata -->
  <!ELEMENT INFORMATION        (DOCNO, COPYRIGHT, DOCURI?, SRC?)>

  <!-- A document identifier as assigned by MITRE -->
  <!ELEMENT DOCNO                      (#PCDATA)>

  <!-- The notice of who holds the copyright of the document -->
  <!ELEMENT COPYRIGHT          (#PCDATA)>

  <!-- Web location of the story -->
  <!ELEMENT DOCURI             (#PCDATA)>

  <!-- Original source on which the simplified CBC account is based on -->
  <!ELEMENT SRC                (#PCDATA)>

  <!-- The news story proper -->
  <!ELEMENT TEXT                       (HEADLINE, DATE, BODY)>

  <!-- Title of the news story -->
  <!ELEMENT HEADLINE           (STRING, TOKENS?, PARSES?)>
  <!ATTLIST HEADLINE                   id CDATA #REQUIRED>

  <!-- Date of the news story (wrt CBC Website) -->
  <!ELEMENT DATE                       (STRING, TOKENS?, PARSES?)>
  <!ATTLIST DATE                       id CDATA #REQUIRED>

  <!-- The textual core -->
  <!ELEMENT BODY                       (P+)>

  <!-- Paragraphs are sets of sentences -->
```

```
<!ELEMENT P                   (S+)>
<!ATTLIST P                   id CDATA #REQUIRED>

<!-- -->
<!ELEMENT S                   (STRING, TOKENS?, PARSES?)>
<!ATTLIST S                   id CDATA #REQUIRED>

<!-- Sentence as a string, each token separated by a space  -->
<!ELEMENT STRING              (#PCDATA)>

<!-- Tokens list -->
<!ELEMENT TOKENS              (TOKEN+)>

<!-- Token representation for each process -->
<!ELEMENT TOKEN               EMPTY>
<!ATTLIST TOKEN               id  CDATA #REQUIRED
                              process CDATA #REQUIRED
                              src CDATA #REQUIRED
                              dst CDATA #REQUIRED>

<!-- Parses list -->
<!ELEMENT PARSES              (PARSE+)>

<!--   -->
<!ELEMENT PARSE               (NODE|LEAF)>
<!ATTLIST PARSE               process CDATA #REQUIRED>

<!ELEMENT NODE                (NODE|LEAF)+>
<!ATTLIST NODE                type CDATA #REQUIRED>
<!ELEMENT LEAF                (#PCDATA)>
<!ATTLIST LEAF                type CDATA #REQUIRED>

<!-- -->
<!ELEMENT QA-SET              (QA*)>
<!-- -->
<!ELEMENT QA                  (QUESTION, ANSWERS, RESULTS?)>
<!ATTLIST QA                  index CDATA #REQUIRED>

<!-- -->
<!ELEMENT QUESTION            (STRING, TOKENS?, PARSES?)>

<!-- -->
<!ELEMENT ANSWERS             (ANSWER+)>

<!-- -->
<!ELEMENT ANSWER                (HUMAN-ANSWER, AUT-SENTS?, HUM-SENTS?)>
<!ATTLIST ANSWER                id         CDATA #REQUIRED
                                difficulty CDATA #REQUIRED>

<!-- A human-generated answer, where humans were told
     to formulate answers close to the text, but still they
     are not necessarily snippets -->
<!ELEMENT HUMAN-ANSWER        (STRING, TOKENS?, PARSES?)>
```

```
  <!-- Set of automatically extracted sentences -->
  <!ELEMENT AUT-SENTS            (AUT-SENT*)>

  <!-- A extracted sentence considered as answer, computed
      on recall where
        recall = O(sentence, human answer)/ W(human answer)
        W = number of filtered words using F
        O(a, b) = W(a) inter W(b)
        F = STEM1 words - STOP1 words  -->
  <!ATTLIST AUT-SENT             id-ref CDATA #REQUIRED
                                 recall CDATA #REQUIRED>

  <!-- Set of human sentences -->
  <!ELEMENT HUM-SENTS            (HUM-SENT*)>
  <!-- Sentence considered as an answer by human annotators -->
  <!ATTLIST HUM-SENT             id-ref CDATA #REQUIRED>

  <!-- -->
  <!ELEMENT RESULTS              (SYS-SENTS*)>

  <!-- Set of system sentences -->
  <!ELEMENT SYS-SENTS            (SYS-SENT*)>
  <!ATTLIST SYS-SENTS             process CDATA #REQUIRED>

  <!-- Sentence considered as an answer by the system -->
  <!ATTLIST SYS-SENT             id-ref CDATA #REQUIRED
                                 score CDATA #REQUIRED>



  <!-- ====================================================================
       end of file "cbc4kids.dtd"
       ==================================================================== -->
]>
```

B Corpus Distribution: Directory Structure and Files

Release 1.

