Overlap Analysis in Textual Entailment Recognition

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

Abstract

CL Research participated in the main and search pilot tasks of the 2009 Recognizing Textual Entailment track (RTE-5). Our system was little changed from what was used in previous RTE exercises. As a result of an apparent increased complexity in the task, our scores have declined from those in previous years. We submitted one 2-way run in the main task, with an accuracy of 0.53, and two runs in the search pilot, with f-scores of 0.28 and 0.29. At present, our system consists solely of routines to examine the overlap of discourse entities between the texts and hypotheses. We present the algorithms used in the step. We examine the potential use of other resources, including WordNet, a Roget-style thesaurus, and FrameNet, but have not yet implemented methods for exploiting these resources.

1 Introduction

CL Research participated in the first three PASCAL Challenges for Recognizing Textual Entailment (RTE). Even though no significant changes were made, we participated in RTE-5 as a prelude to making fuller use of various resources that are part of our overall text processing system, where we perform disambiguations to capture senses of all words from WordNet, FrameNet, a Roget-style thesaurus, and a preposition dictionary assigning semantic roles. 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-5 test sets and submitted three runs, with results similar to those that had been achieved with the development set.

In section 2, we briefly describe the CL Research Linguistic Task Analyzer (LTA), the environment used to perform the RTE task. In section 3, we describe the algorithms that analyze the overlap between the text and the hypothesis, as used in making the entailment judgment. Section 4 provides the official results from our submission and compares them to the results from using the development set. In section 5, we describe our preliminary attempts to use such resources as WordNet, FrameNet, and machine-readable thesauruses, i.e., future work.

2 The Linguistic Task Analyzer

The CL Research Linguistic Task Analyzer (LTA) is a graphical interface that provides an integrated environment for performing various linguistic tasks such as word-sense disambiguation, semantic role labeling, and textual entailment. Each of these tasks involves core modules for processing text into an XML representation. These modules segment a text into sentences, parse each sentence, analyze the parse trees into a discourse structure, and create an XML representation of the text that can then be analyzed for task-specific purposes.

LTA uses lexical resources as an integral component in performing the various tasks. Specifically, LTA 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 LTA 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 LTA 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 LTA, as appropriate. The development of this interface is in itself illuminating about factors that appear relevant to the task.

LTA 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 LTA's 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 LTA routines for processing each text and each hypothesis as a distinct "document" (applying LTA sentence splitting, parsing, discourse analysis, and XML representation routines).[1] The result of this parsing and analysis step was the creation of an XML rendition of the entire RTE set, approximately 13 times the size of the original data. This processing step took approximately 40 minutes.[2]

Once the text and hypothesis components for each instance had been generated, the LTA command to "Run and Evaluate" was executed to make the entailment decisions for each instance. This took approximately 15 minutes for the entire set. Working with the development set, we examined the workings of our algorithms on instances where incorrect judgments were made, making changes and re-executing "Run and Evaluate" to determine the effects of the changes (e.g., how many changes improved and worsened our results).[3]

Since the representation of the text and hypothesis for an instance was in XML, the basic algorithm in the evaluation makes use of XPath expressions to select the data to be analyzed. The core of this evaluation relies upon an examination of the discourse entities (generally noun phrases, but also including constituents such as gerundial phrases). Each discourse entity constitutes an XML node, consisting of child nodes for each word in the noun phrase. All of these nodes have many attributes, identifying such things as a WordNet

[1] In the early RTE evaluations, the text consisted of only one sentence, so the use of LTA discourse analysis functionality was minimal. RTE-5 data consists of multiple sentences, so the resolution of anaphoric references, even for definite noun phrases, has assumed greater importance.

[2] Since the RTE files were not "cleaned", stray marks and unrecognizable entity references had to be cleaned from the source files so that robust parsing could be performed.

