Textual Entailment Recognition Using a Linguistically–Motivated Decision Tree Classifier

Eamonn Newman¹, Nicola Stokes², John Dunnion¹, and Joe Carthy¹

School of Computer Science and Informatics, University College Dublin, Ireland {eamonn.newman, john.dunnion, joe.carthy}@ucd.ie
 NICTA Victoria Laboratory, Department of Computer Science and Software Engineering, University of Melbourne, Australia nicola.stokes@nicta.com.au

Abstract. In this paper we present a classifier for Recognising Textual Entailment (RTE) and Semantic Equivalence. We evaluate the performance of this classifier using an evaluation framework provided by the PASCAL RTE Challenge Workshop. Sentence–pairs are represented as a set of features, which are used by our decision tree classifier to determine if an entailment relationship exisits between each sentence–pair in the RTE test corpus.

1 Introduction

In this paper, we present work undertaken by the Text Summarisation group at University College Dublin on the development of a classification system for recognising Textual Entailment (TE), where a text T entails a hypothesis H if the meaning of H can be inferred from the meaning of T [1].

Automatic text summarisation has a number of distinct stages. Radev [2] describes them thus: "content identification", when the topics of the original text(s) are identified; "conceptual organization", when the concepts to be covered by the summary are selected and ordered; and "realization", the actual generation of the summary. Multi–Document Summarisation (MDS) is the generation of a single summary from multiple documents. An MDS system must consider issues such as managing conflicting contradictory sources, identifying redundant sources and information overlap, adapting to user needs, and being mindful of authors' intentions, etc. One of the most critical issues in this list is the identification of redundant information, since the fundamental objective of the summarisation process is to avoid including repetitive information in the summary at all costs.

Redundancy removal is generally only a problem which arises in MDS, because information is being collated across multiple sources related through overlapping information (especially in certain domains, such as news stories, where a topic will often be introduced in every article described in the cluster). Obviously, this is less of a problem in single—document summarisation because an author is unlikely to continually repeat themselves in a text.

J. Quiñonero-Candela et al. (Eds.): MLCW 2005, LNAI 3944, pp. 372–384, 2006. © Springer-Verlag Berlin Heidelberg 2006

Most MDS redundancy removal techniques are based on some type of word overlap comparison. While this is a somewhat effective approach, we believe that the development of a deeper semantic analysis method would improve summary quality, since shallow methods are prone to missing certain cases (e.g., negation) which would be captured by deeper methods. This was the main motivation behind our participation in the PASCAL RTE Challenge. However, we found that the evaluation framework of the workshop was insufficient for our purposes, since only certain types of information redundancy (or semantic equivalence) were represented in RTE corpora.

The rest of this paper is presented as follows: in Section 2 we review related work in both summarisation research and textual entailment in general; Section 3 provides an overview of our system, the features it uses, and how they are used to detect entailment pairs; Section 4 describes in detail results of our experiments presented at the RTE workshop; and finally, in Section 5, we discuss some future directions for our research.

2 Related Work

In this section, we will first describe some of the research recently published as a result of the RTE challenge, followed by an overview of some related research from the text summarisation community on redundancy removal, i.e. the removal of repetitive information from machine—generated summaries.

2.1 Recognising Textual Entailment and Semantic Equivalence

There are a variety of approaches that can be used to address the problem of Recognising Textual Entailment and Semantic Equivalence (RTESE), as is evident by the breadth of the applications presented at the PASCAL RTE workshop [1] and the ACL workshop on Empirical Modeling of Semantic Equivalence and Entailment [3].

Most of these systems use some sort of lexical matching, be it simple word overlap or some more complex statistical co-occurrence relation (e.g. Latent Semantic Indexing). While these systems perform better than those without lexical matching, it was widely agreed that matching at a word-level alone was not sufficient for the PASCAL corpus. Corley [4] presents an overview of similarity metrics based on WordNet concepts. They showed that a combination of WordNet similarity measures [5] with a lexical matching metric (based on the number of shared words in a sentence–pair) acheived scores on the PASCAL corpus of up to 58.9%, which is comparable with other high–ranking systems at the workshop.

