Microsoft Research at RTE-2:
Syntactic Contributions in the Entailment Task: an implementation

Lucy Vanderwende, Arul Menezes
Microsoft Research
Redmond, WA 98075
{lucyv, arulm}@microsoft.com

Rion Snow
Computer Science Department
Stanford University
Stanford, CA 94305
rion@cs.stanford.edu

1 Introduction

The data set made available by the PASCAL Recognizing Textual Entailment Challenge provides a great opportunity to focus on the very difficult task of determining whether one sentence (the hypothesis, H) is entailed by another (the text, T).

In RTE-1 (2005), we submitted an analysis of the test data with the purpose of isolating the set of T-H pairs whose categorization could be accurately predicted based solely on syntactic cues (Vanderwende and Dolan, 2005). Furthermore, the intent of our analysis was to isolate the impact of syntactic analysis in the limit, and not of any given parser. We therefore relied on human annotators to decide whether syntactic information from an idealized parser would be sufficient to make a judgment. We found that 34% of the test items could be handled by syntax, including basic alternations. We found that 48% of the test items could be handled by syntax plus a general purpose thesaurus. Given that the test data is split evenly between entailments that are True and False, an accuracy of 74% is in principle achievable for a system with access to a general purpose thesaurus, if the system guesses randomly on what it cannot determine using syntax.

With these numbers as our goal, we have developed MENT (**M**icrosoft **ENT**ailment), a system that predicts entailment using syntactic features and a general purpose thesaurus, in addition to an overall alignment score. MENT takes as its premise that it is easier for a syntactic system to predict False entailments, following the observation in Vanderwende and Dolan (2005) that 243/800 test items could be determined to be False using syntax and thesaurus, while only roughly half as many, 147/800, could be determined as True.

2 System Overview

Similar to most other syntax-based approaches to recognizing textual entailment, we begin by representing each text and hypothesis sentence as a pair of *logical forms*. These logical forms are generated using NLPwin, a robust system for natural language parsing and generation which has been successfully used in such diverse applications as summarization, machine translation, and many others (Leskovec et al., 2005; Quirk et al., 2005). Our logical form representation may be considered equivalently as a set of triples of the form REL($node_i$, $node_j$), or as a graph of syntactic dependencies; we use both terminologies interchangeably. Our algorithm proceeds as follows:

1. Parse each sentence with the NLPwin parser, resulting in syntactic dependency graphs for the text and hypothesis sentences.
2. For each *content* node *h* in the syntactic dependency graph of the hypothesis sentence: Attempt to align that node to a node *t* in the text graph using a set of heuristics for alignment (described in Section 3)
3. Given the alignment obtained in the previous part, check the alignment against our battery of syntactic heuristics for recognizing false entailment (described in Section 4); if any match, predict that the entailment is false.
4. If no syntactic heuristic matches, back off to a lexical similarity model (described in section 5)

In addition to the typical syntactic information provided by a dependency parser, the NLPwin

parser provides an extensive number of semantic features obtained from various linguistic resources, creating a rich environment for feature engineering. For example, stemming, part-of-speech tagging, syntactic relationship identification, and semantic feature tagging are all NLPwin capabilities.

We define a *content* node to be any node *n* whose lemma is not on a small stoplist of common stop words. In addition to content vs. non-content nodes, among content nodes we distinguish between *entities* and *non-entities*: an *entity* node is any node classified by the NLPwin parser as being a proper noun, quantity, or time.

Each of the features of our system was developed from inspection of sentence pairs from the RTE-1 development and test data sets and the RTE-2 development set. For the purposes of establishing thresholds and feature weights (described in section 6), we split this data 80/20 between a training and development test set.

3 Linguistic cues for word alignment

Our syntactic heuristics for recognizing false entailment rely heavily on the correct alignment of words and multiword units between the text and hypothesis logical forms. In the notation below, we will consider *h* and *t* to be nodes in the hypothesis and text logical forms, respectively. To accomplish the task of node alignment we rely on the heuristics described below. The heuristics are applied and ambiguity is resolved in a best-first order using the algorithm described in Menezes and Richardson (2001).

