PASCAL

Proceedings of the First Challenge
Workshop

Recognizing Textual Entailment

11–13 April, 2005, Southampton, U.K.

Challenge Organizers

Ido Dagan (Bar Ilan University, Israel)

Oren Glickman (Bar Ilan University, Israel)

Bernardo Magnini (ITC-irst, Trento, Italy)

Recognising Textual Entailment Challenge

Workshop Program

Tuesday, 12 April 2005

09:00-09:30	**Introduction** Ido Dagan, Bernardo Magnini and Oren Glickman
09:30-09:45	**Application of the Bleu algorithm for recognising textual entailments** Diana Pérez and Enrique Alfonseca. *Universidad Aut´onoma de Madrid*
09:45-10:00	**What Syntax can Contribute in Entailment Task** Lucy Vanderwende, Deborah Coughlin and Bill Dolan. *Microsoft Research*
10:00-10:15	**Recognizing textual entailment with edit distance algorithms** Milen Kouylekov. *University of Trento* Bernardo Magnini. *ITC-irst Centro per la Ricerca Scientifica e Tecnologica*
10:15-10:30	**Textual Entailment Recognition Based on Dependency Analysis and WordNet** Jesús Herrera, Anselmo Peñas, Felisa Verdejo. *Universidad Nacional de Educación a Distancia (UNED)*
	10:30-11:00 Coffee Break
11:00-11:25	**Textual Entailment as Syntactic Graph Distance: a rule based and a SVM based approach** Maria Teresa Pazienza, Marco Pennacchiotti. *University of Rome "Tor Vergata"* Fabio Massimo Zanzotto. *University of Milano-Bicocca*
11:25-11:40	**An Inference Model for Semantic Entailment in Natural Language** Rodrigo de Salvo Braz, Roxana Girju, Vasin Punyakanok, Dan Roth and Mark Sammons *University of Illinois at Urbana-Champaign*
11:40-12:00	**Web Based Probabilistic Textual Entailment** Oren Glickman, Ido Dagan and Moshe Koppel. *Bar Ilan University*
12:00-12:30	Panel I: What constitutes textual entailment? – refining the task definition
	12:30-14:00 Lunch Break
14:00-14:15	**Textual Entailment Recognition Based on Inversion Transduction Grammars** Dekai Wu. *The Hong Kong University of Science & Technology*
14:15-14:45	**MITRE's Submissions to the EU Pascal RTE Challenge** Samuel Bayer, John Burger, Lisa Ferro, John Henderson and Alexander Yeh. *The MITRE Corporation*
14:45-15:00	**Can Shallow Predicate Argument Structures Determine Entailment?** Alina Andreevskaia, Zhuoyan Li and Sabine Bergler. *Concordia Universaity*
14:00-15:15	**VENSES – a Linguistically-Based System for Semantic Evaluation** Rodolfo Delmonte, Sara Tonelli, Marco, Aldo Piccolino Boniforti, Antonella Bristot and Emanuele Pianta *University Ca' Foscari (Venice), ITC-IRST*
15:15-15:30	**UCD IIRG Approach to the Textual Entailment Challenge** Eamonn Newman, Nicola Stokes, John Dunnion and Joe Carthy. *University College Dublin*
	15:30-16:00 Coffee Break
16:00-16:25	**Robust Textual Inference Using Diverse Knowledge Sources** Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova, Bill MacCartney, Marie-Catherine de Marneffe, Christopher Manning and Andrew Y. Ng. *Stanford University*
16:25-16:40	**Textual Entailment Resolution via Atomic Propositions** Elena Akhmatova. *Macquarie University*
16:40-17:05	**Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment** Johan Bos. *University of Edinburgh* Katja Markert. *University of Leeds*
17:05-18:00	Panel II: What's next? – planning future benchmarks

Contents

The PASCAL Recognising Textual Entailment Challenge

Ido Dagan, Oren Glickman
Computer Science Department
Bar Ilan University
Ramat Gan, Israel
{dagan,glikmao}@cs.biu.ac.il

Bernardo Magnini
ITC-irst, Centro per la Ricerca
Scientifica e Tecnologica
Trento, Italy
magnini@itc.it

Abstract

This paper describes the PASCAL Network of Excellence *Recognising Textual Entailment (RTE)* Challenge benchmark[1]. The RTE task is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred (entailed) from the other. This application-independent task is suggested as capturing major inferences about the variability of semantic expression which are commonly needed across multiple applications. The Challenge has raised noticeable attention in the research community, attracting 17 submissions from diverse groups, suggesting the generic relevance of the task.

1 Introduction

1.1 Rational

A fundamental phenomenon of natural language is the variability of semantic expression, where the same meaning can be expressed by, or inferred from, different texts. This phenomenon may be considered the dual problem of language ambiguity, together forming the many-to-many mapping between language expressions and meanings. Many natural language processing applications, such as Question Answering (QA), Information Extraction (IE), (multi-document) summarization, and machine translation (MT) evaluation, need a model for this variability phenomenon in order to recognize that a particular target meaning can be inferred from different text variants.

Even though different applications need similar models for semantic variability, the problem is often addressed in an application-oriented manner and methods are evaluated by their impact on final application performance. Consequently it becomes difficult to compare, under a generic evaluation framework, practical inference methods that were developed within different applications. Furthermore, researchers within one application area might not be aware of relevant methods that were developed in the context of another application. Overall, there seems to be a lack of a clear framework of generic task definitions and evaluations for such "applied" semantic inference, which also hampers the formation of a coherent community that addresses these problems. This situation might be confronted, for example, with the state of affairs in syntactic processing, where clear application-independent tasks, communities (and even standard conference session names) have matured.

The *Recognising Textual Entailment (RTE)* Challenge is an attempt to promote an abstract generic task that captures major semantic inference needs across applications. The task requires to recognize, given two text fragments, whether the meaning of one text can be inferred (entailed) from another text. More concretely, *textual entailment* is defined as a directional relationship between pairs of text expressions, denoted by *T* - the entailing "Text", and *H* - the entailed "Hypothesis". We say that *T entails H* if the meaning of *H* can be inferred from the meaning of *T*, as would typically be interpreted by people. This somewhat informal definition is based on (and assumes) common human understanding of language as well as common background knowledge. It is similar in spirit to evaluation of applied tasks such as Question Answering and Information Extraction, in which humans need to judge whether the target answer or relation can indeed be inferred from a given candidate text.

As in other evaluation tasks our definition of textual entailment is operational, and corresponds

[1] http://www.pascal-network.org/Challenges/RTE/

ID	TEXT	HYPOTHESIS	TASK	ENTAILMENT
1	*iTunes software has seen strong sales in Europe.*	*Strong sales for iTunes in Europe.*	IR	True
2	*Cavern Club sessions paid the Beatles £15 evenings and £5 lunchtime.*	*The Beatles perform at Cavern Club at lunchtime.*	IR	True
3	*American Airlines began laying off hundreds of flight attendants on Tuesday, after a federal judge turned aside a union's bid to block the job losses.*	*American Airlines will recall hundreds of flight attendants as it steps up the number of flights it operates.*	PP	False
4	*The two suspects belong to the 30th Street gang, which became embroiled in one of the most notorious recent crimes in Mexico: a shootout at the Guadalajara airport in May, 1993, that killed Cardinal Juan Jesus Posadas Ocampo and six others.*	*Cardinal Juan Jesus Posadas Ocampo died in 1993.*	QA	True

Table 1: Examples of Text-Hypothesis pairs

to the judgment criteria given to the annotators who decide whether this relationship holds between a given pair of texts or not. Recently there have been just a few suggestions in the literature to regard entailment recognition for texts as an applied, empirically evaluated, task (Monz and de Rijke, 2001; Condoravdi et al., 2003; Dagan and Glickman, 2004). Textual entailment is also related, of course, to formal literature about logical entailment and semantic inference. Yet, any attempt to make significant reference to this rich body of literature, and to deeply understand the relationship between the operational textual entailment definition and relevant formal notions, would be beyond the scope of the current challenge and this paper. It may be noted that from an applied empirical perspective, much of the effort is directed at recognizing meaning-entailing variability at the lexical and syntactic levels, rather than addressing relatively delicate logical issues.

It seems that major inferences, as needed by multiple applications, can indeed be cast in terms of textual entailment. For example, a QA system has to identify texts that entail a hypothesized answer. Given the question "*What does Peugeot manufacture?*", the text "*Chrétien visited Peugeot's newly renovated car factory*" entails the hypothesized answer form "Peugeot manufactures cars". Similarly, for certain Information Retrieval queries the combination of semantic concepts and relations denoted by the query should be entailed from relevant retrieved documents. In IE entailment holds between different text variants that express the same target relation. In multi-document summarization a redundant sentence, to be omitted from the summary, should be entailed from other sentences in the summary. And in MT evaluation a correct translation should be semantically equivalent to the gold standard translation, and thus both translations should entail each other. Consequently, we hypothesize that textual entailment recognition is a suitable generic task for evaluating and comparing applied semantic inference models. Eventually, such efforts can promote the development of entailment recognition "engines" which may provide useful generic modules across applications.

1.2 The challenge scope

As a first step towards the above goal we created a dataset of Text-Hypothesis (*T*-*H*) pairs of small text snippets, corresponding to the general news domain (see Table 1). Examples were manually labeled for entailment – whether *T* entails *H* or not – by human annotators, and were divided into a Development and Test datasets. Participating systems were asked to decide for each *T*-*H* pair whether *T* indeed entails *H* or not, and results were compared to the manual gold standard.

The dataset was collected with respect to different text processing applications, as detailed in the next section. Each portion of the dataset was intended to include typical *T*-*H* examples that correspond to success and failure cases of the actual applications. The collected examples represent a range of different levels of entailment reasoning, based on lexical, syntactic, logical and world knowledge, at different levels of difficulty.

The distribution of examples in this challenge has been somewhat biased to choosing non-trivial pairs, and also imposed a balance of *True* and *False* examples. For this reason, systems performances in applicative settings might be different than the figures for the challenge data, due to different distribution of examples in particular applications. Yet, the data does challenge systems to handle properly a broad range of entailment phenomena. Overall, we were aiming at an explorative rather than a competitive setting, hoping that meaningful baselines and analyses for the capabilities of current systems will be obtained.

Finally, the task definition and evaluation methodologies are clearly not mature yet. We expect them to change over time and hope that participants' contributions, observations and comments will help shaping this evolving research direction.