```
CBC4Kids
    |-- README
    `-- annotated
        |-- data
        |   |-- CBC4Kids.dtd                        The XML DTD
        |   |-- html                                HTML-browsable version of the corpus
        |   |   |-- 1999-W02-1.qa.xml.html          and answers computed by our DeepRead-
        |   |   |-- 1999-W02-5.qa.xml.html          style (word overlap) Q&A pipeline
        |   |   |-- 1999-W03-2.qa.xml.html
        |   |   |
        |   |   |-- (...)
        |   |   |
        |   |   |-- 2000-W06-1.tb.qa.xml.html
        |   |   |-- 2000-W06-4.tb.qa.xml.html
        |   |   |-- 2000-W06-5.tb.qa.xml.html
        |   |   |-- index.html
        |   |   |-- main.html
        |   |   `-- menu.html
        |   `-- xml                                 XML document instances for the
        |       |-- cbcDevTest                      Development Test Set
        |       |   |-- 1999-W02-1.qa.xml           (the final test set is undisclosed)
        |       |   |-- 1999-W03-2.qa.xml
        |       |   |-- 1999-W04-1.qa.xml
        |       |
        |       |-- (...)
        |       |
        |       |   |-- 2000-W05-5.tb.qa.xml
        |       |   |-- 2000-W06-1.tb.qa.xml
        |       |   `-- 2000-W06-5.tb.qa.xml
        |       `-- cbcTraining                     Training Test Set
        |           |-- 1999-W02-5.qa.xml
        |           |-- 1999-W03-5.qa.xml
        |           |-- 1999-W04-5.qa.xml
        |           |
        |           |-- (...)
        |           |
        |           |-- 2000-W03-4.qa.xml
        |           |-- 2000-W03-5.qa.xml
        |           `-- 2000-W06-4.tb.qa.xml
        `-- doc                                     Documentation
            |
            |-- dalmas-etal-2003-eacl.pdf           EACL Q&A WS paper on DeepRead
            |                                       replication experiment on CBC4Kids
            |-- leidner-etal-2003-eacl.pdf          EACL LINC paper on Annotated CBC4Kids
            |-- Dalmas-etal-2004-TR.pdf             This Technical Report (latest version)
            |-- Dalmas-etal-2003-EACL.bib           \
            |-- Dalmas-etal-2004-TR.bib              > BibTeX entries
            |-- Leidner-etal-2003-EACL.bib          /
            |-- Leidner-etal-2003-EACL-slides.pdf   Slides from the LINC 2003 presentation
            `-- dalmas-etal-2003-eacl.slides.pdf    Slides from the EACL 2003 presentation
```

C Sample Story from CBC4Kids

Mourning in an Alberta Town
April 29, 1999

Pastor Ken Gartly provided comfort and prayer to people in Taber, Alberta yesterday after two students were shot at the town's high school.

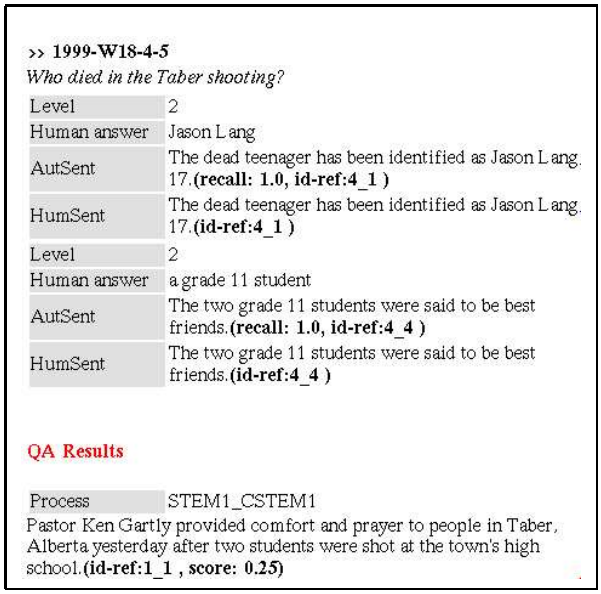
