A simple system for detecting non-entailment

Eric Breck
Rhodes College
2000 N Parkway
Memphis, TN 38112, USA
brecke@rhodes.edu

Abstract

In order for a text to entail a hypothesis, the text usually must mention all of the information in the hypothesis. We use this observation as a basis for a simple system for detecting non-entailment. Unlike many previous lexically-based systems, we do not measure the degree of overlap or similarity, and we do no machine learning. This simple system performs well on the Recognizing Textual Entailment (RTE) evaluation.

1 Introduction

Textual entailment is a relationship between two pieces of text T and H, that holds if whenever T is true, H is also true. Systems that recognize when such entailment holds could play a role in many natural language processing contexts, such as question answering or summarization. The Recognizing Textual Entailment (RTE) challenge has been held now for five years (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Giampiccolo et al., 2008), and presents an opportunity for researchers to compare approaches to the textual entailment task on a common dataset.

This is our first year participating in the RTE task, and we chose to begin with a simple approach. In order for a text T to entail a hypothesis H, many relations must hold between the two, but at the very least, all information mentioned in H usually must be mentioned in T[1]. Our system thus recognizes non-entailment by first identifying units of information in H, looking for each piece of information in T, and rejecting entailment if any information is not found. Our system does not attempt to prove entailment; entailment is the default output if it cannot be rejected. We also do not measure the degree of match, often called similarity or overlap; either all the units match, and the judgment is entailment, or one or more do not match, and the judgment is non-entailment.

The rest of this paper proceeds as follows. In Section 2, we describe our approach in more detail, and add a simplistic system for recognizing contradictions. In Section 3, we present the results of our system on the current RTE task as well as prior tasks. Finally, we conclude in Section 4.

2 Approach

Our system begins with the expectation that for entailment to hold, all information mentioned in the hypothesis H must be mentioned in the text T. There are thus two pieces to the approach: recognizing the units of information in the text and hypothesis, and determining whether a unit of information in the hypothesis is mentioned in the text. We expect that a failure of entailment of this sort will generally be of the "unknown" category, so we also add a simple system for recognizing contradictions.

[1]This does not always hold. For example "John was in California this summer" entails "John was not in New York this summer." Preliminary error analysis suggests this is not a frequent cause of error on the RTE datasets, but we need to investigate this further.

Parameters
$used\_pos \leftarrow$ {common noun, proper noun, number}
$match\_types \leftarrow$ { exact, edit, acronym, wordnet}

Given input pair $text$ and $hypothesis$:
$text\_words \leftarrow$ tokenize and tag $text$
$hypo\_words \leftarrow$ tokenize and tag $hypothesis$
$hypo\_words \leftarrow$ remove any words with part-of-speech not in $used\_pos$
if ($\forall h \in hypo\_words.\exists t \in text\_words.\exists m \in match\_types.matches(h, t, m)$) return ENTAILMENT
else return UNKNOWN

Figure 1: Algorithm for simple non-entailment recognition (without contradiction detection)

2.1 Units of information

In our system, the units of information are simply words, and specifically open-class words: common nouns, proper nouns, verbs, adjectives, adverbs, and numbers. The difficulty of determining whether a hypothesis word is mentioned in the text varies by part of speech. For example, a proper name in the hypothesis is likely to occur as the same string in the text, while a verb like "lives" might be discussed in the text using different words entirely. Therefore in experiments below, we choose a particular set of parts of speech, and ignore all other words in the hypothesis. Based on pilot experiments (see Section 3.2), we use common nouns, proper nouns, and numbers.

2.2 Mentioning

We use several methods to determine whether words mentioned in the hypothesis were also mentioned in the text. The simplest method is simply exact string match (allowing for case variation). This is effective for proper nouns and numbers, and less so for common nouns, and even less so for other parts of speech.

The next method is edit distance. Specifically, we count two words as matching if 80% of the letters of the hypothesis word occur in one or more adjacent text words in the same order. This is clearly simplistic, but it does allow for some typos and spelling errors, and also crudely handles some morphological variation. It also will match hyphenated words against non-hyphenated words, a not uncommon case.