2 Dataset Preparation and Application Settings

The dataset of Text-Hypothesis pairs was collected by human annotators. It consists of seven subsets, which correspond to typical success and failure settings in different application, as listed below. Within each application setting the annotators selected both positive entailment examples (*True*), where *T* is judged to entail *H*, as well as negative examples (*False*), where entailment does not hold (a 50%-50% split). Typically, *T* consists of one sentence (sometimes two) while *H* was often made a shorter sentence (see Table 1). The full datasets are available for download at the Challenge website.[2]

In some cases the examples were collected using external sources, such as available datasets or systems (see Acknowledgements), while in other cases examples were collected from the Web, focusing on the general news domain. In all cases the decision as to which example pairs to include was made by the annotators. The annotators were guided to obtain a reasonable balance of different types of entailment phenomena and of levels of difficulty. Since many *T-H* pairs tend to be quite difficult to recognize, the annotators were biased to limit the proportion of difficult cases, but on the other hand to try avoiding high correlation between entailment and simple word overlap. Thus, the examples do represent a useful broad range of naturally occurring entailment factors. Yet, we cannot say that they correspond to a particular representative distribution of these factors, or of *True* vs. *False* cases, whatever such distributions might be in different settings. Thus, results on this dataset may provide useful indications of system capabilities to address various aspects of entailment, but do not predict directly the performance figures within a particular application.

It is interesting to note in retrospect that the annotators' selection policy yielded more negative examples than positive ones in the cases where *T* and *H* have a very high degree of lexical overlap. This anomaly was noticed also by Bos and Markert, Bayer et al. and Glickman et al., and affected the design or performance of their systems

2.1 Application settings

Information Retrieval (IR):
Annotators generated hypotheses (*H*) that may correspond to meaningful IR queries that express some concrete semantic relations. These queries are typically longer and more specific than a standard keyword query, and may be considered as representing a semantic-oriented variant within IR. The queries were selected by examining prominent sentences in news stories, and then submitted to a web search engine. Candidate texts (*T*) were selected from the search engine's retrieved documents, picking candidate texts that either do or do not entail the hypothesis.

Comparable Documents (CD):
Annotators identified *T-H* pairs by examining a cluster of comparable news articles that cover a common story. They examined "aligned" sentence pairs that overlap lexically, in which semantic entailment may or may not hold. Some pairs were identified on the web using Google news[3] and others taken from an available resource of aligned English sentences (see Acknowledgments). The motivation for this setting is the common use of lexical overlap as a hint for semantic overlap in

[2] http://www.pascal-network.org/Challenges/RTE/Datasets/

[3] http://news.google.com

comparable documents, e.g. for multi-document summarization.

Reading Comprehension (RC):
This task corresponds to a typical reading comprehension exercise in human language teaching, where students are asked to judge whether a particular assertion can be inferred from a given text story. The challenge annotators were asked to create such hypotheses relative to texts taken from news stories, considering a reading comprehension test for high school students.

Question Answering (QA):
Annotators used the TextMap Web Based Question Answering system available online (see Acknowledgments). The annotators were used a resource of questions from CLEF-QA (mostly) and TREC, but could also construct their own questions. For a given question, the annotators chose first a relevant text snippet (*T*) that was suggested by the system as including the correct answer. They then turned the question into an affirmative sentence with the hypothesized answer "plugged in" to form the hypothesis (*H*).

For example, given the question, "*Who is Ariel Sharon*?" and taking a candidate answer text "*Israel's Prime Minister, Ariel Sharon, visited Prague*" (*T*)*,* the hypothesis *H* is formed by turning the question into the statement *"Ariel Sharon is Israel's Prime Minister"*, producing a *True* entailment pair.

Information Extraction (IE):
This task is inspired by the Information Extraction application, adapting the setting for pairs of texts rather than a text and a structured template. For this task the annotators used an available dataset annotated for the IE relations "kill" and "birth place" produced by UIUC (see acknowledgments), as well as general news stories in which they identified manually "typical" IE relations. Given an IE relation of interest (e.g. a purchasing event), annotators identified as the text (*T*) candidate news story sentences in which the relation is suspected to hold. As a hypothesis they created a straightforward natural language formulation of the IE relation, which expresses the target relation with the particular slot variable instantiations found in the text. For example, given the information extraction task of identifying killings of civilians, and a text "*Guerrillas killed a peasant in the city of Flores.*", a hypothesis "*Guerrillas killed a civilian*" is created, producing a *True* entailment pair.

Machine Translation (MT):
Two translations of the same text, an automatic translation and a gold standard human translation (see Acknowledgements), were compared and modified in order to obtain T-H pairs. The automatic translation was alternately taken as either *T* or *H*, where a correct translation corresponds to *True* entailment. The automatic translations were sometimes grammatically adjusted, being otherwise grammatically unacceptable.

Paraphrase Acquisition (PP)
Paraphrase acquisition systems attempt to acquire pairs (or sets) of lexical-syntactic expressions that convey largely equivalent or entailing meanings. Annotators selected a text *T* from some news story which includes a certain relation, for which a paraphrase acquisition system produced a set of paraphrases (see Acknowledgements). Then they created one or several corresponding hypotheses by applying the candidate paraphrases to the original text. Correct paraphrases suggested by the system, which were applied in an appropriate context, yielded *True T-H* pairs; otherwise a *False* example was generated.

2.2 Additional Guidelines

Some additional annotation criteria and guidelines are listed below:

- Given that the text and hypothesis might originate from documents at different points in time, tense aspects are ignored.
- In principle, the hypothesis must be fully entailed by the text. Judgment would be *False* if the hypothesis includes parts that cannot be inferred from the text. However, cases in which inference is very probable (but not completely certain) are still judged at *True*. In example #4 in Table 1 one could claim that the shooting took place in 1993 and that (theoretically) the cardinal could have been just severely wounded in the shooting and has consequently died a few months later in 1994. However, this example is tagged as *True* since the context seems to imply that he actually died in 1993. To reduce the risk of unclear cases, annotators were guided to avoid vague examples for which inference has some positive probability that is not clearly very high.

- To keep the contexts in *T* and *H* self-contained annotators replaced anaphors with the appropriate reference from preceding sentences where applicable. They also often shortened the hypotheses, and sometimes the texts, to reduce complexity.

2.3 The annotation process

Each example *T-H* pair was first judged as *True/False* by the annotator that created the example. The examples were then cross-evaluated by a second judge, who received only the text and hypothesis pair, without any additional information from the original context. The annotators agreed in their judgment for roughly 80% of the examples, which corresponded to a 0.6 Kappa level (moderate agreement). The 20% of the pairs for which there was disagreement among the judges were discarded from the dataset. Furthermore, one of the organizers performed a light review of the remaining examples and eliminated about additional 13% of the original examples, which might have seemed controversial. Altogether, about 33% of the originally created examples were filtered out in this process.

The remaining examples were considered as the gold standard for evaluation, split to 567 examples in the development set and 800 in the test set, and evenly split to *True/False* examples. Our conservative selection policy aimed to create a dataset with non-controversial judgments, which will be addressed consensually by different groups. It is interesting to note that few participants have independently judged portions of the dataset and reached high agreement levels with the gold standard judgments, of 95% on all the test set (Bos and Markert), 96% on a subset of roughly a third of the test set (Vanderwende et al.) and 91% on a sample of roughly 1/8 of the development set (Bayer et al.).

3 Submissions and Results

3.1 Submission guidelines

Submitted systems were asked to tag each *T-H* pair as either *True*, predicting that entailment does hold for the pair, or as *False* otherwise. In addition, systems could optionally add a confidence score (between 0 and 1) where 0 means that the system has no confidence of the correctness of its judgment, and 1 corresponds to maximal confidence. Participating teams were allowed to submit results of up to 2 systems or runs.

The development data set was intended for any system tuning needed. It was acceptable to run automatic knowledge acquisition methods (such as synonym collection) specifically for the lexical and syntactic constructs present in the test set, as long as the methodology and procedures are general and not tuned specifically for the test data.

In order to encourage systems and methods which do not cover all entailment phenomena we allowed submission of partial coverage results, for only part of the test examples. Naturally, the decision as to on which examples the system abstains were to be done automatically by the system (with no manual involvement).

3.2 Evaluation criteria

The judgments (classifications) produced by the systems were compared to the gold standard. The percentage of matching judgments provides the accuracy of the run, i.e. the fraction of correct responses.

As a second measure, a *Confidence-Weighted Score* (*cws*, also known as Average Precision) was computed. Judgments of the test examples were sorted by their confidence (in decreasing order), calculating the following measure:

$$cws = \frac{1}{n}\sum_{i=1}^{n}\frac{\#correct-up-to-rank-i}{i}$$

where n is the number of the pairs in the test set, and i ranges over the sorted pairs.

The *Confidence-Weighted Score* ranges between 0 (no correct judgments at all) and 1 (perfect classification), and rewards the systems' ability to assign a higher confidence score to the correct judgments than to the wrong ones. Note that in the calculation of the confidence weighted score correctness is with respect to classification – i.e. a negative example, in which entailment does not hold, can be correctly classified as false. This is slightly different from the common use of average precision measures in IR and QA, in which systems rank the results by confidence of positive classification and correspondingly only true positives are considered correct.

First Author (Group)		accu-racy	cws	partial coverage		System description					
						Word overlap	Statistical lexical relations	WordNet	Syntactic matching	world knowledge	Logical inference
Akhmatova (Macquarie)		0.519	0.507			X					X
Andreevskaia (Concordia)		0.519	0.515					X	X		
		0.516	0.52								
Bayer (MITRE)		0.586	0.617				X				
		0.516	0.503	73%						X	X
Bos (Edinburgh & Leeds)		0.563	0.593			X		X		X	X
		0.555	0.586			X					
Delmonte (Venice & irst)		0.606	0.664	62%				X	X		X
Fowler (LCC)		0.551	0.56					X		X	X
Glickman (Bar Ilan)		0.586	0.572				X				
		0.53	0.535								
Herrera (UNED)		0.566	0.575			X	X		X		
		0.558	0.571			X					
Jijkoun (Amsterdam)		0.552	0.559			X	X				
		0.536	0.553			X		X			
Kouylekov (irst)		0.559	0.607			X	X		X		
		0.559	0.585								
Newman (Dublin)		0.563	0.592			X	X				
		0.565	0.6								
Perez (Madrid)		0.495	0.517			X					
		0.7	0.782	19%							
Punyakanok (UIUC)		0.561	0.569						X		
Raina (Stanford)		0.563	0.621				X	X	X		X
		0.552	0.686								
Wu (HKUST)		0.512	0.55				X		X		
		0.505	0.536								
Zanzotto (Rome-Milan)		0.524	0.557					X	X		
		0.518	0.559								

Table 2:Accuracy and cws results for the system submissions, ordered by first author. Partial coverage refers to the percentage of examples classified by the system out of the 800 test examples. (The results of the manual analysis by Vanderwende at al. (MSR) are summarized separately in the text.)

3.3 Submitted systems and results

Sixteen groups submitted the results of their systems for the challenge data, while one additional group submitted the results of a manual analysis of the dataset (Vanderwende et al., see below). As expected, the submitted systems incorporated a broad range of inferences that address various levels of textual entailment phenomena. Table 2 presents some common (crude) types of inference components, which according to our understanding, were included in the various systems.