Students at W. R. Myers high school had just settled down after lunch when a 14-year-old boy walked in and shot two students, killing one. The shooting comes a week after the shooting tragedy at Columbine high school in Littleton, Colorado. "We have a son and daughter-in-law in Denver," says Pastor Gartly after an evening service at Taber Evangelical Free Church where worshipers discussed the day's tragic events. The dead teenager has been identified as Jason Lang, 17. The other victim, Shane Christmas, also 17, had emergency surgery yesterday at Lethbridge Regional Hospital. This morning his condition was reported as fair to serious. The two grade 11 students were said to be best friends.

Eight thousand people live in Taber, which is 300 kilometres southeast of Calgary. Many members of the community are members of the Mormon Church or are evangelical Christians.

Taber was founded at the beginning of the century. It is mainly made up of decendents of the area's early homestead pioneers, of Central European, Polish, Japanese, Dutch and various other racial backgrounds. Many Japanese came in 1943, Polish war veterans in 1947, and many Dutch immigrants came after about 1950. Many native Canadians also live in the region.
[...]

Police confirmed the gunman was taken into custody by the school resource officer, who is also a member of the Taber Police Service. 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.

Questions
Who shot two students at a high school in Taber?
What time of day did the Taber shooting take place?
Where is the Taber gunman now?
Who died in the Taber shooting? (see Figure 10)
Why is Shane Christmas in the hospital?
Why were yesterday evening's activities at Taber Evangelical Free Church modified?
When was there a school shooting in Colorado?



>> **1999-W18-4-5**

Who died in the Taber shooting?

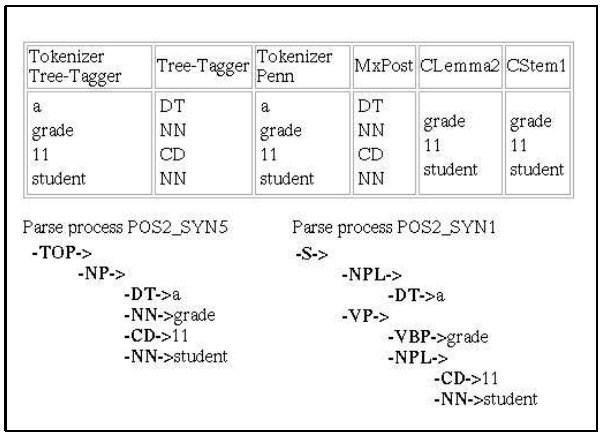
Level	2
Human answer	Jason Lang
AutSent	The dead teenager has been identified as Jason Lang. 17.**(recall: 1.0, id-ref:4_1)**
HumSent	The dead teenager has been identified as Jason Lang. 17.**(id-ref:4_1)**
Level	2
Human answer	a grade 11 student
AutSent	The two grade 11 students were said to be best friends.**(recall: 1.0, id-ref:4_4)**
HumSent	The two grade 11 students were said to be best friends.**(id-ref:4_4)**

QA Results

Process	STEM1_CSTEM1

Pastor Ken Gartly provided comfort and prayer to people in Taber, Alberta yesterday after two students were shot at the town's high school.**(id-ref:1_1 , score: 0.25)**

Figure 10: HTML view of a question.



Tokenizer Tree-Tagger	Tree-Tagger	Tokenizer Penn	MxPost	CLemma2	CStem1
a grade 11 student	DT NN CD NN	a grade 11 student	DT NN CD NN	grade 11 student	grade 11 student

Parse process POS2_SYN5

-TOP->
- **-NP->**
  - **-DT->**a
  - **-NN->**grade
  - **-CD->**11
  - **-NN->**student

Parse process POS2_SYN1

-S->
- **-NPL->**
  - **-DT->**a
- **-VP->**
  - **-VBP->**grade
  - **-NPL->**
    - **-CD->**11
    - **-NN->**student

Figure 11: HTML view of some linguistic layers of the corresponding human answer.