Exact and Synonym match

As in Herrera et al. (2005) and others, we align a node $h \in H$ to any node $t \in T$ that has both the same part of speech and either words are identical, or belong to the same synset in WordNet, or can be found in the Bloomsbury thesaurus. Our alignment considers multiword units, including compound nouns (e.g., we align "Oscar" to "Academy Award" in, e.g., RTE-1 dev set #767, as well as verb particle constructions such as "set off" and "trigger" in RTE-1 test set #1983.

Numeric value match

The NLPwin parser assigns a normalized numeric value feature to each piece of text inferred to correspond to a numeric value; this allows us to align "6th" to "sixth"' in, e.g., Test Set #1175 and to align "a dozen"' to "twelve" in RTE-1 test set #1231.

Acronym match

Many acronyms are recognized using the lexical resource match described above; nonetheless, many acronyms are not yet found in these lexical resources. For these cases we have a specialized acronym match heuristic which aligns pairs of nodes with the following properties: If some node *h* consists of only capitalized letters (with possible interceding periods), and the letters correspond to the first characters of LEMMA(*t*) for some node $t \in T$ containing a multiword lemma, then we consider *h* and *t* to be aligned. This heuristic allows us to align "UNDP" to "United Nations Development Programme" in RTE-1 dev set #357 and "ANC" to "African National Congress" in RTE-1 test set #1300.

Derivational form match

We would like to align words which have the same root form (or have a synonym with the same root form) and which possess similar semantic meaning, but which may belong to different syntactic categories. We perform this by using a combination of the lexical resources and the derivationally-related form information contained within WordNet. Explicitly our procedure for constructing the set of derivationally-related forms for a node *h* is to take the union of all derivationally-related forms of all the synonyms of *h* (including *h* itself). In addition to the noun/verb derivationally-related forms, we detect adjective/adverb derivationally-related forms that differ only by the suffix "ly".

Country / Demonym match

As a special case of the derivational form matching, we align any matches from a gazetteer of place names, adjectival forms, and demonyms[1]. This allows us to align "Turkish" to "Turkey" in RTE-1

[1]List of adjectival forms based on the list at: http://en.wikipedia.org/wiki/List_of_demonyms

dev set #2140 and "Sweden" to "Swedish" in RTE-1 test set #1576.

Other heuristics for alignment

In addition to these heuristics, we initially implemented a hyponym match heuristic similar to that discussed in Herrera et al. (2005); however, this yielded a decrease in our system's accuracy on the training set and was thus left out of our final system. Similarly, we attempted a heuristic based on string edit distance, but this heuristic was also found to result in a decrease in accuracy in training set.

Lexical Similarity

Finally, we back off to a lexical similarity model similar to that described in Glickman (2005). For every content node $h \in H$ not already aligned by one of the heuristics above, we obtain similarity scores *sim(h, t)* from two sources (a) a dynamically acquired thesaurus resulting from word alignment of newswire text (Brockett, in prep) and (b) a similarity database that is constructed automatically from the data contained in MindNet[2]. We then compute the alignment score:

$$\text{Score}(H, T) = 1/|H| \left(\prod_{h \in H} \max_{t \in T} sim(h, t)\right)$$

where heuristic alignments have a score *sim(h, t)*=1.0. This approach is identical to that used in Glickman (2005), except that we use our alignment heuristics and similarity scores in place of their web-based estimation of lexical entailment probabilities, and that we take as our score the geometric mean of the component entailment scores rather than the un-normalized product of probabilities.

4 Recognizing false entailment

For the following heuristics, we define binary functions for the existence of each feature, such that for example if some node *h* possesses the negation feature NEG, we state that NEG (*h*) = TRUE. Similarly we have binary functions for each relation over pairs of nodes, which is defined to be true if and only if that relation is present as an edge in the syntactic dependency graph. Finally, we define the function ALIGN(*h*,*t*) to be true if the node $h \in H$ has been *hard-aligned* to the node $t \in T$ using one of the heuristics in Section 3, and false otherwise.