The most basic type of inference measures the degree of word overlap between *T* and *H*, possibly including stemming, lemmatization, part of speech tagging, and applying a statistical word weighting such as idf. Interestingly, a non-participating system that operated solely at this level, using a simple decision tree trained on the development set, obtained an accuracy level of 0.568, which might reflect a knowledge-poor baseline (Rada Mihalcea, personal communication). Higher levels of lexical inference considered relationships between words that may reflect entailment, based either on statistical methods or WordNet. Next, some systems measured the degree of match between the syntactic structures of *T* and *H*, based on some distance criteria. Finally, few systems incorporated some form of "world knowledge", and a few more applied a logical prover for making the entailment inference, typically over semantically enriched representations. Different decision mechanisms were applied over the above types of knowledge, including probabilistic models, probabilistic Machine Translation models, supervised learning methods, logical inference and various specific scoring mechanisms.

Table 2 shows the results for the submitted runs. Overall system accuracies were between 50 and 60 percent and system cws scores were between 0.50 and 0.70. Since the dataset was balanced in terms of true and false examples, a system that uniformly predicts *True* (or *False*) would achieve an accuracy of 50% which constitutes a natural baseline. Another baseline is obtained by considering the distribution of results in random runs that predict *True* or *False* at random. A run with *cws>0.540* or *accuracy>0.535* is better than chance at the 0.05 level and a run with *cws>0.558* or *accuracy>0.546* is better than chance at the 0.01 level.

Unlike other system submissions, Vanderwende et al. report an interesting manual analysis of the test examples. Each example was analyzed as whether it could be classified correctly (as either *True* or *False*) by taking into account only syntactic considerations, optionally augmented by a lexical thesaurus. An "ideal" decision mechanism that is based solely on these levels of inference was assumed. Their analysis shows that 37% of the examples could (in principle) be handled by considering syntax alone, and 49% if a thesaurus is also consulted.

The Comparable Documents (CD) task stands out when observing the performance of the various systems broken down by tasks. Generally the results on the this task are significantly higher than the other tasks with results as high as 87% accuracy and cws of 0.95. This behavior might indicate that in comparable documents there is a high prior probability that seemingly matching sentences indeed convey the same meanings. We also note that that for some systems it is the success on this task which pulled the figures up from the insignificance baselines.

Our evaluation measures do not favor specifically recognition of positive entailment. A system which does well in recognizing when entailment does not hold would do just as well in terms of accuracy and cws as a system tailored to recognize true examples. In retrospect, standard measures of precision, recall and f in terms of the positive (entailing) examples would be appropriate as additional measures for this evaluation. In fact, some systems recognized only very few positive entailments (a recall between 10-30 percent). Furthermore, all systems did not perform significantly better than the f=0.67 baseline of a system which uniformly predicts true.

4 Conclusions

The PASCAL *Recognising Textual Entailment (RTE)* Challenge is an initial attempt to form a generic empirical task that captures major semantic inferences across applications. The high level of interest in the challenge, demonstrated by the submissions from 17 diverse groups and noticeable interest in the research community, suggest that textual entailment indeed captures highly relevant tasks for multiple applications.

The results obtained by the participating systems may be viewed as typical for a new and relatively difficult task (cf. for example the history of MUC benchmarks). Overall performance figures for the better systems were significantly higher than some baselines. Yet, the absolute numbers are relatively low, with small, though significant, differences between systems. Interestingly, system complexity and sophistication of inference did not correlate fully with performance, where some of the best results were obtained by rather naïve lexically-based systems. The fact that quite sophisticated inference levels were applied by some groups, with 5 systems using logical provers, provide an additional indication that applied NLP research is progressing towards deeper semantic analyses. Further refinements are needed though to obtain sufficient robustness for the Challenge types of data. Further detailed analysis of systems performance, relative to different types of examples and entailment phenomena, are likely to yield future improvements.

Being the first benchmark of its types there are several lessons for future similar efforts. Most notably, further efforts can be made to create "natural" distributions of Text-Hypothesis examples. For example, *T-H* pairs may be collected directly from the data processed by actual systems, considering their inputs and candidate outputs. An additional possibility is to collect a set of candidate texts that might entail a given single hypothesis, thus reflecting typical ranking scenarios. Data collection settings may also be focused on typical "core" semantic applications, such as QA, IE, IR and summarization. Overall, we hope that future similar benchmarks will be carried out and will help shaping clearer frameworks, and corresponding research communities, for applied research on semantic inference.

Acknowledgements

The following sources were used in the preparation of the data:

- Document Understanding Conferences (DUC) 2004 Machine Translation evaluation data, from the National Institute of Standards and Technology (NIST). http://duc.nist.gov/duc2004/
- TextMap Question Answering online demo, from the Information Sciences Institute (ISI) http://brahms.isi.edu:8080/textmap/
- Relation Recognition dataset, from University of Illinois at Urbana-Champaign. http://l2r.cs.uiuc.edu/~cogcomp/
- DIRT paraphrase database (online demo), from the University of southern California. http://www.isi.edu/~pantel/Content/Demos/demosDirt.htm
- The output of the TEASE system for extracting entailment relations and paraphrases (Szpektor et al., 2004).
- Corpus of Sentence Alignment in monolingual comparable corpora, Columbia University. http://www.cs.columbia.edu/~noemie/alignment/

We would like to thank the people and organizations that made these sources available for the challenge.

We would also like to acknowledge the people involved in creating and annotating the data: Danilo Giampiccolo, Tracy Kelly, Einat Barnoy, Alessandro Vallin, Ruthie Mandel, and Melanie Joseph.

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. We wish to thank the managers of the PASCAL Challenges program, Florence d'Alché-Buc and Michele Sebag, for their enthusiastic efforts and assistance, and Eileen Simon and John Shawe-Taylor from PASCAL for being so supportive all along.

References

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, Daniel G. Bobrow. 2003. *Entailment, intensionality and text understanding*. Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning.

Ido Dagan and Oren Glickman. 2004. *Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability*. In PASCAL workshop on Learning Methods for Text Understanding and Mining, 26 - 29 January 2004, Grenoble, France.

Christof Monz, Maarten de Rijke. 2001. Light-Weight Entailment Checking for Computational Semantics. In Proc. of the third workshop on inference in computational semantics (ICoS-3).

Szpektor, I.;Tanev, H.; Dagan, I.;Coppola, B. 2004. Scaling Web-based Acquisition of Entailment Relations. *Proceedings of Empirical Methods in Natural Language Processing (EMNLP) 2004*.

Application of the Bleu algorithm for recognising textual entailments

Diana Pérez and Enrique Alfonseca
Department of Computer Science
Universidad Autónoma de Madrid
Madrid, 28049, Spain
{diana.perez, enrique.alfonseca}@uam.es

Abstract

The BLEU algorithm has been applied to many different fields. In this paper, we explore a new possible use: the automatic recognition of textual entailments. BLEU works at the lexical level by comparing a candidate text with several reference texts in order to calculate how close the candidate text is to the references. In this case, the candidate would be the text part of the entailment and the hypothesis would be the unique reference. The algorithm achieves an accuracy around 50% that proves that it can be used as a baseline for the task of recognising entailments.

1 Introduction

In the framework of the Pascal Challenges, we are now in the position of tackling a new application: the automatic recognition of textual entailments. It is, without doubt, a complex task that, as it is first approached in this event, needs both a preliminary study to find out which are the best techniques that can be applied, and the development of new techniques specifically designed for it. Another issue is to study if a combination of shallow techniques is able to face this problem, or whether it will be necessary to go into deeper techniques. If so, it will be interesting to know what the advantages of deep analyses are, and how the results differ from just using shallow techniques.

In the current situation, textual entailment is defined as the relation between two expressions, a text (T), and something entailed by T, called an entailment hypothesis (H). Our approach consists in using the BLEU algorithm (Papineni et al., 2001), that works at the lexical level, to compare the entailing text (T) and the hypothesis (H). Next, the entailment will be judged as true or false according to BLEU's output.

Once the algorithm is applied, we have seen that the results confirm the use of BLEU as baseline for the automatic recognition of textual entailments. Furthermore, they show how a shallow technique can reach around a 50% of accuracy.

The article is organised as follows: Section 2 explains how BLEU works in general, next Section 3 details the application of this algorithm for recognising entailments and gives the results achieved using the development and test sets. Finally, Section 4 ends with a discussion about the contribution that BLEU can make to this task and as future work, how far it can be improved to increase its accuracy.

2 The BLEU Algorithm

The BLEU (BiLingual Evaluation Understudy) algorithm was created by (Papineni et al., 2001) as a procedure to rank systems according to how well they translate texts from one language to another. Basically, the algorithm looks for n-gram coincidences between a candidate text (the automatically produced translation) and a set of reference texts (the human-made translations).

The pseudocode of BLEU is as follows:

- For several values of N (typically from 1 to 4), calculate the percentage of n-grams from the candidate translation which appears in any of the human translations. The frequency of each n-gram is limited to the maximum frequency with which it appears in any reference.

- Combine the marks obtained for each value of N, as a weighted linear average.

- Apply a brevity factor to penalise short candidate texts (which may have n-grams in common with the references, but may be incomplete). If the candidate is shorter than the references, this factor is calculated as the ratio between the length of the candidate text and the length of the reference which has the most similar length.

It can be seen from this pseudocode that BLEU is not only a keyword matching method between pairs of texts. It takes into account several other factors that make it more robust:

- It calculates the length of the text in comparison with the lengths of reference texts. This is because the candidate text should be similar to the reference texts (if the translation has been well done). Therefore, the fact that the candidate text is shorter than the reference texts is considered an indicative of a poor quality translation and thus, BLEU penalises it with a Brevity Penalty factor that lowers the score.

- The measure of similarity can be considered as a precision value that calculates how many of the n-grams from the candidate appear in the reference texts. This value has been modified, as the number of occurrences of an n-gram in the candidate text is clipped at the maximum number of occurrences it has in the reference texts. Therefore, an n-gram that is repeated very often in the candidate text will not increment the score if it only appears a few times in the references.

- The final score is the result of the weighted sum of the logarithms of the different values of the precision, for n varying from 1 to 4. It is not interesting to try higher values of n since coincidences longer than four-grams are very unusual.

BLEU's output is always a number between 0 and 1. This value indicates how similar the candidate and reference texts are. In fact, the closer the value is to 1, the more similar they are. (Papineni et al., 2001) report a correlation above 96% when comparing BLEU's scores with the human-made scores. This algorithm has also been applied to evaluate text summarisation systems (Lin and Hovy, 2003) and to help in the assessment of open-ended questions (Alfonseca and Pérez, 2004).

