Componential Analysis for Recognizing Textual Entailment

Ken Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872
http://www.clres.com
ken@clres.com

Abstract

CL Research participated in the PASCAL Challenge for Recognizing Textual Entailment using components from its Knowledge Management System. These components, particularly those involved in summarization, proved useful. CL Research submitted two test runs, obtaining accuracy scores of 0.581 and 0.566, largely consistent with results obtained using the development set. The core routine used in assessing entailment was based on an overlap metric that used a test for preponderance of similarity between the hypothesis and the text. Several other mechanisms, e.g., syntactic and semantic tests, or using WordNet, a Roget-style thesaurus, and FrameNet alternation patterns, were considered, but none appeared likely to yield improvements, suggesting the need for an integrated solution. Qualitative observations about these mechanisms suggests future work.

1 Introduction

CL Research participated in the second PASCAL Challenge for Recognizing Textual Entailment (RTE-2) to assess the extent to which experience gained in summarization, question answering, and other similar NLP technologies could be applied to this task. CL Research's Knowledge Management System (KMS) provides several modules for summarization, question answering, information extraction, and document exploration. In participating in RTE, we explored whether and how these modules could be combined to perform the basic task and identified issues that emerged in working with the RTE-2 development set. After developing our system to perform the task, we processed the RTE-2 test set and submitted two runs, with results similar to those that had been achieved with the development set.

In section 2, we briefly describe KMS and the primary modules used to perform the RTE task. In section 3, we describe the system that was constructed for performing the task and that allowed examination of the underlying issues. Section 4 provides the official results from our submission and compares them to the results from using the development set. In section 5, we assess the different tasks and in section 6, we describe our preliminary attempts to use such resources as WordNet, FrameNet, and machine-readable thesauruses, i.e., future work.

2 The Knowledge Management System

The CL Research KMS is a graphical interface that enables users to create repositories of files (of several file types) and to perform a variety of tasks against the files. The tasks include question answering, summarization, information extraction, document exploration, semantic category analysis, and ontology creation. The text portions of files (selected according to DTD elements) are processed into an XML representation; each task is then performed with an XML-based analysis of the texts.

KMS uses lexical resources as an integral component in performing the various tasks. Specifically, KMS employs dictionaries developed using CL Research's DIMAP dictionary maintenance programs, available for rapid lookup of lexical items. CL Research has created DIMAP dictionaries for a machine-readable version of the *Oxford Dictionary of English*, WordNet, the Unified Medical Language System (UMLS) Specialist Lexicon (which provides a considerable amount of syntactic information about lexical items), *The Macquarie Thesaurus*, and specialized verb and preposition dictionaries. These lexcial resources are used seamlessly in a variety of ways

in performing the various tasks.

The KMS text processing component consists of three elements: (1) a sentence splitter that separates the source documents into individual sentences; (2) a full sentence parser which produces a parse tree containing the constituents of the sentence; and (3) a parse tree analyzer that identifies important discourse constituents (sentences and clauses, discourse entities, verbs and prepositions) and creates an XML-tagged version of the document.

The XML representations of the documents are used in performing the various KMS tasks. To perform the RTE task, we made use of summarization and question answering modules, each of which employ lower level modules for dictionary lookup, WordNet analysis, linguistic testing, and XML functions. Litkowski (2006), Litkowski (2005a), and Litkowski (2005b) provide more details on the methods used in TREC question answering and DUC summarization.

3 System for Assessing Textual Entailment

To perform the RTE task, we developed a graphical user interface on top of various modules from KMS, as appropriate. The development of this interface is in itself illuminating about factors that appear relevant to the task.

KMS is document-centric, so it was first necessary to create an appropriate framework for analyzing each instance of the RTE data sets (working initially with only the development set). Since these data were available in XML, we were able to exploit KMS' underlying XML functionality to read the files. We first created a list box for displaying information about each instance as the file was read. Initially, this list box contained a checkbox for each item (so that subsets of the data could be analyzed), its ID, its task, its entailment, an indication of whether the text and the hypothesis were properly parsed, the results of our evaluation, and a confidence score (used initially, but then discarded since we did not develop this aspect further). Subsequently, we added columns to record and characterize any problem with our evaluation and to identify the main verb in the hypothesis.