Unaligned entity

If some node *h* has been recognized as an entity but has not been aligned to any node *t*, we predict that the entailment is false. For example, we reject RTE-1 test Set #1863 because none of the entities "Suwariya", "20 miles", or "35" in *H* are aligned.

Negation mismatch

If any two nodes (*h*,*t*) are aligned, and one (and only one) of them is negated, we predict that the entailment is false. Negation is conveyed by the NEG feature produced by NLPwin. This heuristic allows us to detect false entailment in the example "Pertussis is not very contagious" and "...pertussis, is a highly contagious bacterial infection" in RTE-1 test set #1144.

Modal mismatch

If any two nodes (*h*,*t*) are aligned, and *t* is modified by a modal auxiliary verb (e.g, *can*, *might*, *should*, etc.) but *h* is not similarly modified, we predict that the entailment is false. Modification by a modal auxiliary verb is conveyed by the MOD feature in NLPwin. This heuristic allows us to detect false entailment between the text phrase "would constitute a threat to democracy", and the hypothesis phrase "constitutes a democratic threat" in RTE-1 test set #1203.

Antonym match

If two aligned noun nodes (h_1,t_1) are both subjects or both objects of verb nodes (h_0,t_0) in their respective sentences, i.e., REL(h_0,h_1) $\wedge$ REL(t_0,t_1) $\wedge$ REL $\in$ {SUBJ, OBJ}, then we check for an antonym match between (h_0,t_0). We construct the set of verb antonyms using WordNet; we consider the antonyms of h_0 to be the union of the antonyms of the first three senses of LEMMA(h_0), or of the nearest antonym-possessing hypernyms if those senses do not themselves have antonyms in WordNet. This heuristic allows us to detect false entailment between "Black holes can lose mass..." and "Black holes

[2] http://atom.research.microsoft.com/mnex (see Richardson et al., 1997)

can regain some of their mass..." in RTE-1 test set #1445.

In addition to the antonyms in WordNet, we also detect the prepositional antonym pairs (*before/after*, *to/from*, and *over/under*). This heuristic allows us to detect false entailment between "Profits nearly doubled to $1.8 billion." and "Profits nearly doubled from $1.8 billion." in RTE-1 dev set #1993. We report the contribution of the prepositional antonym match separately in Table 2.

Argument mismatch

For any two aligned verb nodes (h_1 ,t_1), we consider each noun child h_2 of h_1 that has a subject, object, indirect object relation, or location to h_1, i.e., $\exists$REL(h_1, h_2), REL $\in$ {SUBJ, OBJ, IND, LOC}. If there is some node t_2 such that ALIGN(h_2, t_2), but *Rel*(t_1, t_2) $\neq$ *Rel*(h_1, h_2), then we predict that the entailment is false. In Table XX, we separately report accuracy figures for verb-subj, verb-obj, and verb-locn.

As an example, consider RTE-1 dev set #1916:

```
T: U.N. officials are dis-
mayed that Aristide killed a
conference called by Prime
Minister Robert Malval.
H: Aristide kills Prime Min-
ister Robert Malval.
```

Here let (h_1, t_1) correspond to the aligned verbs with lemma *kill*, where the object of h_1 is h_2, *Prime Minister Robert Malval*, and the object of t_1 is t_2, *conference*. Since h_2 is aligned to some node t_n in the text graph, but $\neg$OBJ(t_1, t_n), the sentence pair is rejected as a false entailment.

Superlative mismatch

If some adjective node h_1 in the hypothesis is identified as a superlative, check that all of the following conditions are satisfied:

1. h_1 is aligned to some superlative t_1 in the text sentence.
2. The noun phrase h_2 modified by h_1 is aligned to the noun phrase t_2 modified by t_1.
3. Any additional modifier t_3 of the noun phrase t_2 is aligned to some modifier h_3 of h_2 in the hypothesis sentence (reverse subset match).