3 Application of BLEU for recognising textual entailments

For recognising entailments using BLEU, the first decision is to choose whether the candidate text should be considered as the text part of the entailment (T) or as the hypothesis (and, as a consequence whether the reference text should be considered as the H or the T part). In order to make this choice, we did a first experiment in which we considered the T part as the reference and the H as the candidate. This setting has the advantage that the T part is usually longer than the H part and thus the reference would contain more information that the candidate. It could help the BLEU's comparison process since the quality of the references is crucial and in this case, the number of them has been dramatically reduced to only one (when in the rest of the applications of BLEU the number of references is always higher).

Then, the algorithm was applied according to its pseudocode (see Section 2). The output of BLEU was taken as the confidence score and it was also used to give a TRUE or FALSE value to each entailment pair. We performed an optimisation procedure for the development set that chose the best threshold according to the percentage of success of correctly recognised entailments. The value obtained was 0.157. Thus, if the BLEU's output is higher than 0.157 the entailment is marked as TRUE, otherwise as FALSE.

The results achieved are gathered in Table 1. Besides, in order to confirm that this setting was truly better, we repeated the experiment this time choosing the T part of the entailment as the candidate and the H part as the reference. The results are shown in Table 2. In this case, the best threshold has been 0.1.

Task	NTE	A	NTR	NFR	NTW	NFW
CD	98	77%	39	36	12	11
IE	70	44%	16	15	20	19
MT	54	52%	18	10	17	9
QA	90	41%	9	28	17	36
RC	103	51%	30	23	28	22
PP	82	57%	22	25	18	17
IR	70	44%	10	21	14	25
Total	567	53%	144	158	126	139

Table 1: Results for the development sets considering the T part of the entailment as the reference text (threshold = 0.157). Columns indicate: task id; number of entailments (NTE); accuracy (A); number of entailments correctly judged as true (NTR); number of entailments correctly judged as false (NFR); number of entailments incorrectly judged as true (NTW); and, number of entailments incorrectly judged as false (NFW).

Task	NTE	A	NTR	NFR	NTW	NFW
CD	98	72%	40	31	17	10
IE	70	50%	23	12	23	12
MT	54	52%	21	7	20	6
QA	90	50%	22	23	22	23
RC	103	50%	33	19	32	19
PP	82	60%	25	24	19	14
IR	70	41%	8	21	14	27
Total	567	54%	172	137	147	111

Table 2: Results for the development sets considering the T part of the entailment as the candidate text (threshold = 0.1). Columns indicate: task id; number of entailments (NTE); accuracy (A); number of entailments correctly judged as true (NTR); number of entailments correctly judged as false (NFR); number of entailments incorrectly judged as true (NTW); and, number of entailments incorrectly judged as false (NFW).

This is the value that has been fixed as threshold for the test set.

It is important to highlight that the average correlation achieved was 54%. Moreover, it reached a 72% accuracy for the Comparable Document (CD) task as it could be expected since Bleu's strength relies on making comparisons among texts in which the lexical level is the most important. For example, the snippet of the development test with identifier 583, whose T part is *"While civilians ran for cover or fled to the countryside, Russian forces were seen edging their artillery guns closer to Grozny, and Chechen fighters were offering little resistance"* and H part is *"Grozny is the capital of Chechnya"*, is an ideal example case for BLEU. This is because only the word Grozny is present both in the T and H texts. BLEU will mark it as false since there is no n-gram co-occurrence between both texts.

On the other hand, BLEU cannot deal examples in which the crucial point to correctly recognise the entailment is at the syntactical or semantics level. For example, those cases in which the T and H parts are the same except for just one word that reverses the whole meaning of the text. For example, the snippet 148 of the development set, whose T part is *"The Philippine Stock Exchange Composite Index rose 0.1 percent to 1573.65"* and the H part is *"The Philippine Stock Exchange Composite Index dropped."* This is a very difficult case for BLEU. It will be misleading since BLEU would consider that both T and H are saying something very similar, while in fact, the only words that are different in both texts, *"rose"* and *"dropped"*, are antonyms, making the entailment FALSE.

It can also be seen how the results contradict the insight that the best setting would be to have the T part as the reference text. In fact, the results are not so much different for both configurations. A possible reason for this could be that all cases when BLEU was misled into believing that the entailment was true (because the T and H parts have many n-grams in common except the one that is the crucial to solve the entailment) are still problematic. It should be noticed that BLEU, irrespectively of the consideration of the texts as T or H, cannot deal with these cases.

The results for the test set confirm the same conclusions drawn for the development tests. In fact, for the first run in which BLEU was used for all the tasks, it achieved a 52% confidence-weighted score and a 50% accuracy. See Table 3 for details.

As can be seen, not only the overall performance continues being similar to accuracy obtained with the development test. Also the best task for the test set keeps being the CD. To highlight this fact, we

TASK	CWS	A
CD	0.7823	0.7000
IE	0.5334	0.5000
MT	0.2851	0.3750
QA	0.3296	0.4231
RC	0.4444	0.4571
PP	0.6023	0.4600
IR	0.4804	0.4889
TOTAL	0.5168	0.4950

Table 3: Results for the test set (threshold = 0.1). Columns indicate: task id; confidence-weighted score or average precision (CWS); and, the accuracy (A).

implemented a preliminary step of the algorithm in which there was a filter for the CD snippets, and only they were processed by BLEU. In this way, we created a second run with the CD set that achieved a CWS of 78% and a 70% accuracy. This high result indicates that, although, in general, BLEU should only be considered as a baseline for recognising textual entailments, in the case of CD, it can be used as a stand-alone system.

4 Conclusion and future work

Some conclusions can be drawn from the experiments previously described:

- BLEU can be used as a baseline for the task of recognising entailments, considering the candidate text as T and the reference text as the H part of the entailment, since it has achieved an accuracy above 50%.

- BLEU's results depend greatly on the task considered. For example, for the Comparable Documents (CD) task it reaches its maximum value (77%) and for Information Retrieval (IR) the lowest (41%).

- BLEU has a slight tendency to consider a hypothesis as TRUE. In 319 out of 567 pairs, BLEU said the entailment was true. Out of these, it was right in 172 cases, and it was wrong in 147 cases. On the other hand, there were only 111 false negatives.

It is also interesting to observe that, although the origin of BLEU is to evaluate MT systems, the results for the MT task are not specially higher. The reason for that could be that BLEU is not being used here to compare a human-made translation to a computer-made translation, but two different sentences which contain an entailment expression, but which are not alternative translations of the same text in a different language.

The main limit of BLEU is that it does not use any semantic information and, thus, sentences with many words in common but with a different meaning will not be correctly judged. For instance, if T is *"The German officer killed the English student"* and H is *"The English students killed the German Officer"*, BLEU will consider the entailment hypothesis as TRUE, while it is FALSE.

It would be interesting, as future work, to complement the use of BLEU with some kind of syntactic processing and some treatment of synonyms and antonyms. For example, if BLEU were combined with a parser translating all sentences from passive to active and allowed the comparison by syntactic categories such as subject, direct object, indirect object, etc., it would be able to recognise more entailments.

References

E. Alfonseca and D. Pérez. 2004. Automatic assessment of short questions with a BLEU-inspired algorithm and shallow nlp. In *Advances in Natural Language Processing*, volume 3230 of *Lecture Notes in Computer Science*, pages 25–35. Springer Verlag.

C. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003)*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. Research report, IBM.

What Syntax can Contribute in Entailment Task

Lucy Vanderwende, Deborah Coughlin, Bill Dolan

Microsoft Research
Redmond, WA 98052
{lucyv; deborahc; billdol}@microsoft.com

Abstract

We describe our submission to the PASCAL Recognizing Textual Entailment Challenge which attempts to isolate the set of T-H pairs whose categorization can be accurately predicted based solely on syntactic cues. Two human annotators examined each pair, showing that a surprisingly large proportion of the data – 37% of the test items – can be handled with syntax alone, while adding information from a general-purpose thesaurus increases this to 49%.

1 Introduction

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

Our goal was to isolate the class of T-H pairs whose categorization can be accurately predicted based solely on syntactic cues. Human annotators made this judgment; we wanted to abstract away from the analysis errors that any specific parsing system would inevitably introduce. This work is part of a larger ablation study aimed at measuring the impact of various NLP components on entailment and paraphrase.

We have chosen to provide a partial submission that addresses the following question: what proportion of the entailments in the PASCAL test set could be solved using a robust parser? We are encouraged that other entrants chose to focus on different baselines, specifically those involving lexical matching and edit distance. Collectively, these baselines should establish what the minimal system requirements might be for addressing the textual entailment task.

2 Details of MSR submission

Various parsers providing constituent level analysis are now available to the research community, and state-of-the-art parsers have reported accuracy of between 89% and 90.1% F-measure (Collins and Duffy, 2002, Henderson 2004, and see Ringger et al., 2004, for a non-treebank parser). There are also efforts to produce parsers that assign argument structure (Gildea and Jurafsky, 2002, and for example, Hacioglu et al., 2004). With these developments, we feel that syntax can be defined broadly to include such phenomena as argument assignment, intra-sentential pronoun anaphora resolution, and a set of alternations to establish equivalence on structural grounds.

In order to establish a baseline for the entailment task that reflects what an idealized parser could accomplish, regardless of what any specific parser can do, we annotated the test set as follows. Two human annotators evaluated each T-H pair, deciding whether the entailment was:

- True by Syntax,
- False by Syntax,
- Not Syntax,
- Can’t Decide

Additionally, we allowed the annotators to indicate whether recourse to information in a general purpose thesaurus entry would allow a pair to be judged True or False. Both annotators were skilled linguists, and could be expected to determine what an idealized syntactic parser could accomplish. We should note at this point that it could prove impossible to automate the judgment process described in this paper; the rules-of-thumb used by the annotators to make True of False judgments may turn out to be incompatible with an operational system.

We found that 37% of the test items can be handled by syntax, broadly defined; 49% of the test items can be handled by syntax plus a general purpose thesaurus. The results of this experiment are summarized in table 1:

	Without thesaurus	Using thesaurus
True	78 (10%)	147 (18%)
False	217 (27%)	244 (31%)
Not syntax	505 (63%)	409 (51%)

Table 1: Summary of MSR partial submission; Run1 is without thesaurus, Run2 is with thesaurus

Overall, inter-annotator agreement was 72%. Where there were disagreements, the annotators jointly decided which judgment was most appropriate in order to annotate all test items. Of the disagreements, 60% were between False and Not-Syntax, and 25% between True and Not-Syntax; the remainder of the differences were either annotation errors or where one or both chose Can't Decide. This confirms our anecdotal experience that it is easier to decide when syntax can be expected to return True, and that the annotators were uncertain when to assign False. In some cases, there are good syntactic clues for assigning False, which is why we designed the evaluation to force a choice between True, False, and Not-Syntax. But in many cases, it is the absence of syntactic equivalence or parallelism rather than a violation that results in a judgment of False, and most of the disagreements centered on these cases.