The interface was designed with text boxes so that an item could be selected from the instances and both the text and the hypothesis could be displayed. We associated a menu of options with the list box so that we could perform various tasks. Initially, the options consisted of (1) selecting all items, (2) clearing all selections, and (3) parsing all items.

The first step in performing the RTE task was to parse the texts and hypotheses and to create XML representations for further analysis. We were able to incorporate KMS' routines for processing each text and each hypothesis as a distinct "document" (applying KMS' sentence splitting, parsing, discourse analysis, and XML representation routines).[1] After performing this step (taking about 15 minutes for the full set), it was found that several texts had not been parsed, due to a bug in a sentence splitting routine. As a result, another option was added to reparse selected items, useful when corrections were made to underlying routines. The result of this parsing step was the creation of an XML rendition of the entire RTE set, approximately 10 times the size of the original data.[2]

The next extension of the interface was the addition of an option to make our evaluation of whether the texts entailed the hypotheses. Our initial implementation of this evaluation was drawn from the KMS summarization functionality. As used in multi-document DUC summarization (see Litkowski, 2005b, for details), KMS extracts top sentences that have a high match with either the terms in the documents or the terms in a topic description. KMS has generally performed quite well in DUC, primarily through its use of an overlap assessment that excludes relevant sentences that are highly repetitive of what has already been included in a growing summary. A key feature of that success is the use of anaphoric references in place of the anaphors. While this feature is significant in multi-document summarization, it is less so for the RTE task. Notwithstanding, this overlap assessment is the basis for the RTE judgment.

The overlap analysis is not strict, but rather based on an assessment of "preponderance." In RTE, the analysis looks at each discourse entity in the hypothesis and compares them to the discourse

[1] Since only a relatively small number of the RTE texts consisted of more than one sentence, the use of KMS discourse analysis functionality was minimal.

[2] The developers of the RTE data sets are to be commended for the integrity of the data. Processing of the data proceeded quite smoothly, enabling us to focus on the task, rather than dealing with problems in the underlying data.

entities in the texts (with all anaphors and coreferents replaced by their antecedents). As used in KMS, discourse entities are essentially noun phrases, including gerundial phrases. Since the overlap analysis is based only on discourse entities, other sentence components, specifically verbs and prepositions, are not considered. And, while discourse entities are further analyzed into lexical components (i.e., nouns, adjectives, adverbs, and conjunctions), the overlap analysis does not make use of these distinctions. Since our summarization performance in DUC has proved adequate without consideration of other leaf nodes, we have not attempted to develop overlap metrics which take them into account, nor have we assessed whether they are important.

Each discourse entity in the hypothesis is compared to the full set of discourse entities in the texts, one by one. In an individual comparison, both discourse entities are lowercased and then split into constituent words. Words on a stop list are ignored. If at least one word in a discourse entity from the hypothesis is contained in a discourse entity from the text, the test returns true.[3] If a match does not occur, a counter of "new" discourse entities is incremented; if a match does occur, a counter of "old" discourse entities is incremented. When all discourse entities from the hypothesis have been tested, the number of new discourse entities is compared to the number of old discourse entities. If there are more new entities than old entities, a sentence (in this case, the hypothesis) is judged to provide sufficient new information so as to be said not to be overlapping. In this case, the judgment is made that the hypothesis is **not entailed** by the text. If the preponderance of old entities is greater than or equal to the number of new entities, the judgment is made that the hypothesis is **entailed** by the text.

After selecting and making entailment judgments on the full set of instances, the interface shows the score, i.e., the number of judgments that match the entailments in the development set.[4] The full evaluation for all instances in development set took about 10 minutes. Thus, in summary, a full run for an RTE data set of 800 items takes less than 30 minutes.

Having made the judgments and computed the accuracy, the next steps of our process involved extending the interface to permit a more detailed analysis of the results. Two major components were added to the interface: (1) the ability to look in detail at the XML representations of the texts and the hypotheses and (2) the ability to examine results for subsets of the full set.

We added a button to view details about a particular item. This displays the XML representation of the text and the hypothesis, as well as the list of discourse entities for each. The display also shows the entailment and our evaluation. It also contains a drop-down list of "problems." If our evaluation is incorrect, we can assign a reason (and use a growing list of problem assessments). When this display is closed, any problem that has been assigned is then listed next to the item.