A number of the systems (de Salvo Braz et al. [6]; Akhmatova [7]; Bos and Markert [8]) used logical inference in which a representation of the text and hypothesis is constructed, and then a proof of the hypothesis is derived for the text (some of these systems appealed to world knowledge (hand-coded [9]; geographical [8]), or to formal lexical resources such as WordNet).

A number of systems represented the texts as parse trees (e.g. syntactic, dependency, semantic)(Pazienza [10]; Herrera [11]). This action reduces the problem of textual entailment recognition to one of (sub-)graph matching.

Interestingly, Vanderwende [12] showed that using no more than syntactic matching, one could match up to 37% of classifications correctly. Appealing to a thesaurus yields up to 49%. This is supported by empirical evidence from Herrera et al. [11] and Marsi and Krahmer [13]. Hence, it seems that relatively simple metrics used in combination perform better than more complex, "deeper" metrics such as logical inference or the incorporation of world knowledge into the classification computation. We suggest that this is the case because deep linguistic and inferential analysis is more prone to errors due to problems arising from word sense disambiguation.

One of the top systems in the PASCAL evaluation (Raina et al. [14, 15]) used all of the methods outlined above to some degree or another. Parsed sentences are represented as logical formulae. A theorem prover is then used to find the minimum cost of "proving" that the hypothesis is entailed by the text. These costs are learned from syntactic and semantic features and resources such as WordNet.

2.2 MDS Redundancy Removal

In this section, we describe some of the similarity detection and redundancy removal techniques which have been used in various multi-document summarisation systems. Many of the techniques used have evolved from similarity measures used in areas such as Information Retrieval [16, 17].

Possibly the most well-known and successful approach to similarity detection in automatic summarisation is the SIMFINDER [18] system. SIMFINDER is a multi-document summariser which uses clustering to reduce redundancies in its summaries. The similarity of texts (paragraphs or sentences) are judged using 43 separate features, from common words to synonyms and hypernym/hyponym matching. Texts are then clustered using a learning algorithm. This algorithm selected 11 of the 43 available features for its final set of classification rules, validating the authors' claims that "more than word matching is needed for effective paragraph matching". The clusters generated by SIMFINDER have been used by Barzilay in her system (described below) and by CENTRIFUSER, where summaries are generated by selecting one sentence from each cluster.

Maximal Marginal Relevance [19] is a technique, based around the cosine similarity metric, that was was originally developed to detect diversity in a list of retrieved documents relevant to a specific query. It measures the relevance and the novelty of a document independently, and linearly combines the two measures to calculate the "marginal relevance". This technique has also been applied to multi-document summarisation research by Goldstein [20] in which MMR was used to select passages from multiple documents. More specifically, given a selection of relevant documents, MMR can be parameterised to rank passages according to certain criteria, such as whether the summary should be

very specific to a particular topic, or whether it should cover a wide range of related issues.

More recently, Allan et al. [21] describe a method that generates temporal summaries from online news streams by adding novel information to a summary as news stories describing a particular event arrive. Although this work focuses on novelty detection, this task is obviously analogous to redunancy detection and so is relevant to our current discussion. Allan et al. define two concepts of novelty and usefulness using probabilistic language models. Novelty applies to a relevant sentence which is new to the presentation, e.g. the first sentence about an event is obviously novel. Usefulness refers to all relevant sentences which have the potential to contribute to the summary. The models are based on all of the previously-seen documents. These models are then used to determine if an incoming sentence is either novel or useful. If so, then they are added to the summary. In other words, the model is trying to capture novelty based on the "probability that the later sentence could have been generated from the same language model as the earlier sentence". A second novelty model was also investigated which compares the incoming sentence to clusters of related setences in order to overcome the data sparsity problem associated with generating language models for single sentences. This model proved to be the better-performing approach of the two.