[3] LTA was set up to allow selection of instances with certain properties of an instance (such as entailment, evaluation, and main verb of the hypothesis). This enabled us to rerun the system on subsets of the development set.

sense number, anaphoric references to discourse entities earlier in the text, number, and type). All of this information is available for analysis.

```
Select all discourse entities from the instance text t
        (seenDEs)
Select the hypothesis sentence (h)
If Overlap(seenDEs,h)
        If not SubjectMismatch (t, h, mainVerb)
                Entailment = YES
        Else
                Entailment = NO
Else
        Entailment = NO
```

Figure 1. Main Entailment Algorithm

Figure 1 shows the top-level algorithm, where we obtain all the discourse entities from the text and the XML node for the hypothesis sentence. We then determine the overlap between the text discourse entities and the hypothesis (see Figure 2). If this function returns a true value, then if there is not a mismatch in the subject (using the main verb of the hypothesis, see Figure 4), we judge that the text entails the hypothesis. If there is a subject mismatch or there is not an overlap, we judge that the text does not entail the hypothesis.

```
Select all discourse entities from sentence
        (currDEs)
Set newDEs and oldDEs to 0
For each string ce of currDEs
        foundDE = false
        For each string se of seenDEs
                If seenDe has antecedent
                        se = antecedent
                If DEContained (se, ce)
                        foundDE = true
                        break
        If not foundDE
                newDEs++
        Else
                oldDEs++
If newDEs > oldDEs
        return false
Else
        return true
```

Figure 2. Overlap (seenDEs, sentence)

Figure 2 shows the overlap algorithm. First, we obtain all the discourse entities from the hypothesis. We then enter a loop to determine the extent to which the hypothesis entities are present in the text discourse entities. The core function of this routine is a test of whether each hypothesis discourse entity can be considered as contained in a text discourse entity, based on a word by word comparison (see Figure 3). A crucial part of this assessment replaces a referring expression by its antecedent, if present (i.e., as an attribute of the XML node for the text discourse entity). If a hypothesis discourse entity is found to be contained in a text discourse entity, it is construed as an old discourse entity; otherwise, it is counted as new. After all hypothesis discourse entities have been tested, the functin returns true if the number of discourse entities in the hypothesis found in the text sentences is not less than the number not found.

```
Lowercase de1 and de2
Split de1 and de2 into individual words
Set atleastone to false
For each word in de2
        Next if word is on stop list
        Set atleastone to true
        If word is not found in de1 list
                return false
If not atleastone
        return false
Return true
```

Figure 3. DEContained(de1, de2)

Figure 3 shows the test used to determine whether a hypothesis discourse entity (**de2**) is contained in a text discourse entity (**de1**). The hypothesis discourse entity must contain at least one non-stop word. If it contains a non-stop word that is not present in **de1**, the function returns false. Thus, every content word in a hypothesis discourse entity must be present in the discourse entity being compared for there to be a match. This allows a text discourse entity to contain other words and does not examine the order of the words.

Figure 4 shows the test of the main verb in the hypothesis that is used when the overlap test (Figure 2) has been met. In this test, we essentially examine whether the subject of the main verb in the hypothesis is consistent with the subject of the same verb in a text sentence.This test is looking for a subject mismatch, so it only returns that one has occurred when there is a clear difference in the discourse entities of the subjects of these verbs. In general, these conditions are difficult to meet, so the function returns false whenever the hypothesis verb

*Select a verb node from the text (**tv**) and the hypothesis (**hv**) equal to **verb** (either exactly or as the base form)*
*If both **tv** and **hv** exist and are preceded by discourse entities* (**tvde** and **hvde**)
*If not **DEContained (tvde, hvde)***
return true
Return false

Figure 4. SubjectMismatch (t, h, verb)

does not appear in any of the text sentences, whenever the respective subjects cannot be clearly identified, and whenever it appears that the subject of the hypothesis verb is contained in the subject of the text verb (see Figure 3).

Performing the search pilot task essentially uses the same steps as the main task, with the principle modifications coming from the setup of the task. In this task, a set of hypotheses is associated with a corpus of (10) documents. The task is to find all sentences in the corpora that entail the hypotheses. As in the main task, all the sentences (in both the hypotheses and the corpora) are processed into XML representations, which are then used in making the entailment assessments. The main difference between the main task and the search pilot task is that each sentence in the corpora is assessed against each hypothesis sentence.

