UCD IIRG Approach to the Textual Entailment Challenge

Eamonn Newman, Nicola Stokes, John Dunnion, Joe Carthy
Intelligent Information Retrieval Group, Department of Computer Science
University College Dublin
Ireland
{eamonn.newman, nicola.stokes, john.dunnion, joe.carthy}@ucd.ie

Abstract

This report outlines the approach taken by members of the IIRG at University College Dublin in the PASCAL Textual Entailment Challenge 2005. Our technique measures the semantic equivalence of each text/hypothesis pair by examining both linguistic and statistical features in these sentences using a decision tree classifier.

1 Introduction

Our system uses a decision tree classifier whose features include lexical, semantic and grammatical attributes of nouns, verbs and adjectives to identify an entailment relationship between a text/hypothesis pair. We generated our final classifier from the issued development sets using the C5.0 machine learning algorithm.

The features used are calculated using the Word-Net taxonomy, the VerbOcean semantic network (developed at ISI) and a Latent Semantic Indexing technique. Other features are based on the ROUGE n-gram overlap metrics and cosine similarity between the text and hypothesis.

Our most sophisticated linguistic feature finds the longest common subsequence in the entailment-pair, and then detects contradictions in the pair by examining verb semantics for the presence of synonymy, near-synonymy, negation or antonymy in the subsequence.

2 System Description

We investigated the usefulness of a number of distinct features during the development of our decision tree approach to textual entailment. Not all of these features were contributing factors in our final classification systems, but we list all of them here for the sake of completeness because some features are combinations of other atomic features. These features can be classified into two types: measures of syntactic equivalence and measures of semantic equivalence.

In addition to these measures, there is also a **task** feature which identifies the task definition from which the entailment pair was derived. This allowed the system to build separate classifiers for each task which we hoped would capture the different aspects of entailment specific to each task.

2.1 Syntactic Equivalence Features

The first syntactic equivalence features are derived using the **ROUGE metrics** (ROUGE, 2004), which were used as a means of automatically evaluating summary quality against a set of human generated summaries in the DUC 2004 evaluation workshop. The metrics provide a measure of word overlap (i.e., unigram, bigram, trigram and 4-gram), and a weighted and unweighted longest common subsequence measure. Our final feature in this class is provided by the **cosine similarity measure**, which calculates the distance (or cosine of the angle) between the text/hypothesis pair in an n-dimensional vector space.

2.2 Semantic Equivalence Features

WordNet (WordNet, 1998) was used to identify entailment between sentence pairs where corresponding synonyms are used. Words from the same synset were considered to indicate a greater likelihood of entailment. We believe that the accuracy of this feature could be greatly improved by disambiguating the sentence pair before calculating synset overlap. More specifically, in some instances multiple senses of a single term could be matched with terms in the corresponding entailment pair, which results in sentences appearing more semantically similar than they actually are.

Using a **Latent Semantic Indexing** (Deerwester et al., 1990) matrix constructed using the DUC 2004 corpus, we attempted to identify words in entailment pairs which have high cooccurrence statistics. This is an enhancement of the similarity measure given by the WordNet features, as it matches not only synonymy in the plaintext, but also uses data from other corpora to identify other latent relationships.

VerbOcean (Chklovski and Pantel, 2004) is a broad coverage lexical resource that provides fine-grained semantic relationships between verbs. These related verb pairs were gleaned from the web using lexico-syntactic patterns that captured 5 distinct verb relationships: similar–to (e.g., *escape*, *flee*), strength (e.g., *wound* is stronger than *kill*), antonymy (e.g., *win*, *lose*), enablement (e.g., *fight*, *win*), happens–before (*marry* happens before *divorce*). VerbOcean also lists relationship strengths between verb pairs. In our experiments we only use the antonym and similar–to relationships for verb semantics analysis.

Examination of the development set suggested that for a significant proportion of sentence pairs, the **longest common subsequence** [1] is largely similar to the hypothesis element. For this feature, we only examined verb semantics in the longest common subsequence of the two sentences rather than in the full sentences. An example is shown in Figure 1. There are three variations of this feature: lcs, lcs_pos and lcs_neg.

- **lcs** This feature holds one of three values

---

[1] The Longest Common Subsequence of a pair is the longest sequence of words which is common to both text and hypothesis.

---

id=1954; task=PP; judgement=FALSE
Text: *France on Saturday* flew *a planeload of United Nations aid into eastern Chad* where French soldiers prepared to deploy from their base in Abeche towards the border with Sudan's Darfur region.
Hypothesis:*France on Saturday* crashed *a planeload of United Nations aid into eastern Chad*

Figure 1: Longest Common Subsequence. Italics denote the longest common subsequence.