In contrast Barzilay [22, 23, 24] adopted a more linguistically—motivated approach to the measurement of semantic equivalence. Her research focussed on the generation of abstractive summaries. In particular, her technique analysed dependency graph [25] representations of sentences to identify common paraphrase units between two potentially redundant sentences in a summary. Once these paraphrases (or redundancies) have been detected this information facilitates "information fusion", and the generation of a single sentence representing the information in both sentences. This text generation technique is used by the Columbia NewsBlaster MDS system [26].

From this discussion, it can be seen that advances in the area of Recognising Textual Entailment and Semantic Equivalence would be of great benefit to the Text Summarisation community.

3 System Description

In this section, we present an overview of our Textual Entailment Recognition system, which was originally presented at the PASCAL RTE workshop. Our system uses a decision tree classifier to detect an entailment relationship between pairs of sentences which are represented using a number of difference features such as lexical, semantic and grammatical attributes of nouns, verbs and adjectives. We generated our classifier from the RTE training data using the C5.0 machine learning algorithm [27]. We chose to use C5.0 as it can be used to build a decision—tree classifier which can branch on a numeric range, as opposed to many other such algorithms, which can only work on discrete values.

The features used are calculated using the WordNet taxonomy [28], the Verb-Ocean semantic network [29] (developed at ISI) and a Latent Semantic Indexing [30, 31] technique. Other features are based on the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [32] n-gram overlap metrics and cosine similarity between the text and hypothesis.

Our most sophisticated linguistic feature finds the longest common subsequence in the sentence—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.

In addition to these measures, there is also a **task** feature which identifies the application domain from which the sentence pair was derived. This allows the system to build separate classifiers for each task in order to capture the different aspects of entailment specific to each task.

We investigated the usefulness of a number of distinct features during the development of our decision tree approach to textual entailment. These features were developed using the training part of the corpus made available for the PAS-CAL Recognising Textual Entailment Workshop [1]¹. 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. Table 1 gives a list of the features we used, and their C5.0 data types.

Table 1. Features used by decision—tree classifier. < name>indicates a tuple of related features.

Feature	data type
entails	boolean, unknown
<rouge></rouge>	continuous
<wordnet></wordnet>	continuous
LSI	continuous
cosine	continuous
<pre><verb0cean></verb0cean></pre>	continuous
negation_t	continuous
negation_h	continuous
negdiff	continuous
<lcs></lcs>	boolean
lcs+not	boolean

3.1 Sentence-Pair Features

The first of our equivalence features are derived using the **ROUGE metrics**, which were used as a means of evaluating summary quality against a set of human–generated summaries in the 2004 Document Understanding Conference workshop [35]. The metrics provide a measure of word overlap (i.e. unigram,

The corpus may be downloaded from: http://www.pascal-network.org/Challenges/RTE/Datasets

bigram, trigram and 4-gram), and a weighted and unweighted longest common subsequence measure.

In the **WordNet-based measure**, we define the similarity between two sentences as the sum of the maxmimal similarity scores between component words in WordNet, using the Hirst-St-Onge measure [28, 5]. To implement this we used Perl language Wordnet modules [33, 34] and WordNet version 2.0.

WordNet was used to identify entailment between sentence pairs where corresponding synonyms are used. Words from the same synset (set of one or more synonyms, as defined by WordNet) 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, resulting in sentences appearing more semantically similar than they actually are.

A simple method for measuring sentence similarities is to use a vector–based method such as **cosine similarity** [16]. We use a vector-space model [17] as the primary data structure in the Cosine Similarity and Latent Semantic Indexing measures. Sentences are stopped and stemmed using the Porter stemming algorithm [36], and a count of all the words used in the sentences of the corpus is calculated. This count provides us with the information to construct the termspace, an n-dimensional vector space, where n is the number of unique terms found in the corpus. With a vector representation of all of the sentences in the corpus, we can take a simple measure of the similarity of any pair of sentences by looking at the size of the angle between their vectors: the smaller the angle, the greater the similarity.

Latent Semantic Indexing [30, 31] takes as input a term—document matrix constructed in exactly the same way as for Cosine Similarity. Before applying a similarity measure between the vectors, an *information spreading* technique known as Singular Value Decomposition is applied to the matrix.