To examine subsets for more in-depth analysis, we added a set of five lists of selection criteria. The lists are (1) the subtask, (2) the official entailment, (3) our evaluation, (4) our problem assessment, and (5) the main verb of the hypothesis. Any combination of these selection criteria can be made (including "All"). The set of options was then expanded to select all items meeting those criteria. The subset can then be scored by itself (e.g., to determine our score on just the QA task).

Finally, the interface was extended to permit an assessment of any changes that were made to the underlying system. Thus, given a current evaluation, and then making some change in an underlying component, we could determine changes in the evaluation (YES to NO or NO to YES) and changes to our score (CORRECT to INCORRECT or INCORRECT to CORRECT).[5]

4 Results

Most of our efforts have been spent on examining the results of our system on the development set. We used this set to examine and consider various modifications to our system before making our

[3] A test is made of whether there is an exact match between two discourse entities, but this result is not currently used.

[4] The interface also shows the confidence weighted score, but as mentioned, this aspect was not further developed and all scores were set to the same value, so that this score is equal to the accuracy.

[5] In using the interface, it has become clear that a further extension is desirable to assess the effects of different changes. This would involve the modularization of different underlying components in making different tests so that the effect of various combinations could be tested. The interface would enable the selection of different combinations.

official submissions. We made two official runs, with only one major change. After our official submission, we made a full run using the RTE-1 test set. Table 1 provides the summary results over all 800 test items in each of these runs. Tables 2 and 3 breaks down the results by subtask for the development set and for the first official run.

Table 1. Summary Results	
Run	**Accuracy**
RTE-2 Development	0.590
RTE-2 Test (run1)	0.581
RTE-2 Test (run2)	0.566
RTE-1 Test	0.549

Table 2. RTE-2 Development Subtask Results	
Subtask	**Accuracy**
Information Extraction (IE)	0.550
Information Retrieval (IR)	0.570
Question Answering (QA)	0.560
Summarization (SUM)	0.690

Table 3. RTE-2 Test (run1) Subtask Results	
Subtask	**Accuracy**
Information Extraction (IE)	0.500
Information Retrieval (IR)	0.615
Question Answering (QA)	0.520
Summarization (SUM)	0.690

As shown in Table 1, the initial accuracy[6] achieved in the RTE-2 development set was 0.590, higher than what any full run obtained in RTE-1 (Dagan et al., 2005). This was somewhat encouraging, particularly since the results were based on making a positive decision for each item (as opposed to making a default decision based on chance except when a positive decision could be made). For run1 of the RTE-2 test set, we used the identical system and the overall results were roughly consistent. However, as shown in Tables 2 and 3, there was significant variation by subtask between the development and the test sets. We have not yet examined the reasons for these differences.

For run2 of the RTE-2 test set, one major modification was made to the underlying system, the addition of a test for subject mismatch. The RTE-1 Test set was run after our official submissions, in preparation for this paper, and after a change in our underlying XML rendition routines, which decreased the likelihood of a positive overlap assessment, thus resulting in a lower accuracy. These are discussed further in the next section.

[6]Since all items were answered, accuracy is equivalent to precision.

5 Interpretation and Analysis of Results

In all runs, a considerable majority of the 800 entailment judgments were in the affirmative, as shown in Table 4. Our system is clearly erring on the side of making the judgment that the hypotheses overlap with the texts. This reflects the reliance of our method on assessing only the noun phrases in the hypotheses against those in the texts. For the most part, it is to be expected that the test items used terms in the hypotheses that appeared in the texts, perhaps modifying the way that they appeared in relation to one another. It is noteworthy that our accuracy on the positive answers was somewhat lower than for the negative answers (0.578 vs. 0.614). That is, when our method asserts that the preponderance of discourse entities in the hypotheses is toward new items, our system is more likely to judge that the text does not entail the hypothesis. The lower number of positive answers for RTE-1, and the lower accuracy shown in Table 1, reflects the inclusion of adverbs as discourse entities, without a modification to the overlap assessment that should have excluded these items in the test.