3 Results of Partial Submission

Our test results are not comparable to those of other systems, since obviously, our runs were produced by human annotators. In this section, we only want to briefly call attention to those test items where there was a discrepancy between our adjudicated human annotation and those provided as gold standard. It is worth mentioning that we believe the task is well-defined; for the 295 test items returned in Run1 of our submission, 284 matched the judgment provided as gold standard, so that our inter-annotator agreement with the test set is 96%.

In Run1 (using an idealized parser, but no thesaurus), there were 11 discrepancies. Of the 3 cases where we judged the test item to be True but the gold standard for the item is False, one is clearly an annotation error (despite having two annotators!) and two are examples of strict inclusion, which we allowed as entailments but the data set does not (test items 1839 and 2077); see (1).

(1) (pair id="2077", value="FALSE", task="QA")
<T> They are made from the dust of four of Jupiter's tiniest moons.
<H> Jupiter has four moons.

More difficult to characterize as a group are the 8 cases where we judged the test item to be False but the gold standard for the item is True (although 5/8 are from the QA section) The test items in question are: 1335, 1472, 1487, 1553, 1584, 1586, 1634, and 1682. It does appear to us that more knowledge is needed to judge these items than simply what is provided in the Text and Hypothesis, and these items should be removed from the data set accordingly since pairs for which there was disagreement among the judges were discarded. Item 1634 is a representative example.

(2) (pair id="1634", value="TRUE", task="IE")
<T> William Leonard Jennings sobbed loudly as was charged with killing his 3-year-old son, Stephen, who was last seen alive on Dec. 12, 1962.
<H> William Leonard Jennings killed his 3-year-old son, Stephen.

4 Requirements for a syntax-based system

There are many examples where predicate-argument assignment will give clear evidence for the judgment. (3a) and (3b) provide a good illustration:

(3) <T> Latvia, for instance, is the lowest-ranked team in the field but defeated World Cup semifi-

nalist Turkey in a playoff to qualify for the final 16 of Euro 2004.

(3a) <H> Turkey is defeated by Latvia.
(pair id="1897", value="TRUE", task="IE")

(3b) <H> Latvia is defeated by Turkey.
(pair id="1896", value="FALSE", task="IE")

4.1 Syntactic Alternations

By far the most frequent alternation between Text and Hypothesis that a system needs to identify is an appositive construction promoted to main clause. This alternation accounted for approximately 24% of the data.

(4) (pair id="760", value="TRUE", task="CD")
<T> The Alameda Central, west of the Zocalo, was created in 1592.
<H> The Alameda Central is west of the Zocalo.

Examples of other alternations that need to be identified are: nominalization → tensed clause (*Schroeder's election → Shroeder was elected*), shown in (5), and finite → non-finite construction (*where he was surfing → while surfing*), shown in (6).

(5) (pair id="315", value="TRUE", task="IR")
<T> The debacle marked a new low in the erosion of the SPD's popularity, which began shortly after Mr Schroeder's election in 1998.
<H> Schroeder was elected in 1998.

(6) (pair id="1041", value="TRUE", task="RC")
<T> A 30-year-old man has been killed in a shark attack at a surfing beach near Perth in West Australia where he was surfing with four other people.
<H> A 30-year-old man was killed in a shark attack while surfing.

4.2 Establishing False Entailment

We found two main categories of T-H pairs that we judged to be False: False, where there was a violation of a syntactic nature, and False, where there was no syntactic structure shared by the T-H pair. Although we can annotate this by hand, we are unsure whether it would be possible to create a system to automatically detect the absence of syntactic overlap, though the main verb in the Hypothesis should be the initial area of focus.

Examples of judging False by violation of syntax are those in which the Subject and Verb align (with or without thesaurus), but the Object does not, as in (7):

(7) (pair id="103", value="FALSE", task="IR")
<T> The White House ignores Zinni's opposition to the Iraq War.
<H> White House ignores the threat of attack.

The following examples illustrate an absence of shared syntactic structure in the major argument positions. In (8), the entailment is judged False since *baby girl* is not the subject of any verb of *buying*, nor is *ambulance* the object of any verb of *buying*; additionally, there is no mention of *buying* in T at all. In (9), the entailment is judged False because there is no mention of *Douglas Hacking* in the Text, nor any mention of *physician*. While a system using lexical matching might well rule the second example False, there are enough lexical matches in the former that a system using syntax is likely required.

(8) (pair id="2179", value="FALSE", task="RC")
<T> An ambulance crew responding to an anonymous call found a 3-week-old baby girl in a rundown house Monday, two days after she was snatched from her mother at a Melbourne shopping mall.
<H> A baby girl bought an ambulance at a Melbourne shopping mall.

(9) (pair id="2169", value="FALSE", task="CD")
<T> Scott and Lance Hacking talked with their younger brother at the hospital July 24.
<H>Douglas and Scott Hacking are physicians.

5 Interesting "Not Syntax" Examples

The number of examples that can be handled using syntax, broadly defined, is significant, but more than 50% were judged to be outside the realm of syntax, even allowing for the use of a thesaurus.
Some test items exhibited phrasal-level synonymy, which the annotators did not expect would be available in a general purpose thesaurus. Consider, *X bring together Y* and *Y participate in X* in (10):

(10) (pair id="287", value="TRUE", task="IR")

<T> The G8 summit, held June 8-10, brought together leaders of the world's major industrial democracies, including Canada, France, Germany, Italy, Japan, Russia, United Kingdom, European Union and United States.

<H>Canada, France, Germany, Italy, Japan, Russia, United Kingdom and European Union participated in the G8 summit.

There are some examples with apparent alternation, but the alternation cannot be supported by syntax. Consider *three-day* and *last three days* in the following example:

(11) (pair id="294", value="TRUE", task="IR")

<T> The three-day G8 summit will take place in Scotland.

<H> The G8 summit will last three days.

In other cases, the annotators considered that there were too many alternations and thesaurus replacements necessary to confidently say that syntax could be used. Consider the following example, where *more than half* has to align with *many*, *saying* aligns with *thinking*, and *not worth fighting* aligns with *necessary*.

(12) (pair id="306", value="TRUE", task="IR")

<T> The poll, for the first time, has more than half of Americans, 52 percent, saying the war in Iraq was not worth fighting.

<H> Many Americans don't think the war in Iraq was necessary.

6 Discussion and Conclusion

Our goal is to contribute a baseline consisting of a system which uses an idealized parser, broadly defined, that can detect alternations, and optionally has access to a general purpose thesaurus. In order to explore what is possible, in principle, we used two human annotators and resolved their disagreements to produce a partial submission. It is interesting to note that the task is well-defined; for the 295 test items returned in our submission (without thesaurus), 284 matched the judgment provided as gold standard, so that our inter-annotator agreement is 96%.

A syntax-based system can account for 37% of the test items, and, with the addition of information from a general purpose thesaurus, 49%. This finding is promising, though we expect the numbers to decrease subject to an implementation with a real-world parser and set of matching rules. We also are keen to compare our baseline results with those obtained by the systems using lexical matching and edit distance, as we expect that some of the items that can be handled by syntax alone could also be accounted for by these simpler methods.

We hope that the challenge workshop is well served by offering these baselines, as it is clear to us that more than half of the test items represent an opportunity to work on very interesting entailment and paraphrase problems.

Acknowledgements

The authors wish to thank the organizers of the RTE Challenge and PASCAL for making creating this dataset and making it available. The authors also thank the Butler Hill Group for designing and maintaining the annotation environment.

References

Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. *Proceedings of ACL 2002*, Philadelphia, PA.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245-288.

Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin, and Daniel Jurafsky, 2004. Semantic Role Labeling by Tagging Syntactic Chunks. *Proceedings of the Eighth Conference on Natural Language Learning (CONLL-2004)*, Boston, MA, May 6-7.

James Henderson. 2004. Discriminative training of a neural network statistical parser. *Proceedings of ACL 2004*, Barcelona, Spain.

Eric Ringger, Robert C. Moore, Eugene Charniak, Lucy Vanderwende, and Hisami Suzuki. 2004. Using the Penn Treebank to Evaluate Non-Treebank Parsers. *Proceedings of the 2004 Language Resources and Evaluation Conference (LREC)*. Lisbon, Portugal.

Recognizing Textual Entailment with Tree Edit Distance Algorithms

Milen Kouylekov^{1,2} **and Bernardo Magnini**^{2}
ITC-irst, Centro per la Ricerca Scientifica e Tecnologica ^{1}
University of Trento^{2}
38050, Povo, Trento, Italy
milen@kouylekov.net,magnini@itc.it

Abstract

This paper summarizes ITC-irst participation in the PASCAL challenge on Recognizing Textual Entailment (RTE). Given a pair of texts (the *text and the* hypothesis), the core of the approach we present is a tree edit distance algorithm applied on the dependency trees of both the text and the hypothesis. If the distance (i.e. the cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then we assign an entailment relation between the two texts.

1 Introduction

The problem of language variability (i.e. the fact that the same information can be expressed with different words and syntactic constructs) has been attracting a lot of interest during the years and it poses significant issues in front of systems aimed at natural language understanding. The example below shows that recognizing the equivalence of the statements *came in power*, *was prime-minister* and *stepped in as prime-minister* is a challenging problem.

- Ivan Kostov came in power in 1997.
- Ivan Kostov was prime-minister of Bulgaria from 1997 to 2001.
- Ivan Kostov stepped in as prime-minister 6 months after the December 1996 riots in Bulgaria.

While the language variability problem is well known in Computational Linguistics, a general unifying framework has been proposed only recently in (Dagan and Glickman 2004). In this approach, language variability is addressed by defining the notion of *entailment* as a relation that holds between two language expressions (i.e. a text T and an hypothesis H) if the meaning of H as interpreted in the context of T, can be inferred from the meaning of T. The entailment relation is directional as the meaning of one expression can entail the meaning of the other, while the opposite may not.

For our participation in the Pascal RTE Challenge we designed a system based on the intuition that the probability of an entailment relation between T and H is related to the ability to show that the whole content of H can be mapped into the content of T. The more straightforward the mapping can be established, the more probable is the entailment relation. Since a mapping can be described as the sequence of editing operations needed to transform T into H, where each edit operation has a cost associated with it, we assign an entailment relation if the overall cost of the transformation is below a certain threshold, empirically estimated on the training data.

The paper is organized as follows. Section 2 presents the Tree Edit Distance algorithm we have adopted and its application to dependency trees. Section 3 describes the system which participated at the RTE challenge and in Section 4 we present and discuss the results we have obtained.