This technique scrutinises the term–document matrix for significant levels of word co–occurence and modifies magnitudes along appropriate dimensions (i.e. scores for particular words) accordingly. Thus, a sentence such as "The Iraqi leader was deposed" may have its vector representation modified with increased magnitude along dimensions corresponding to the terms "Saddam Hussein", "Baghdad" and "George W. Bush", for example.

Using a Latent Semantic Indexing matrix constructed from the DUC 2004 corpus, we attempted to identify words in entailment pairs which have high cooccurrence statistics. We took a term–document matrix of 10028 terms and converted this to a LSI matrix of 50 dimensions, using the GNU Scientfic Library for C [37].

VerbOcean is a lexical resource that provides fine–grained semantic relationships between verbs. These related verb–pairs and their relationship strengths were gleaned from the web using lexico–syntactic patterns that captured 5 distinct verb relationships:

- similar-to (e.g., escape, flee)
- strength (e.g., kill is stronger than wound)
- antonymy (e.g., win, lose)
- enablement (e.g., fight, win)
- happens-before (marry happens before divorce)

The VerbOcean online demo searches for paths between nodes in a semantic network generated from the VerbOcean data [38]. Given that the VerbOcean semantic network is not currently available for download but the verb pairs, their relationship types, and strengths are, we used this data to build our own verb-verb association matrix. We then extracted additional semantic relationships between verbs in the VerbOcean data by calculating the similarity between each verb vector pair using the cosine metric. In our experiments we only examined VerbOcean antonym and similar—to relationships when analysing verb semantics in the entailment pair; however, all VerbOcean relationships were used to generate the association matrix.

We also identify **adverbial negation** in the sentences. Adverbial negation occurs where the presence of a word (e.g., "nor", "not") modifies the meaning of the verb in the sentence. We generate three features from this information:

- negation_t counts the number of occurences of adverbial negation in the text
- negation_h counts the number of occurences of adverbial negation in the hypothesis.
- **negdiff** is the difference between negation_t and negation_h.

Examination of the development set suggested that for a significant proportion of sentence pairs, the **longest common subsequence**² is largely similar to the hypothesis element, i.e. most of the hypothesis is contained in the text 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.

- The **lcs** feature holds one of three values $\{-1,0,1\}$, 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 and lcs_neg are simpler features which indicate the presence of a synonym relationship or antonym relationship, respectively.

lcs+not is another feature based on the longest common subsequence. It combines the above lcs features and 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

² The Longest Common Subsequence of a sentence pair is the longest (not necessarily contiguous) 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.

 $\label{thm:constraint} \mbox{Hypothesis:} France \ on \ Saturday \ {\it crashed} \ a \ planeload \ of \ United \ Nations \ aid \ into \ eastern \ Chad$

Fig. 1. Longest Common Subsequence. Italics denote the longest common subsequence.

features we still retain the simpler features as it has been shown that they improve entailment accuracy.

4 Experiments

We submitted two systems to the PASCAL workshop. The systems are described below, evaluated according to the workshop criteria and this evaluation is analysed in the following section.

4.1 System Performance

Our two submitted systems differ only in the parameters they use: 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 2 and 3. 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.

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

Table 2. Accuracy results for the classifiers. Scores marked with ** are statistically significant to 99% confidence.

	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
IΕ	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

	Sys 1	Sys 2	Sys 3	Sys 4
Average	0.5917**	0.6000**	0.5818**	0.5794**
$^{\mathrm{CD}}$	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 3. Confidence—weighted scores (CWS) for the classifiers. Scores marked with ** are statistically significant to 99% confidence.

some cases, e.g., especially on the IR and PP tasks, but it can help in certain cases such as the RC and IE tasks.

To examine the effects of using all the available features, we ran two new systems: System 3 uses all available features, and System 4 uses all features except the task feature.

The training sets indicated the extra features did not contribute anything to the classifiers and since we could only submit two systems to the workshop we ran our system submissions without these features.