$\{-1, 0, 1\}$, which correspond to the presence of an antonym, no relationship, or a synonym relationship between the longest common subsequence of the text and the hypothesis sentence respectively.

- **lcs_pos** is a simpler feature which indicates the presence of a synonym relationship, zero otherwise.

- **lcs_neg** is the corollary of lcs_pos, indicating an antonym relationship, zero otherwise.

Another feature based on the longest common subsequence is **lcs+not**, which not only combines the above lcs features, but also looks for the presence of words like "not", which reverse the meaning of the sentence. Thus, for example, if an antonym and "not" occur in a sentence then this is considered to be a positive indication of entailment.

Even though lcs+not is a combination of our lcs features we still retain these simpler features as they improve entailment accuracy . We believe this to be the case because the classifier treats them as additional evidence of negative/positive entailment. It is likely that when more training data becomes available that these atomic features will not be needed and the lcs+not feature will be sufficient.

3 System Performance

Our two submitted systems are largely similar: System 1 uses all the syntactic equivalence features, the atomic lcs features and the task feature; System 2 uses the syntactic equivalence features, the composite lcs+not feature, and does not use the task feature.

This gave rise to System 1 performing much better for some tasks, but System 2 performed (marginally) better on average. This is shown in Tables 1 and 2. Our choice of features for each system was based on their performance on the second

development set, having been trained on the first development set.

	Sys 1	Sys 2	Sys 3	Sys 4
Average	0.5625**	0.5650**	0.5675**	0.5663**
CD	0.7467**	0.7400**	0.7467**	0.8467**
IE	0.5583**	0.4917	0.5167	0.5417*
IR	0.4456	0.5444*	0.4333	0.5556**
PP	0.5200	0.5600**	0.5600**	0.5000
MT	0.4750	0.5083	0.5667**	0.4083
QA	0.5154	0.5385*	0.5000	0.4846
RC	0.5714**	0.5286	0.5714**	0.5286

Table 1: Accuracy results for both classifiers. Scores marked with * are statistically significant to 95% confidence. Scores marked with ** are statistically significant to 99% confidence.

	Sys 1	Sys 2	Sys 3	Sys 4
Average	0.5917**	0.6000**	0.5818**	0.5794**
CD	0.8602**	0.7764**	0.7873**	0.7526**
IE	0.5083**	0.5260	0.4958	0.5715**
IR	0.3789	0.6130**	0.4585	0.5201
PP	0.3968	0.5006	0.5320	0.4651
MT	0.5536*	0.5130	0.5498*	0.4108
QA	0.6003**	0.5006	0.4684	0.4846
RC	0.6003**	0.5685**	0.5961**	0.5866**

Table 2: Confidence–weighted scores (CWS) for both classifiers. Scores marked with * are statistically significant to 95% confidence. Scores marked with ** are statistically significant to 99% confidence.

As already stated, when the task feature is enabled, the C5.0 algorithm uses it to make specific classifiers for each task. This seems to lead to over–fitting in some cases, e.g., IR and MT, but can help in certain cases, e.g., RC and IE.

On release of the *gold standard*, we were able to train our classifiers on both development sets, fully examine our systems, and determine which features produced the best classifier on the test data. We ran two new systems: System 3 uses all available features, and System 4 uses all features except the task feature.

Before the gold standard was available, experiments on the training sets indicated the extra features did not contribute anything to the classifiers. Consequently, we left them out to minimise noise in the data. However, when used on the full test set, we see that the accuracy scores significantly improved in some tasks (most notably, CD and PP), al-

---

id=1560; task=QA; judgement=TRUE
Text: The technological triumph known as GPS - the Global Positioning System of satellite-based navigation - was incubated in the mind of Ivan Getting.
Hypothesis: Ivan Getting invented the GPS.

id=858; task=CD; judgement=TRUE
Text: Each hour spent in a car was associated with a 6 percent increase in the likelihood of obesity and each half-mile walked per day reduced those odds by nearly 5 percent, the researchers found.
Hypothesis: The more driving you do means you're going to weigh more – the more walking means you're going to weigh less.

Figure 2: Compositional Paraphrases (misclassified by our system).

beit to the detriment of others; the average accuracy score for the systems does not vary significantly. However, there is a slight reduction in the reliability of the confidence scores assigned by the system for some tasks, indicated by lower confidence–weighting scores for Systems 3 and 4.