*For each hypothesis **h***
*For each document **d** in the corpus*
*For each sentence **s** in **d***
Select all discourse entities (**seenDEs**)
*If **Overlap(seenDEs,h)** or **MainVerbMatch(s,h)***
Entailment = YES

Figure 5. Seaerch Pilot Entailment Algorithm

The algorithm for the search pilot is shown in Figure 5. At its core, this algorithm is essentially identical to the one used in the main task, with the test for a subject mismatch removed.[4] While it may appear that each sentence in a corpus document is evaluated independently, this is not the case. Each document in the corpus is processed as a coherent text, so that many anaphoric references are incorporated into the XML representation. When a sentence's discourse entities are evaluated, any antecedents replace the referring expression and are used in the comparison of the discourse entities. An additional test on the verbs was added to expand the set of sentences judged to entail the hypotheses. This test is described below.

*Select all verbs from the hypothesis (**hverbs**)*
*For each verb **v** in **hverbs***
*Set **v** to its base form if necessary*
*Next if **v** = "be"*
*If **v** occurs in **s** (either directly or in its base form)*
Return true
Return false

Figure 6. MainVerbMatch (s, h)

The main verb test algorithm is shown in Figure 6. This test allows for the possibility that multiple verbs may be present in the hypothesis. The test always examines only the base forms of the verbs. The copular verb "be" is ignored (and hence not used as the basis for making a judgment of entailment). This test was added during development. It was not included in the first run, but was used in the second run.

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 entities in the texts (with all anaphors and coreferents replaced by their antecedents). 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.

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. If a match does not occur, a counter of "new" discourse entities is

[4]In general, it seems as if the need for the subject mismatch test arose because of an attempt to provide different subjects for the same verb in the main task, in an effort to "trick" the systems. This wasn't necessary in the search pilot task.

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.

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.

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).

4 Results

All of our efforts were 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 official submissions. We made one official 2-way run for the main task and two official runs for the pilot search task. Table 1 provides the summary results over all test items for RTE-5, as well as for our earlier participation in the first three RTEs. Tables 2 breaks down the results by subtask for the main task.

Table 1. Summary Results (Main)

Run	Accuracy
RTE-5 Test	0.532
RTE-5 Development	0.557
RTE-3 Test	0.613
RTE-2 Test (run1)	0.581
RTE-2 Test (run2)	0.566
RTE-1 Test	0.549

Table 2. RTE-5 Main Subtask Results

Subtask	Accuracy
Information Extraction (IE)	0.480
Information Retrieval (IR)	0.620
Question Answering (QA)	0.495

As shown in Table 1, the accuracy for the test set was lower than for the development set. Although not shown, this pattern has been true for the earlier RTEs as well. Our system has not changed substantially over the years, so the variation of scores seems to reflect the difficulty of the test instances from year to year. The effect is even more pronounced for RTE-5, since the initial scores for the development set were somewhat lower (approximately 0.530). Slight changes were made in the overlap algorithm and thus account for the improvements.

Table 3. RTE-5 Pilot Search Test Results (Macro-average per hypothesis)

Measure	Dev	Run1	Run2
Precision	0.258	0.313	0.326
Recall	0.342	0.467	0.530
F-Score	0.239	0.375	0.404

The results for the pilot search task, shown in Table 3, are considerably better than were obtained during the development phase. For both the development and the test set, our performance on some hypotheses and document sets were extremely low. At this time, we have no explanation for the significant difference between the development and test phases. As noted earlier, the inclusion of the test for the verbs in the hypotheses accounts for the differences between run 1 and run 2 in the test set. It is noteworthy that, not only did this test increase the recall significantly, but it did not come at the expense of precision.

5 Considerations for Future Work

We did not perform and submit the results of any

ablation tests. As indicated by the algorithms presented in this paper, we did not make overt use of any resources. However, as also indicated, the generation of an XML representation of the texts is completely integrated with several lexical resources. At present, this leads to the assignment of many attributes to each node of the XML, particularly for nouns, verbs, and prepositions. We make use of WordNet, FrameNet, data from The Preposition Project, an integrated dictionary (the *Oxford Dictionary of English*), and a Roget-style thesaurus (the Macquarie Thesaurus).

However, at this time, we do not make use of any of the information that is generated. And, in particular, we have not developed strategies for dealing with particular kinds of entailment issues. With LTA, we have developed an elaborate set of mechanisms for testing out many strategies, but this remains a topic for future development.

References

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.