If any of these conditions are not satisfied, we predict that the entailment is false. This heuristic allows us to predict false entailment in the following sentence pair (RTE-1 dev set #908), where "largest media and Internet company" fails the reverse subset match to "largest company":

```
T: Time Warner is the world's
largest media and Internet
company.
H: Time Warner is the world's
largest company.
```

Conditional and counter-factual mismatch

For any pair of aligned nodes (h_1, t_1), if there exists a second pair of aligned nodes (h_2, t_2) such that the path PATH(t_1, t_2) contains the conditional relation, then PATH(h_1, h_2) must also contain the conditional relation, or else we predict that the entailment is false. Similarly, if PATH(t_1, t_2) contains a word indicative of a counter-factual , and PATH(h_1, h_2) does not contain such a counter-factual word, then predict that the entailment is false.

For example, consider the following false entailment in RTE1-dev set #60:

```
T: If a Mexican approaches
the border, he's assumed to
be trying to illegally cross.
H: Mexicans continue to ille-
gally cross border.
```

Here, "Mexican" and "cross" are aligned, and the path between them in the text contains the conditional relation, but not in the hypothesis; thus the entailment is predicted to be false.

Similarly, in RTE1-dev set #2025, join, Poland, and "European Union" are aligned, but in T, "join" is embedded in an if-clause, indicative of a counter-factual, and so the entailment is predicted to be false.

```
T: There are a lot of farmers
in Poland who worry about
their future if Poland joins
the European Union.
H: Poland joins the European
Union.
```

IS-A mismatch

We defined a collection of graph patterns that generally indicated an IS-A relationship in the logical form. These included an Appostn or Equiv relationship, an explicit "be", and use of "as", "in-

clude" and possessives under certain conditions. We then applied the following heuristic. If the hypothesis contains *ISA(h1, h2)* and both *h1*and *h2* are aligned to *t1* and *t2* respectively and neither *ISA(t1, t2)* nor *ISA(t2, t1)* are found in the text, then predict a false entailment.

5 Training feature weights

We combined RTE1 development and test sets and the RTE2 development set into a single corpus and split it randomly into a training set of 1717 sentences and a development set of 450 sentences.

For Run-2, we use only the syntactic heuristics and the alignment score. We treat the syntactic heuristics as "hard" i.e. if any heuristic fires the entailment is considered false (all of the heuristics only predict false entailment). For the remaining sentences we learn a threshold on the alignment score so as to maximize accuracy on the training set. Sentences with alignment scores better than the threshold are considered true entailments whereas those below the threshold are considered false.

For Run-1 we use the alignment score and each of the heuristics as distinct features. We also use as features sub-components of each heuristic, as well as features of the alignment such as the number of exact matches, the number of derivational matches, the number of Wordnet synonyms etc. We then trained a Maximum Entropy model (Berger et al, 1996) to learn weights for all of these features. To help prevent over-fitting, the model used a Gaussian prior over the weights. This prior was tuned to maximize development set accuracy. This gave us an improvement of approx 2.5% over the method used for Run-2.

6 Results

Table 1 displays the accuracy of our system on the training, development and RTE2-test data respectively.

	Run1	Run2
Training (1717 sents)	67.79	65.40
Dev (450 sents)	66.22	63.77
RTE2-test (800 sents)	60.25	58.50

Table 1: Summary of accuracies across different data sets for MENT with weighted features (Run1) and using "hard" syntactic heuristics (Run2).

7 Discussion

In order to better understand the drop in accuracy for MENT between the dev-test and the RTE2-test, we analyzed the accuracy of the features for determining false entailment across these different data sets in Table 2. Separately, we analyze the frequency with which these features apply in Table 3.