Table 4. Number of Positive Answers	
Run	**Positive**
RTE-2 Development	476
RTE-2 Test (run1)	515
RTE-2 Test (run2)	475
RTE-1 Test	439

The observation that our system was providing more positive judgments than negative judgments is also reflected in an overall assessment of the errors. Table 5 shows the error types by subtask, e.g., YES-NO indicates a YES entailment, but our system judged a NO entailment. The differences by subtask are noteworthy, suggesting that where differences in discourse entities are likely to reflect real differences in the text (IR and SUM), the errors are generally equal, whereas in cases where differences in ordering are likely to be significant, considerably more errors were made in asserting entailment when it was not present.

Table 5. Error Types by Subtask (RTE-2 Development Set)		
Subtask	**YES-NO**	**NO-YES**
IE	22	68
IR	48	38
QA	21	67
SUM	34	28

Having grouped the error types in this general way, we were able to focus the error analysis in ways that otherwise would not have been possible. In particular, it quickly became clear that there were significant differences in the types of errors and that different approaches were necessary. In general, YES answers require different types of analysis from NO answers. YES answers imply that there is sufficient overlap in the discourse entities, but that after this assessment, it is necessary to determine if the discourse entities bear similar synactic and semantic relations to one another. NO answers, on the other hand, require a further analysis to determine if we have overlooked synonyms or paraphrases.

In examining YES answers which should have been NO answers, we were able to observe many cases where the difference was in the subject of a verb. That is, the subject of the verb in the hypothesis was different from the subject of the same verb in the text (even though this subject appeared somewhere in the text). We termed this a case of "subject mismatch" and implemented this test for those cases where an initial assessment was entailment. We modified the underlying code to make this test, working with one item (126, with the hypothesis, "North Korea says it will rejoin nuclear talks", where the subject of "say" in the text was "Condoleezza Rice").

After making this change on the basis of one item, we reran our evaluations for all items. This is the difference between **run1** and **run2**. As indicated in Table 5, the effect of this change was a reduction in the number of positive answers from 515 to 475. However, as shown in Table 1, the effect on the accuracy was a decline from 0.581 to 0.566, a net decline of 12 correct answers. Of the 40 changed answers, 26 were changed from correct to incorrect. We were able to investigate these cases in detail, making a further assessment of where our system had made an incorrect change. Several problems emerged (e.g., incorrect use of a genitive as the main verb). Making slight changes would have improved our results slightly, but were not made because of time limitations and because it seemed that they ought to be part of more general solutions.

As indicated, inclusion of the subject mismatch test (also observed in working with the development set) seemed to decrease our performance. As a result, it was not included in **run1**, but only in **run2**, with the expectation of a decline, borne out when the score was computed.

6 Considerations for Future Work

Similar results to that of using the subject mismatch test seemed likely when considering other possible modifications to our system, namely, that although the result for some specific items would be changed from incorrect to correct, the effect over the full set would likely result in an overall negative effect. We describe the avenues we investigated.

As mentioned earlier, we observed the need for different types of tests depending on the initial result returned by the system. We also observed that the subtask is significant, and perhaps has a bearing on some of the test items. The main concern is the plausibility of a hypothesis and how this might be encountered in a real system.

For summarization, the hypotheses appear to be valid sentences retrieved from other documents, overlapping to some extent. This task appears to be well drawn. In this subtask, the key is the recognition of novel elements (similar to the novelty task in TREC in 2003 and 2004). For our system, this would mean a more detailed analysis of the novel elements in a hypothesis. The information retrieval task appears to be somewhat similar, in that the hypotheses might be drawn from real texts.

For question answering, the task appears to be less well-drawn. Many of the non-entailed hypotheses are unlikely to appear in real texts. We believe this is reflected in the many NO-YES errors that appeared in our results. A similar situation occurs for the information extraction task, where it is unlikely that non-entailed hypotheses would be found in real text, since they are essentially counterfactual. (The non-entailed hypotheses in item 209, *Biodiesel produces the 'Beetle'*, and item 226, *Seasonal Affective Disorder (SAD) is a worldwide disorder*, are unlikely to occur in real text.)