The third method we use is to match acronyms. We match words in all caps against sequences of capitalized words whose initial characters concatenate to form the acronym. Clearly this is specific to proper nouns, but it represents a fairly common case.

Finally, we use lexicon-based matching. We experimented with WordNet (Fellbaum, 1998), and also an automatically derived thesaurus created by Dekang Lin[2]. In pilot experiments, the Lin thesaurus did not produce high enough precision matches to be useful. The WordNet matching did prove to be helpful. While many metrics have been developed for WordNet-based semantic similarity[3], we chose a very simple metric: whether the two words were connected by a path of distance at most 2 in the WordNet graph, using any links (i.e. not just hyponymy and hypernymy, but also meronymy, pertainymy, etc.). We used WordNet's lemmatization, so this matching method handles not just synonymy but also morphological similarity.

2.3 Algorithm

Figure 1 describes the simple algorithm for non-entailment recognition. In this algorithm, $matches(h, t, m)$ means that using matching method m, hypothesis word h corresponds to text word t.

[2] `http://www.cs.ualberta.ca/~lindek/downloads.htm`

[3] Implementations for many metrics can be found in the `WordNet::Similarity` package for Perl: `http://wn-similarity.sourceforge.net/`

2.4 Contradiction detection

Since RTE-3, the category of non-entailment has been divided into two types - unknown entailments and contradictions. We therefore add a simple module to detect contradictions. There are two methods: noticing antonymy relations between the text and the hypothesis, and noticing negation.

To recognize antonymy, we consider the paths through WordNet between hypothesis words and text words. If any path for any word passes through an antonymy relation, the instance is judged to be a contradiction.

To notice negation, we first attempt to match the hypothesis verb (whether or not verbs are among our units of information) to a text verb. If the text verb is preceded (within two words) by a negation word, the instance is judged to be a contradiction. Our list of negation words consists of the negation words "no" "n't" "none" "neither" "nor" "few" "each" "every" "without", along with a much longer list from a recent study that automatically learned a large set of so-called downward-entailing operators (Danescu-Niculescu-Mizil et al., 2009) (such as "hardly" or "nobody"). The downward-entailing operators are not precisely negation words, but we hoped that they might help to detect negation.

3 Experiments

In this section, we describe our experimental setup and present results on the official RTE-5 task as well as post hoc experiments on prior RTE datasets.

3.1 Tools and external resources

For tokenization and part-of-speech tagging, we used the Stanford parser[4]. As described above, we used WordNet and Lin's thesaurus. No other resources were used.

3.2 Results

Table 1 presents the official results of our system on RTE-5. Our simple system ranks roughly in the middle of all submitted runs for the two-way categorization, and in the upper half on three-way categorization. Table 2 shows the results of running our current system on test data from previous RTE evaluations, demonstrating that our system compares favorably with systems in past evaluations.

[4] http://nlp.stanford.edu/software/lex-parser.shtml

two-way results	
High for all systems	0.7350
Median for all systems	0.6117
Low for all systems	0.5000
our system	0.61
three-way results	
High for all systems	0.6833
Median for all systems	0.5200
Low for all systems	0.4383
our system	0.57

Table 1: Official RTE-5 results.

Evaluation	two-way score	rank
RTE 1	0.586	1/16
RTE 2	0.605	6/24
RTE 3	0.672	4/27
RTE 4	0.614	7/27

Table 2: Performance of our system on past RTE evaluations (post hoc). Rank is among groups participating in the evaluation, eliminating partial submissions

Table 3 shows that the performance of our system varies widely by subtask. The performance on the information retrieval task is strong, but the performance on information extraction is very weak (essentially random), with the question answering task in between.