Subsequently, we ran further experiments to fully investigate the effect of the features on classification accuracy. We found that accuracy scores for particular tasks (most notably, CD and PP) showed a significant increase. However, the average accuracy score across all tasks does not vary significantly.

Examination of the classifications made by each system (see Table 4) show that Systems 1 and 3, the systems using the task feature, tended to be quite balanced in their classifications, i.e. they had approximately the same number of positive and negative classifications. On the other hand, Systems 2 and 4 showed a bias towards marking instances as cases of true entailment (between 75% and 85% of cases were classified as "true"). This shows that the task indicator is highly informative to the classifiers, allowing them to specialise for particular tasks and thus improve their performance.

Table 4. Precision, Recall and F1 scores on Positive and Negative Entailments	Table 4.	Precision.	Recall	and F1 sco	res on Positive	and Negative	e Entailments
--	----------	------------	--------	------------	-----------------	--------------	---------------

	Sys 1	Sys 2	Sys 3	Sys 4		
	Positive Entailment					
Precision						
Recall	0.5500	0.8200	0.5555	0.8050		
F1	0.5479	0.6563	0.5620	0.6490		
	Negative Entailment					
Precision						
	0.5425					
F1	0.5445	0.4240	0.5729	0.4302		

4.2 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 3 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.

```
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.

Fig. 2. Compositional Paraphrases (misclassified by our system)

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 analysis. 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 oversimplification 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 VerbOcean 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.

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 Underground has apologised and agreed to withdraw the poster.

Hypothesis: London Underground opposed to withdraw the poster.

Fig. 3. Longest Common Subsequence Faults

id=868; task=CD; judgement=FALSE

Text: Several other people, including a woman and two children, suffered injuries in the incident.

Hypothesis: Several people were slightly wounded, including a woman and three children.

Fig. 4. Numerical example (misclassified by our system)

5 Future Work

There are a number of planned improvements for our system. In particular, we will consider new sentence features in the detection process such as a measure of numerical equivalence between sentence pairs as illustrated in example 4, and a syntactic analysis component. Currently, our system does not have the capacity to recognise the different syntactic forms that a sentence may take.

In addition, we also intend to replicate Pantel's VerbOcean semantic network to allow us to search along paths in the network and thus increase the system's ability to detect semantically related verbs.

An empirical evaluation of other machine learning algorithms is also planned, to investigate if any other techniques would yield a better classifier than the C5.0 algorithm.

We also intend to further evaluate our RTE system by judging its performance as a module in a Multi–Document Summarisation system. We will aim to show that the identification of semantically–equivalent sentences using our RTE system improves the overall performance of the multi-document summariser.

References

- Dagan, I., Glickman, O., Magnini, B.(eds): Proceedings of the PASCAL Recognising Textual Entailment Challenge Workshop. April 11th-13th 2005, Southampton, UK.
- 2. Radev, D.: Summarisation Tutorial. SIGIR 2004. At http://www.summarization.com/sigirtutorial2004.ppt

- Dolan, B., and Dagan, I. (eds): Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. June 30th 2005, Ann Arbor, Michigan, USA.
- 4. Corley, C., and Mihalcea, R.: *Measuring the Semantic Similarity of Texts.* In Proceedings of ACL Workshop on Empirical Modelling of Semantic Equivalence and Entailment, ACL, June 2005.
- Budanitsky A. and Hirst G.: Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In Proceedings of Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics. 2001.