False Entailment Feature	Train	Dev-test	RTE-2 Test
Unaligned entity	73.92	74.32	64.49
Negation	73.52	33.33	72.72
Modal	72.72	50.00	100.00
Antonym	58.82	100.00	50.00
Preposition-Ant	100.00	100.00	100.00
Verb-subj	64.15	64.71	62.50
Verb-obj	50.00	56.25	37.50
Verb-locn	58.82	50.00	28.57
Superlative	70.00	0	100.00
Counter-factual	80.00	30.00	80.00
Conditional	80.00	0	50.00
IS-A	54.46	77.27	44.78

Table 2: Accuracy of false entailment features for Run1

False Entailment Feature	Train	Dev-test	RTE-2 Test
Unaligned entity	17.65	16.44	13.38
Negation	1.98	1.33	1.38
Modal	0.64	0.44	0.50
Antonym	0.99	0.67	0.50
Preposition-Ant	0.17	0.22	0.25
Verb-subj	3.09	3.78	4.00
Verb-obj	2.68	3.56	3.00
Verb-locn	0.99	1.78	0.88
Superlative	0.58	0.22	0.63
Counter-factual	0.87	2.22	1.88
Conditional	0.58	0.22	0.25
IS-A	6.52	4.89	8.38

Table 3: Relative frequency of false entailment features in the respective data sets for Run1

Considering the data in Tables 2 and 3, no single category appears to have suffered significantly with the exception of the IS-A category. The IS-A features applied to more sentences in RTE-2 test data than in other data sets, and the accuracy with which they applied dropped from 54.46% on the training set to 44.78% on the RTE-2 test set. Other syntactic features that dropped below 50% accu-

racy were verb-locn and verb-obj. The verb-obj feature set was barely above 50% accuracy even in the training and dev-test sets, and already was considered as having negative impact on the overall scores. Surprisingly, the accuracy of the verb-locn feature set was almost half that of the training set, while applying roughly equally frequently. Further error analysis on this category will follow.

	RUN1	
TRUTH	YES	NO
YES	268	132
NO	186	214

Table 4: Comparison of MENT entailment judgments and the Truth for the RTE-2 test set

Finally, table 4 illustrates that MENT succeeds in predicting 53.50% of the false entailments, while predicting false 43.25% of the time overall. However, MENT still over-predicts false, with 33% false negative errors. Some of these errors are due to parser error, and some may be due to over-fitting. However, at least some of false negatives are due to the lack of lexical resources that incorporate phrasal similarity (with or without syntactic information). Consider, for example, RTE-1 dev set #912, for example:

```
Rodriguez told detectives he
never touched the burning
backpack, which was loaded
with plastic pipes packed
with gunpowder and BBs.
The burning backpack con-
tained plastic pipes packed
with gunpowder and BBs.
```

Currently, MENT predicts a false entailment because "backpack" and "pipes" are both aligned, but verbs with which they are in a subject and object relationship are unaligned. A strategy for acquiring such phrasal similarity which uses distributional similarity obtained using a search engine may well prove effective in eliminating the false negatives (see Snow et al., 2006) when applied narrowly in such contexts.

Acknowledgements

The authors gratefully acknowledge the help of Chris Brockett who provided several thesauri automatically harvested from news corpora, and Chris Quirk who helped train model weights.

References

L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39–72,March.

Chris Brockett. In prep.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web Based Probabilistic Textual Entailment. In *Proceedings of PASCAL RTE 2005.*

Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. 2005. Textual Entailment Recognition Based on Dependency Analysis and WordNet. In *Proceedings of PASCAL RTE 2005.*

Jure Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. 2005. Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts. In *Proceedings of AAAI 2005*, Pittsburgh, PA.

Arul Menezes and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the Workshop on Data-driven Machine Translation at the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, pp. 39-46

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of ACL 2005*.

Rion Snow, Lucy Vanderwende and Arul Menezes. 2006. Effectively using syntax for recognizing false entailment. In *Proceedings of the HLT-NAACL 2006*.

Lucy Vanderwende and William B. Dolan. 2006. What syntax can contribute in entailment task. In *MLCW 2005*, LNAI 3944, pp. 205-216. J. Quinonero-Candela et al. (eds.). Springer-Verlag.