2 Tree Edit Distance on Dependency Trees

We adopted a tree edit distance algorithm applied to the syntactic representations (i.e. dependency trees) of both T and H. A similar use of tree edit distance has been presented by (Punyakanok et al. 2004) for a Question Answering system, showing that the technique outperforms a simple bag-of-word approach. While the cost function presented in (Punyakanok et al. 2004) is quite simple, for the RTE challenge we tried to elaborate more complex and task specific measures.

According to our approach, T entails H if there exists a sequence of transformations applied to T such that we can obtain H with an overall cost below a certain threshold. The underlying assumption is that pairs between which an entailment relation holds have a low cost of transformation. The kind of transformations we can apply (i.e. deletion, insertion and substitution) are determined by a set of predefined entailment rules, which also determine a cost for each editing operation.

We have implemented the tree edit distance algorithm described in (Zhang and Shasha 1990) and applied to the dependency trees derived from T and H. Edit operations are defined at the level of single nodes of the dependency tree (i.e. transformations on subtrees are not allowed in the current implementation). Since the (Zhang and Shasha 1990) algorithm does not consider labels on edges, while dependency trees provide them, each dependency relation R from a node A to a node B has been re-written as a complex label B-R concatenating the name of the destination node and the name of the relation. All nodes except the root of the tree are relabeled in such way. The algorithm is directional: we aim to find the better (i.e. less costly) sequence of edit operation that transform T (the source) into H (the target). According to the constraints described above, the following transformations are allowed:

- **Insertion**: insert a node from the dependency tree of H into the dependency tree of T. When a node is inserted it is attached with the dependency relation of the source label.

- **Deletion**: delete a node N from the dependency tree of T. When N is deleted all its children are attached to the parent of N. It is not required to explicitly delete the children of N as they are going to be either deleted or substituted on a following step.

- **Substitution**: change the label of a node N1 in the source tree into a label of a node N2 of the target tree. Substitution is allowed only if the two nodes share the same part-of-speech. In case of substitution the relation attached to the substituted node is changed with the relation of the new node.

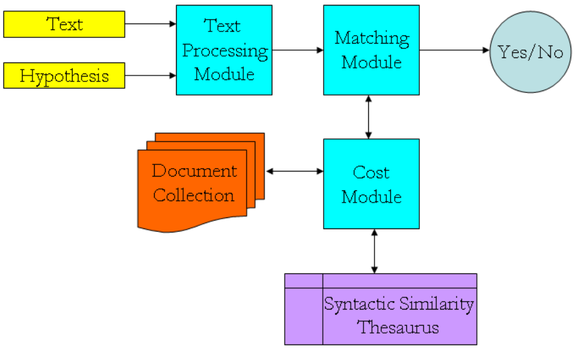


Figure 1: System architecture

3 System Architecture

The system is composed by the following modules, showed in Figure 1: (i) a text processing module, for the preprocessing of the input T/H pair; (ii) a matching module, which performs the mapping between T and H; (iii) a cost module, which computes the costs of the edit operations.

3.1 Text processing module

The *text processing module* creates a syntactic representation of a T/H pair and relies on a sentence splitter and a syntactic parser. For sentence splitting we used the Maximum entropy sentence splitter *MXTerm* (Ratnaparkhi 1996). For parsing we used *Minipar*, a principle-based English parser (Lin 1998) which has high processing speed and good precision.

3.2 Matching module

The *matching module* finds the best sequence of edit operations between the dependency trees obtained from T and H. It implements the edit distance algorithm described in Section 2. The module makes requests to the *cost module* to receive the cost of the edit operations needed to transform T into H.

3.3 Cost module

The *cost module* returns the cost of an edit operation between tree nodes. To estimate such cost, we define a weight of each single word representing its relevance through the *inverse document frequency (idf)*, a measure commonly used in *Information Retrieval.* If *N* is the number of documents in a text collection and N_w is the number of documents of the collection that contain word *w* then the *idf* of this word is given by the formula:

$$idf(w) = \log \frac{N}{N_w} \tag{1}$$

The weight of the *insertion* operation is the *idf* of the inserted word. The most frequent words (e.g. stop words) have a zero cost of insertion. In the current version of the system we are still not able to implement a good model that estimates the cost of the deletion operation. In order not to punish pairs with short contents of T we set the cost of deletion to 0. To determine the cost of substitution we used a dependency based thesaurus available at *http://www.cs.ualberta.ca/l̃indek/downloads.htm.* For each word, the thesaurus lists up to 200 most similar words and their similarities. The cost of a *substitution* is calculated by the following formula:

$$subs(w_1, w_2) = ins(w_2) * (1 - sim(w_1, w_2)) \tag{2}$$

where w_1 is the word from T that is being replaced by the word w_2 from H and $sim(w_1, w_2)$ is the similarity between w_1 and w_2 in the thesaurus multiplied by the similarity between the corresponding relations. The similarity between relations is stored in a database of relation similarities obtained by comparing dependency relations from a parsed local corpus. The similarities have values from 1 (very similar) to 0 (not similar). If there is no similarity, the cost of substitution is equal to the cost of inserting the word *w2*.

3.4 Global Entailment Score

The *entailment* score of a given pair is calculated in the following way:

$$score(T, H) = \frac{ed(T, H)}{ed(, H)} \tag{3}$$

where $ed(T, H)$ is the function that calculates the edit distance cost and $ed(, H)$ is the cost of inserting the entire tree H. A similar approach is presented in (Monz and de Rijke 2001), where the entailment score of two document d and d' is calculated by comparing the sum of the weights (idf) of the terms that appear in both documents to the sum of the weights of all terms in d'.

To define the threshold that separates the positive from the negative examples we used the training set provided by the task organizers.

4 Results and Discussion

Table 1 shows the results obtained by the system on the two runs we submitted. The first run used the edit-distance approach on all the subtasks, while the second run used the edit distance for the Comparable Documents (CD) subtask task and and a linear sequence of words for the rest of the tasks. We decided to do the second run because we wanted to evaluate the real impact of using deep syntactic analysis. Results are slightly better for the first run both in the cws (0.60 against 0.58) and recall (0.64 against 0.50).

A relevant problem we encountered, affecting about 30% of the pairs, is that the parser represents in a different way occurrences of similar expressions, making harder to apply edit transformations. For instance, “Wal-Mart” and “Wal-Mart Stores inc.” have different trees, being “Mart” the governing node in the first case and the governed node in the second. The problem could be addressed by changing the order of the nodes in T which is however complex because it introduces changes in the tree edit-distance algorithm. Another solution, which we intend to explore in the future, is the integration of specialized tools and resources for handling named entities and acronyms. In addition, for about 20% of the pairs, the parser did not produce the right analysis either for T or for H.

Another drawback of the tree-edit distance approach is that it is not able to observe the whole tree, but only the subtree of the processed node. For example, the cost of the insertion of a subtree in H

run	measure	CD	IE	MT	QA	RC	PP	IR	Overall
1	accuracy	0.78	0.48	0.50	0.52	0.52	0.52	0.47	0.55
	cws	0.89	0.50	0.55	0.49	0.53	0.48	0.51	0.60
	precision								0.55
	recall								0.64
2	accuracy	0.78	0.53	0.49	0.48	0.54	0.48	0.47	0.55
	cws	0.89	0.53	0.53	0.42	0.58	0.43	0.50	0.58
	precision								0.56
	recall								0.50

Table 1: ITC-irst results at PASCAL-RTE

could be smaller if the same subtree is deleted from T in prior or later stage.

The current implementation of the system does not use resources (e.g. WordNet, paraphrases in (Lin and Pantel 2001), entailment patters as acquired in (Szpektor et al. 2004)) that could significantly wide the application of entailment rules and, consequently, improve performances. We estimated that for about 40% of the the true positive pairs the system could have used entailment rules found in entailment and paraphrasing resources. As an example, the pair 565:

> T - Soprano's Square: Milan, Italy, home of the famed La Scala opera house, honored soprano Maria Callas on Wednesday when it renamed a new square after the diva.
>
> H - La Scala opera house is located in Milan, Italy.

could be successfully solved using a paraphrase pattern such as *Y home of X <=> X is located in Y*, which can be found in (Lin and Pantel 2001). However, in order to use this kind of entailment rules, it would be necessary to extend the "single node" implementation of tree edit distance to address editing operations among subtrees.

Our participation in the RTE challenge served as a first test of our system. In the future, we plan to expand the system by searching for solutions for the mentioned problems and introducing *entailment* and *paraphrasing* resources.

References

Dagan, I., Glickman, O. 2004 Generic applied modeling of language variability *In Proceedings of PASCAL Workshop on Learning Methods for Text Understanding and Mining* Grenoble

Lin, D. 1998. Dependency-based evaluation of MINIPAR. *In Proceedings of the Workshop on Evaluation of Parsing Systems at LREC-98.* Granada, Spain.

Lin, D. and Pantel, P. 2001. Discovery of inference rules for Question Answering. *Natural Language Engineering, 7(4)*, pages 343-360.

Monz, C. and de Rijke, M. 2001. Light-Weight Entailment Checking for Computational Semantics. *The third workshop on inference in computational semantics (ICoS-3).*

Punyakanok., V.,Roth, D. and Yih, W., 2004 Mapping Dependencies Trees: An Application to Question Answering *Proceedings of AI & Math 2004*

Ratnaparkhi, A. 1996 A Maximum Entropy Part-Of-Speech Tagger. *In proceeding of the Empirical Methods in Natural Language Processing Conference, May 17-18, 1996*

Szpektor I., Tanev H., Dagan I., and Coppola B. 2004 Scaling Web-based Acquisition of Entailment Relations *In Proceedings of EMNLP-04 - Empirical Methods in Natural Language Processing, Barcelona, July 2004*

K. Zhang K., Shasha D. 1990 Fast algorithm for the unit cost editing distance between trees. *Journal of algorithms, vol. 11, p. 1245-1262, December 1990.*

Textual Entailment Recognition Based on Dependency Analysis and WordNet

Jesús Herrera, Anselmo Peñas, Felisa Verdejo
Departmento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
Madrid, Spain
{jesus.herrera, anselmo, felisa}@lsi.uned.es

Abstract

The UNED-NLP Group [1] Recognizing Textual Entailment System is based on the use of a broad-coverage parser to extract dependency relations and a module which obtains lexical entailment relations from *WordNet*. The work aims at comparing whether the matching of dependency trees substructures give better evidence of entailment than the matching of plain text alone.

1 Introduction

The system of the UNED-NLP Group which has taken part in the 2005 PASCAL [2] Recognizing Textual Entailment Challenge is a proposal towards the resolution of the Recognizing Textual Entailment (RTE) problem. The present approach explores the possibilities of matching between dependency trees of text and hypothesis. System's components are the following:

1. A dependency parser, based on Lin's *Minipar* (Lin, 1998), which normalizes data from the corpus of text and hypothesis pairs, accomplishes the dependency analysis and creates into memory appropriated structures to represent it.