We reviewed material from RTE-1 (Dagan et al.) to identify areas for exploration. As indicated, this led to our major approach of using overlap analysis as employed in KMS' summarization routines. We considered the potential for various forms of syntactic analysis, particularly as described in Vanderwende et al. (2005), since many of these constructs are similar to what KMS employs in its question answering routines (i.e., appositives, copular constructions, predicate arguments, and active-passive alternations. However, when we examined instances where these occurred in the development set and were relevant

to determination of entailment, we found that, similar to the subject mismatch test, many answers were already correctly assessed and that it was likely that implementation of general routines would lead to an overall decline in performance.

We next turned to questions of synonymy and paraphrasing. We considered the use of WordNet on several levels. First, we observed cases where some general categorization schema (to capture classes of words, e.g., the tops in WordNet) would provide useful results. Similarly, the use of synsets appeared as if they would provide some benefit, but these did not appear to provide sufficiently broad categories. The addition of derivational links in WordNet is also too sporadic to use systematically.

Limitations in WordNet led to consideration of using a broader category analysis provided in a Roget-style thesaurus. Such a thesaurus would provide a grouping to recognize more instances of synonymy as well as nominalizations (such as *direct* and *director*). However, we were not able to implement use of the thesaurus at this time.

Finally, we considered the use of FrameNet. Although it does not provide a considerable range of lexical items, it may have some information that could be employed effectively in entailment analysis. In The Preposition Project (Litkowski & Hargraves, 2005), syntactic alternation patterns for particular frame elements are being identified. Thus, for example, the verb *work* has a frame element *Employer* associated wth the preposition *for* (there are 19 instances where *work* is the main verb of the hypothesis in the development set). The Preposition Project identifies other lexical items and syntactic patterns that instantiate the *Employer* frame element. Thus, we find that an *Employer* can be found as a noun modifier (*ECB spokesman*, *Bank of Italy governor*), the object of the preposition *by* following the verb *employed* (*employed by the United States*), the object of the preposition *of* (*of FEMA*), and the object of the preposition *at* (*press official at the United States embasssy*). These patterns can be used to look in the text associated with the hypotheses to determine whether the particular pattern is present. However, once again, implementation of such strategies will not necessarily result in an overall net improvement. Thus, for example, our score for the cases involving *work* was 0.526 without the inclusion of such tests. Further exploration of the possible use of FrameNet seems worthwhile.

7 Conclusions

The PASCAL Challenge for Recognizing Textual Entailment has provided a useful mechanism for examining basic strategies involved in assessing equivalence in meaning. While basic overlap analysis as used in summarization has proved worthwhile, RTE has revealed some shortcomings in its use and suggests some possible avenues making improvements. Attempts to make use of lexical resources for making assessments of synonymy and paraphrase has suggested that WordNet does not provide adequate levels of granularity. Possible improvements might be gained from a Roget-style thesaurus and syntactic alternation patterns derived from FrameNet. In general, it appears that an integrated approach is needed to provide consistent improvements.

References

Ido Dagan, Bernardo Magnini and Oren Glickman. 2005. The PASCAL Recognizing Textual Entailment Challenge. In PASCAL: Proceedings of the First Challenge Workshop. *Recognizing Textual Entailment*, Southhampton, U.K, pp. 1-8.

Kenneth C. Litkowski (2005a). Evolving XML and Dictionary Strategies for Question Answering and Novelty Tasks. In E. M. Voorhees & L. P. Buckland (eds.), *The Twelfth Text Retrieval Conference (TREC 2004)*. NIST Special Publication 500-261. Gaithersburg, MD., TREC 2004 Proceedings CD.

Kenneth C. Litkowski (2005b). Evolving XML Summarization Strategies in DUC 2005. Available http://duc.nist.gov/pubs.html.

Kenneth C. Litkowski. 2006. Exploring Document Content with XML. In Voorhees, E. And L. Buckland (eds). *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004*. NIST Special Publication 500-261. Gaithersburg, MD, pp. 52-62.

Kenneth C. Litkowski & Orin Hargraves. 2005. The Preposition Project. ACL-SIGSEM Workshop on "The Linguistic Dimensions of Prepositions and their Use in Computational Linguistic Formalisms and Applications", University of Essex - Colchester, United Kingdom. 171-179.

Lucy Vanderwende, Deborah Coughlin, and Bill Dolan. 2005. What Syntax Can Contribute in Entailment Task. In PASCAL: Proceedings of the First Challenge Workshop. *Recognizing Textual Entailment*, Southhampton, U.K, pp. 13-16.