- de Salvo Braz, R., Girju, R., Punyakanok, V., Roth, D. and Sammons, M: An Inference Model for Semantic Entailment in Natural Language. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- 7. Akhmatova, E.: Textual Entailment Resolution via Atomic Propositions. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- 8. Bos, J. and Markert, K.: Combining Shallow and Deep NLP methods for Recognizing Textual Entailment. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A., and Stephan J.: Applying COGEX to Recognize Textual Entailment. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- Pazienza, M. T., Pennacchiotti, M., Zanzotto, F. M.: Textual Entailment as Syntactic Graph Distance. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- Herrera, J., Peñas, A., Verdejo, F.: Textual Entailment Recognition based on dependency analysis and WordNet. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- Vanderwende, L., Coughlin, D., Dolan, W.: What Syntax can Contribute in Entailment Task. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- 13. Marsi, E. and Krahmer, E.: Classification of semantic relations by humans and machines. In Proc. ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, June 2005.
- Raina, R., et al.: Robust Textual Inference using Diverse Knowledge Sources. In Proc. PASCAL Workshop on Recognising Textual Entailment, 2005.
- 15. Raina, R., Ng, A. Y., Manning, C. D.: Robust Textual Inference via Learning and Abductive Reasoning. AAAI, 2005.
- van Rijsbergen, C. J.: Information Retrieval, http://www.dcs.gla.ac.uk/Keith/ Preface.html
- 17. Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, ACM Press, 1999.
- Hatzivassiloglou, V., et al.: SIMFINDER: A Flexible Clustering Tool for Summarization. In Workshop on Automatic Summarization, NAACL, Pittsburg, USA, 2001.
- Carbonell, J., and Goldstein, J.: The use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. SIGIR, 1998, Melbourne, Australia.
- Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M.: Multi-Document Summarization by Sentence Extraction. Automatic Summarization, Proceedings of the ANLP/NAACL Workshop, April 2000, Seattle, WA

- 21. Allan, J., Gupta, R., and Khandewal, V.: *Temporal Summaries of News Topics*. In Proceedings of SIGIR 2001.
- Barzilay, R and McKeown, K. R.: Sentence Fusion for Multidocument News Summarization. Computational Linguistics, 2005.
- Barzilay, R. and Elhadad, N.: Sentence Alignment for Monolingual Comparable Corpora. Proceedings of Empirical Methods in Natural Language Processing (EMNLP), Sapporo, Japan, 2003.
- 24. Barzilay, R.: Multidocument Summarizer, PhD Thesis, 2002, Columbia University.
- 25. Melcuk, I.: Dependency Syntax: Theory and Practice, Albany, State of New York University Press.
- 26. NewsBlaster: Columbia University, 2005. http://newsblaster.cs.columbia.edu/
- 27. Quinlan, J.R.: C5.0 Machine Learning Algorithm. At http://www.rulequest.com
- 28. Miller, G. A., et al.: WordNet: Lexical Database for the English language, Cognitive Science Laboratory, Princeton University. At http://www.cogsci.princeton.edu/~wn
- Chklovski, T., Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP-04), 2004.
- Deerwester, S., Dumais, S. T., Furna, G. W., Landauer, T. K., and Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 1990.
- 31. Landauer, T.K., Foltz, P.W., Latham, D: Introduction to Latent Semantic Analysis. Discourse Processes, 1998.
- 32. Lin, C.-Y., Hovy, E.: Automatic Evaluation of Summaries using n-gram cooccurrence statistics. Proc. Document Understanding Conference (DUC), National Institute of Standards and Technology, 2004.
- 33. Patwardhan, S., Michelizzi, J., Banerjee, S., and Pedersen, T.: WordNet::Similarity Perl Module At http://search.cpan.org/dist/WordNet-Similarity/lib/WordNet/Similarity.pm
- 34. Rennie, J: WordNet::QueryData Perl Module At http://search.cpan.org/~jrennie/WordNet-QueryData-1.39/QueryData.pm
- 35. Document Understanding Conference (DUC), National Institute of Standards and Technology, USA. At http://duc.nist.gov.
- 36. Porter, M.: An Algorithm for Suffix Stripping, in Progam, vol. 14, no. 3, July 1980. At http://www.tartarus.org/~martin/PorterStemmer/def.txt.
- 37. Galassi, M., et al,: GNU Scientific Library Reference Manual (2nd Ed.)
 At http://www.gnu.org/software/gsl/
- 38. Chklovski, T., Pantel, P.: Global Path-based Refinement of Noisy Graphs Applied to Verb Semantics. In Proceedings of The Second International Joint Conference on Natural Language Processing (IJCNLP-05), Jeju Island, South Korea, October 11-13, 2005.