2. A lexical entailment module, which takes the information given by the parser and returns hypothesis' nodes entailed by the text.

3. A matching evaluation module, which searches for paths into hypothesis' dependency tree, conformed by the lexically entailed nodes.

Section 2 shows how lexical entailment is accomplished. Section 3 presents the methodology followed to evaluate matching between dependency trees. Section 4 describes some experiments and their results. Finally, some conclusions are given.

2 Lexical Entailment

After the dependency parsing, a module of lexical entailment is applied over the nodes of both text and hypothesis. The output of this module is a list of pairs *(T,H)* where *T* is a node in the text's dependency tree whose lexical unit entails the lexical unit of the node *H* in the hypothesis' dependency tree. This entailment at the word level considers *WordNet* relations, detection of *WordNet* multiwords and negation, as follows:

2.1 Synonymy and Similarity

The lexical unit *T* entails the lexical unit *H* if they can be *synonyms* according to *WordNet* or if there is a relation of *similarity* between them. Some examples were found in the PASCAL Challenge training corpus such as, for example: *discover* and *reveal*, *obtain* and *receive*, *lift* and *rise*, *allow* and *grant*, etcetera.

[1] Natural Language Processing and Information Retrieval Group at the Spanish Distance Learning University. http://nlp.uned.es/

[2] Pattern Analysis, Statistical Modelling and Computational Learning Network of Excellence. http://www.pascal-network.org/

2.2 Hyponymy and WordNet Entailment

Hyponymy and *entailment* are relations between *WordNet synsets* having a transitive property. The entailment predicate between two *synsets* was implemented according to these relations as the searching of a path from *synset* S_T to *synset* S_H, in which *hyponymy* and *WordNet entailment* relations between intermediate *synsets* are considered in the direction from S_T to S_H. Then, the lexical unit *T* entails the lexical unit *H* if there is a path from one *synset* of *T* to one *synset* of *H*. Some examples after processing the training corpus of PASCAL Challenge are: *glucose* entails *sugar* (i.e. *glucose* is a hyponym of *sugar*), *crude* entails *oil*, *death* entails *kill*.

2.3 Multiwords

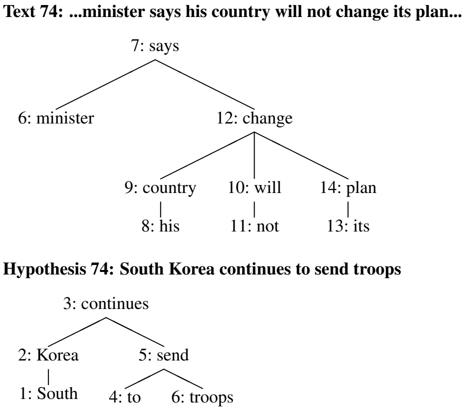
The recognition of multiwords cannot be a previous to lemmatization and parsing step, so a pre and a post processing must be performed in order to avoid errors in the processing. For example, the recognition of the multiword *came_down* requires the previous obtention of the lemma *come*, because the multiword present in *WordNet* is *come_down*.

The variation of multiwords is not due only to lemmatization. Sometimes there are some characters that change as, for example, a dot in an acronym or a proper noun with different wordings. For this reason, a fuzzy matching between candidate and *WordNet* multiwords was implemented using the Levenshtein's edit distance (1965). If the two strings differ in less than 10%, then the matching is permitted. For example, the multiword *Japanise_capital* in hypothesis 345 of the training corpus is translated into the *WordNet* multiword *Japanese_capital*, allowing the entailment between Tokyo and it. Some other examples of entailment after multiword recognition are, regarding synonymy, *blood_glucose* and *blood_sugar*, *Hamas* and *Islamic_Resistance_Movement*, *Armed_Islamic_Group* and *GIA* and, regarding hyponymy, *war_crime* entails *crime*, *melanoma* entails *skin_cancer*.

2.4 Negation and Antonymy

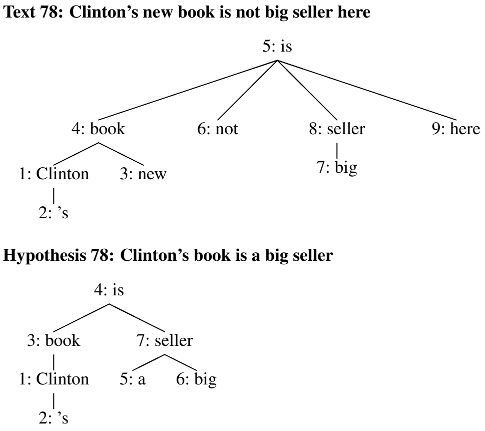
Negation is detected after searching leaves with a negation relation in the dependency tree. This negation relation is then propagated to its ancestors until the head. For example, Figures 1 and 2 show an excerpt of the dependency trees for the training examples 74 and 78 respectivelly. Negation at node 11 of text 74 is propagated to node 10 (*neg(will)*) and node 12 (*neg(change)*). Negation at node 6 of text 78 is propagated to node 5 (*neg(be)*).

Entailment is not possible between a lexical unit and its negation. For example, before considering negation, node 5 in text 78 (*be*) entails node 4 in hypothesis 78 (*be*). Now, this entailment is not possible.


Text 74: ...minister says his country will not change its plan...

Hypothesis 74: South Korea continues to send troops

Figure 1: Dependency trees for pair 74 from training corpus.


Text 78: Clinton's new book is not big seller here

Hypothesis 78: Clinton's book is a big seller

Figure 2: Dependency trees for pair 78 from training corpus.

The entailment between nodes affected by negation is implemented considering the antonymy relation of *WordNet*, and applying the previous processing to them (sections 2.1, 2.2, 2.3). For example, since node 12 in text 74 is negated (*neg(change)*), the antonyms of *change* are considered in the entailment relations between text and hypothesis. Thus, *neg(change)* in text entails *continue* in the hypothesis because the antonym of *change*, *stay*, is a synonym of *continue*.

3 Matching between Dependency Trees

Dependency trees give a structured representation for every text and hypothesis. Matching between dependency trees can give an idea about how semantically similar are two text snippets; this is because a certain semantic information is implicitly contained into dependency trees. The technique used to evaluate matching between dependency trees is inspired in Lin's proposal (Lin, 2001). The initial idea was to use a very simple matching algorithm, focused on searching for all the branches starting at any leaf from hypothesis' tree and showing a matching with any branch from text's tree. Hence, a hypothesis' matching branch is defined as the one whose all nodes show a lexical entailment with nodes from a branch of the correspondent text.

The existence or not of an entailment relation from a text to its correspondent hypothesis was determined by means of their similarity. Similarity between text and hypothesis is defined as the proportion of hypothesis' nodes pertaining to matching branches. From the results obtained against the training corpus, it was empirically determined a threshold for that similarity value. Best accuracy for the system was obtained when 50% was assigned as threshold value. Hence, it was said that a text entailed a hypothesis if hypothesis' dependency tree showed a percentage of matching nodes greater or equal than 50%. If that percentage was less than 50% it was said that no entailment existed from text to hypothesis.

4 Experiments

Along the development time of the proposed system some experiments were accomplished in order to obtain feedback about succesive improvements made to it. For this purpose, several baseline systems – whose results against the training corpus were compared – were developed.

4.1 Baselines

Two different baselines were generated in order to analyse the behaviour of the proposed system against the training corpus. Since lexical entailment is previous to matching between dependency trees, two more simple systems were developed to obtain the mentioned baselines:

- Baseline system I calculated the ratio of words from the hypothesis which appeared into the text.

- Baseline system II computed the ratio of lemmas from the hypothesis which are entailed by any lemma from the text.

In all cases the classification threshold was 50%, as explained in section 3.

4.2 Results over the Training Corpus

The proposed system and the baselines show similar results. Accuracy, calculated for every type of application setting, ranges between 46.67% and 55.56%, except for comparable documents (CD), showing 76.29%, 71.13% and 80.41% accuracy for baseline system I, baseline system II and proposed system, respectively. The overall results are 54.95%, 55.48% and 56.36% accuracy for baseline system I, baseline system II and proposed system, respectively.

4.3 Official Results at the Challenge

Since up to two runs were admitted for submission, it was decided to prepare a third baseline to compare the system against the test corpus. For this baseline system III, queries to *WordNet* were not used but only coincidence between lemmas from text and hypothesis. Hence, one of the submitted runs was generated by this latter baseline system.

The proposed system was refined for its run against the test corpus. This last implementation searched for *subject* or *object* relations along hypothesis' matching branches, requiring also a matching between these relations.

Accuracy, calculated for every type of application setting, ranges between 42.55% and 55.83%, except for CD, showing 79.33% and 78.67% accuracy for baseline system III and proposed system, respectively. The overall results are 55.75% and 54.75%

accuracy for baseline system III and proposed system, respectively.

The behaviour of both systems is similar to the ones executed against the training corpus. However, consideration of *subject* and *object* relations cause a slight decrease of accuracy.

5 Analysis and Conclusions

Results show that a matching-based approach (as shown here) is not enough to tackle appropriately the problem except, perhaps, for CD tasks.

The analysis of cases shows that a high lexical overlap does not mean a semantic entailment and a low lexical overlap does not mean different semantics. Both lexical and syntactic issues to be improved have been detected.

Some kind of paraphrasing between n-grams would be useful; for example, in pair 96[3] of the training corpus is necessary to detect the equivalence between *same-sex* and *gay* or *lesbian*; or, in pair 128[4], *come into conflict with* and *attacks* must be detected as equivalent. Previous work has been developed; for example, Szpektor et al. (2004) propose a web-based method to acquire entailment relations; Barzilay and Lee (2003) use multiple-sentence alignment to learn paraphrases in an unsupervised way; Hermjakob et al. (2002) show how *WordNet* can be extended as a reformulation resource; Pang et al. (2003) represent paraphrases as word lattices; Tomuro (2003) studies the case of question paraphrases.

Other problem is that, in certain cases, a high matching between hypothesis' nodes and text's nodes is given but, simultaneously, hypothesis' branches match with disperse text's branches; then, syntactic relations between subestructures of the text and the hypothesis must be analyzed in order to determine the existence of an entailment. This fact suggests to accomplish an in-depth treatment of syntactic relations.

Hence, it is observed that for RTE is necessary to tackle a wide set of linguistic phenomena in a specific way, at the lexical level and at the syntactic level.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Technology within the following project: TIC-2003-07158-C04-02 Multilingual Answer Retrieval Systems and Evaluation, SyEMBRA.

References

Regina Barzilay and Lillian Lee. 2003. *Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment.* NAACL-HLT-2003.

U. Hermjakob, A. Echihabi, and D. Marcu. 2002. *Natural Language Based Reformulation Resource and Web Exploitation for Question Answering*. Proceedings of TREC-2002.