Table 4 presents the results of some ablation: removing the contradiction detection, and restricting the used parts of speech. It seems that contradiction detection and numbers are not useful, although Table 5 shows that these did provide a consistent small performance improvement on previous RTE datasets. Table 6 shows the results of using additional parts of speech (verbs, adjectives, and adverbs), which were not used in the above experi-

	all	QA	IE	IR
two-way results				
our system	0.61	0.565	0.51	0.755
three-way results				
our system	0.57	0.545	0.49	0.675

Table 3: Official RTE-5 results - results on subtasks.

	all	QA	IE	IR
two-way results				
our system	0.61	0.565	0.51	0.755
our system-CON	0.6167	0.58	0.51	0.76
our system-CON-NUM	0.6133	0.57	0.515	0.755
our system-CON-NUM-NNP	0.58	0.57	0.52	0.65
three-way results				
our system	0.57	0.545	0.49	0.675
our system-CON	0.5783	0.56	0.495	0.68
our system-CON-NUM	0.5767	0.55	0.5	0.68
our system-CON-NUM-NNP	0.5583	0.55	0.505	0.62

Table 4: Official RTE-5 results: ablation. -CON represents removing contradiction detection. -NUM represents not requiring that numbers in the hypothesis match the text. -NNP represents not requiring that proper nouns in the hypothesis match the text.

	RTE1	RTE2	RTE3	RTE4
our system	0.586	0.605	0.672	0.614
our system-CON	0.584	0.603	0.671	0.610
our system-CON-NUM	0.579	0.600	0.659	0.611
our system-CON-NUM-NNP	0.570	0.579	0.586	0.587

Table 5: Ablation results on past RTE test datasets (post hoc). Two-way classification results only. -CON represents removing contradiction detection. -NUM represents not requiring that numbers in the hypothesis match the text. -NNP represents not requiring that proper nouns in the hypothesis match the text.

	RTE1	RTE2	RTE3	RTE4	RTE5
our system	0.586	0.605	0.672	0.614	0.610
our system+VB	0.566	0.593	0.639	0.618	0.622
our system+RB	0.580	0.609	0.674	0.608	0.622
our system+JJ	0.559	0.614	0.675	0.612	0.620

Table 6: Results of using additional parts of speech (post hoc). Two-way classification results only

	RTE1	RTE2	RTE3	RTE4	RTE5
our system	0.586	0.605	0.672	0.614	0.610
our system-acronym	0.586	0.605	0.666	0.610	0.612
our system-WordNet	0.554	0.584	0.667	0.594	0.578
our system-edit	0.578	0.601	0.662	0.608	0.617
our system with only exact	0.540	0.564	0.626	0.565	0.570

Table 7: Ablation results, using subsets of the matching methods (post hoc)

ments. Unlike numbers and names, using these other parts of speech show much more mixed results on past RTE problems. While using lexical resources like WordNet and the Lin thesaurus can often identify that a hypothesis word like "killed" describes the same concept as the text word "died," they can also identify spurious matches that result in a non-entailment not being rejected.

Table 7 shows the results of using subsets of the matching methods. WordNet clearly yields a large performance improvement on all datasets, while acronym and edit distance yield small improvements on the earlier datasets.

4 Conclusion

We have demonstrated a simple system for recognizing non-entailment that performs well on the RTE task. In the future, we are interested in improving the contradiction detection, and investigating ways of matching other parts of speech (verbs, adjectives, and adverbs) in a way that is precise enough to provide performance improvement.

Acknowledgments

We are grateful to Cristian Danescu-Niculescu-Mizil for helpful comments on earlier drafts of this paper.

References

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In J. Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and F. d'Alché Buc, editors, *Machine Learning Challenges*, volume 3944, pages 177–190. Springer.

Cristian Danescu-Niculescu-Mizil, Lillian Lee, and Richard Ducott. 2009. Without a 'doubt'? Unsupervised discovery of downward-entailing operators. In *Proceedings of NAACL HLT*, pages 137–145.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic.

Danilo Giampiccolo, Bernardo Magnini, Elena Cabrio, Hoa Trang Dang, Ido Dagan, and Bill Dolan. 2008. The fourth PASCAL recognizing textual entailment challenge. In *Proceedings of the First Text Analysis Conference (TAC 2008)*, Gaithersburg, MD, USA.