4 Analysis

In this section, we discuss with examples some common system errors made by our decision tree classifier. It is clear from our system description in Section 2 that the majority of our features deal with the identification of word–level, atomic paraphrase units (e.g., child = kid; eat = devour). Consequently, there are a number of examples where phrasal and compositional paraphrasing has resulted in misclassifications by our system. Some examples of this are shown in Figure 2.

Another important type of paraphrase, not addressed explicitly by our system, is the syntactic paraphrase (e.g., "I ate the cake" or "the cake was eaten by me"). However, although we didn't include a parse tree analysis in our approach, it appears that the ROUGE metrics (and to some extent the cosine metric) were an adequate means of detecting syntactic paraphrases. The position of the ROUGE features in high-level nodes in the decision tree confirms that n-gram overlap is an important aspect of textual entailment, but obviously not the full story. However, we also observed that in some cases syntactic paraphrases prevented the detection of longest common subsequences, and reduced the effectiveness of features that relied on this syntactic anal-

```
id=2028; task=QA; judgement=FALSE
Text: Besancon is the capital of France's watch and clock-
making industry and of high precision engineering.
Hypothesis: Besancon is the capital of France.


id=1964; task=PP; judgement=FALSE
Text: Under the avalanche of Italian outrage London Under-
ground has apologised and agreed to withdraw the poster.
Hypothesis: London Underground opposed to withdraw the
poster.
```

Figure 3: LCS features

```
id=868; task=CD; judgement=FALSE
Text: Several other people, including a woman and two chil-
dren, suffered injuries in the incident.
Hypothesis: Several people were slightly wounded, including a
woman and three children.
```

Figure 4: Numerical example (misclassified by our system).

ysis. Consequently, parse tree analysis and subsequent normalisation of sentence structure could be an effective solution to this problem.

Overall, our LCS–based features were critical to the classification decision; however, we did find instances where sentence pairs were misclassified by over–simplification of the textual entailment task. For example, pair 2028 in Figure 3 shows how the true meaning of the text sentence can extend beyond the longest common subsequence. In addition, pair 1964 shows how coverage limitations in the Verb-Ocean resource resulted in this example being misclassified as negative, because an antonym relationship between "agree" and "oppose" was not listed.

Finally, during our manual examination of the results we also noticed another crucial analysis component missing from our system: numerical string evaluation. An example is shown in Figure 4. Future development will focus on a normalisation method for evaluating numeric values in the entailment pair.

5   Gold Standard Quality

In general, we found that the gold standard judgements were unambiguous. However, there were some instances where external knowledge was needed to determine entailment. For example, in Figure 5 the text does not imply that the Liffey is a river (i.e., it could be a road). Although it appears that the majority of examples were chosen to avoid

such ambiguity, it does highlight the need for a formal, explicit definition of entailment. This example also highlights the fact that in a real world application the context surrounding the entailment pair will also be needed to make a full judgement, an issue that this year's Textual Entailment Challenge doesn't address.

```
id = 1538; task=QA; judgement=TRUE;
Text: Dividing the Northside of Dublin from the Southside, the
Liffey is spanned by road bridges.
Hypothesis: The Liffey flows through Dublin.
```

Figure 5: Ambiguity in gold standard classification

6   Conclusions

Our work so far shows that Textual Entailment is a very difficult task. Clearly, a larger corpus of data is required to enable a more detailed analysis of the domain. More data will also mean that we can build more accurate classifiers.

In our own particular case, the evaluation suggests that a hybrid classifier may be of some use, taking the best case classifier for each task and combining them appropriately.

References

J. R. Quinlan, 2000. *C5.0 Machine Learning Algorithm*
`http://www.rulequest.com`

Chin-Yew Lin and Ed Hovy, *Automatic Evaluation of Summaries using n-gram co–occurence statistics*, in "Proc. Document Understanding Conference (DUC)", National Institute of Standards and Technology, 2004.

George A. Miller et al., *WordNet: Lexical Database for the English Language*, Cognitive Science Laboratory, Princeton University. At `http://www.cogsci.princeton.edu/~wn`.

S. Deerwester, S. T. Dumais, G. W. Furna, T. K. Landauer and R. Harshman, *Indexing by Latent Semantic Analysis*, Journal of the American Society for Information Science, 1990.

Timothy Chklovski and Patrick Pantel, *VerbOcean: Mining the Web for Fine–Grained Semantic Verb Relations*, Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP-04), 2004. At `http://semantics.isi.edu/ocean`.