V. I. Levenshtein. 1965. *Binary Codes Capable of Correcting Deletions, Insertions, and Reversals.* Soviet Physics - Doklady, Vol. 10 No. 8 pp. 707-710, February 1966. Translated from Doklady Akademii Nauk SSSR, Vol. 163 No. 4 pp. 845-848, August 1965.

Dekang Lin. 1998. *Dependency-based Evaluation of MINIPAR.* Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, 1998.

Dekang Lin and Patrick Pantel. 2001. *Discovery of Inference Rules for Question Answering*. Natural Language Engineering 7(4):343-360.

B. Pang, K. Knight, and D. Marcu. 2003. *Sintax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences.* NAACL-HLT-2003.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. *Scaling Web-based Acquisition of Entailment Relations.* Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics, Barcelona, 2004.

N. Tomuro. 2003. *Interrogative Reformulation Patterns and Acquisition of Question Answering Paraphrases..* Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP2003).

[3] *Text 96:* The Massachusetts Supreme Judicial Court has cleared the way for lesbian and gay couples in the state to marry, ruling that government attorneys "failed to identify any constitutionally adequate reason" to deny them the right.
Hypothesis 96: U.S. Supreme Court in favor of same-sex marriage

[4] *Text 128:* Hippos do come into conflict with people quite often.
Hypothesis 128: Hippopotamus attacks human.

Textual Entailment as Syntactic Graph Distance: a rule based and a SVM based approach

Maria Teresa Pazienza, Marco Pennacchiotti
DISP, University of Rome "Tor Vergata",
Viale del Politecnico 1, Roma, Italy,
{pazienza, pennacchiotti}@info.uniroma2.it

Fabio Massimo Zanzotto
DISCo, University of Milano-Bicocca,
Via Bicocca degli Arcimboldi 8, Milano, Italy,
zanzotto@disco.unimib.it

Abstract

In this paper we define a measure for textual entailment recognition based on the *graph matching theory* applied to syntactic graphs. We describe the experiments carried out to estimate measure's parameters with SVM and we report the results obtained on the Textual Entailment Challenge development and testing set.

1 Introduction

Graph distance/similarity measures are widely recognized to be powerful tools for *matching problems* in computer vision and pattern recognition applications (Bunke and Shearer, 1998). Objects to be matched (two images, patterns, etc.) are represented as graphs, turning the recognition problem into a graph matching task. As hypothesis (*H*) and text (*T*) may be seen as two syntactic graphs we can reduce the *textual entailment* (Dagan and Glickman, 2004) recognition problem to a graph similarity measure estimation even if textual entailment has particular properties: *a)* unlike the classical graph problems, is not symmetric; *b)* node similarity can not be reduced to the *label level* (e.g. token similarity); *c)* similarity should be estimated considering also linguistically motivated *graph transformations* (e.g., nominalization and passivization).

In principle, textual entailment is a transitive oriented relation holding in one of the following cases:

1. T *semantically subsumes* H (e.g., in *H:*[The cat eats the mouse] and *T:*[the cat devours the mouse], *eat* generalizes *devour*).
2. T *syntactically subsumes* H (e.g., in *H:*[The cat eats the mouse] and *T:*[the cat eats the mouse in the garden], T contains a specializing prepositional phrase).
3. T *directly implies* H (e.g., *H:*[The cat killed the mouse], *T:*[the cat devours the mouse]).

Taking this into account we define a measure $\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H)$ for the entailment relation based on $\mathcal{XDG}_T$ and $\mathcal{XDG}_H$, i.e., the syntactic representation of the two sentences T and H. We work under two simplifying assumptions: H is supposed to be a sentence describing completely a fact in an assertive or negative way and H should be a simple S-V-O sentence. Our measure has to satisfy the following properties: (a) having a range between 0 and 1, assigning higher values to couples that are more likely in entailment relation, and a specific orientation, $\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H) \neq \mathcal{E}(\mathcal{XDG}_H, \mathcal{XDG}_T)$; (b) the overlap between $\mathcal{XDG}_T$ and $\mathcal{XDG}_H$ has to describe if a subgraph of $\mathcal{XDG}_T$ implies the graph $\mathcal{XDG}_H$. Linguistic transformations (such as nominalization, passivization, and argument movement), as well as negation, must be also considered, as they can play a very important role.

2 Basic Definitions

For the syntactic representation we rely on the extended dependency graph (XDG) (Basili and Zanzotto, 2002). An $\mathcal{XDG} = (C, D)$ is basically a dependency graph whose nodes C are *constituents* and whose edges D are the *grammatical relations* among the constituents. Constituents are lexicalised syntactic trees with explicit *syntactic heads* and *potential semantic governors* (gov). Dependencies in D represent typed and ambiguous relations among a constituent, the $head$, and one of its $modifiers$. Ambiguity is represented using $plausibility$ (between 0 and 1).

Having the formalism it is possible to define how two structurally similar graphs are one subsumption of the other. Given $\mathcal{XDG}_H = (C_H, D_H)$ and $\mathcal{XDG}_T = (C_T, D_T)$, $\mathcal{XDG}_H$ is in a *isomorphic subsumption* relation with $\mathcal{XDG}_T$ ($\mathcal{XDG}_H \preceq \mathcal{XDG}_T$), if two bijective functions f_C and f_D exist respectively related to the constituents C and the dependencies D ($f_C : C_T \rightarrow C_H$ and $f_D : D_T \rightarrow D_H$). They describe the oriented relation of subsumption between nodes and edges of H and T. *Isomorphic subsumption* will capture textual entailment cases 1 and 3, that is, circumstances in which each node and edge of H has a correspondent in T, and vice-versa.

We denote with $\mathcal{XDG}'_T \subseteq \mathcal{XDG}_T$ a subgraph of $\mathcal{XDG}_T$. A *subgraph subsumption isomorphism* between $\mathcal{XDG}_H$ and $\mathcal{XDG}_T$, written as $\mathcal{XDG}_H \sqsubseteq \mathcal{XDG}_T$, holds if it exists $\mathcal{XDG}'_T \subseteq \mathcal{XDG}_T$ so that $\mathcal{XDG}_H \preceq \mathcal{XDG}'_T$. *Subgraph subsumption isomophism* correspond to textual entailment case 2, i.e, when there are nodes/edges of T not mapped in H, but all nodes/edges of H are mapped in T. Indeed, as the text entailment definition suggests, T can contain more information than H.

To tackle the problem of distortions in the syntactic and semantic interpretation, we can imagine an entailment measure based on the maximal subgraph $\mathcal{XDG}'_H$ of $\mathcal{XDG}_H$ (hereafter *maximal common subsumer subgraph*, *mcss*) that is in a *subgraph subsumption isomorphism* relation with $\mathcal{XDG}_T$, i.e. $\mathcal{XDG}'_H \sqsubseteq \mathcal{XDG}_T$. The measure should consider both the distance between $\mathcal{XDG}'_H$ and $\mathcal{XDG}_H$ and the generalisation steps necessary to draw the relation $\mathcal{XDG}'_H \sqsubseteq \mathcal{XDG}_T$.

3 A Rule-based Similarity Measure

To settle the measure the first problem is to extract $\mathcal{XDG}'_T$, i.e., the maximal subgraph of $\mathcal{XDG}_T$ that is in a subgraph isomorphism relation with $\mathcal{XDG}_H$, through the definition of the functions f_C (Sec.3.1) and f_D (Sec.3.2).

3.1 Node subsumption

To find the $mcss$ graph, we need to check that $\mathcal{XDG}'_H \subseteq \mathcal{XDG}_H$ and $\mathcal{XDG}'_T \subseteq \mathcal{XDG}_T$ are in the isomorphic relation $\mathcal{XDG}'_H \preceq \mathcal{XDG}'_T$. This is possible if the selection process of the subsets of the graphs nodes guarantees the possibility of defining the function f_C. This procedure should try to map each constituent of $\mathcal{XDG}_H$ to its most similar constituent in $\mathcal{XDG}_T$. If this is done, the bijective function f_C is derived by construction. The mapping process is based on the notion of *anchors*, defined as $a = (ch, ct, sm)$, holding an hypothesis and a text constituent (ch and ct), and the degree of *semantic similarity* $sm \in [0, 1]$ between the two. The set of anchors A for an entailment pair contains an anchor for each one of the hypothesis constituents having a correspondences in the text T. For example in the entailment pair of Fig. 1, f_C produces the mapping pairs *[The red cat - The carmine cat], [killed - devours], [the mouse - the mouse].*

To determine the best set A, it is necessary to define the semantic similarity sm. If ch is a noun or a prepositional phrase, similarity is evaluated as:

$$sm(ch, ct) = \alpha * sim(gov_{ch}, gov_{ct}) + (1 - \alpha) * simsub(ch, ct)$$

where gov is the constituent governor, α is an empirically evaluated parameter used to weight the importance of the governor, and $simsub$ takes into account similarity among the all the other subcostituents of ch and ct. This latter is defined as:

$$simsub(ch, ct) = \frac{\sum_{sh \in S_{ch}} \max_{st \in S_{ct}} sim(sh, st)}{|S_{ch}|}$$

where S_{ch} and S_{ct} are the set of remaining simple constituents respectively of ch and ct. Finally, sim expresses the similarity among two simple constituents (set to 1 if simple constituents have the same surface or stem); otherwise, a semantic similarity weight β ∈(0,1) is assigned looking at possible WordNet relations (synonymy, entailment and generalization).

When ch is a verb phrase a different analysis occurs. In fact, a verb anchor can assume different *levels* of similarity, according to the semantic value of its modal. For example *must go-could go* should get a lower similarity than *must go-should go*. A verb phrase is thus composed by its governor gov and its

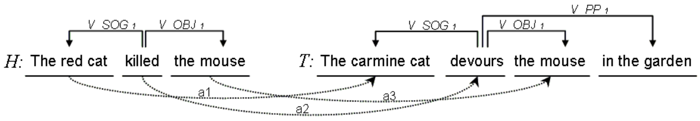


Figure 1: An example of entailment couple in the XDG formalism. Solid lines express grammatical relations D (with *type* and *plausibility)*; dotted lines express anchors a_i between H and T constituents.

modal constituents mod. The overall similarity is thus:

$$sm(ch, ct) = \gamma * sim(gov_{ch}, gov_{ct}) + (1 - \gamma) * dist(mod_{ch}, mod_{ct})$$

where $dist \in [0, 1]$ is empirically derived as the semantic distance between two modals (e.g., $must$ is nearer to $should$ than to $could$) (classified as generic auxiliaries, auxiliaries of possibility and auxiliaries of obligation). Specific cases of syntactic variations, such as active/passive alternation and nominalization are properly treated.

3.2 Edge subsumption

The anchor set A represents the nodes of the $mcss$. We will use f_D to derive the edges of the $mcss$. As XDG edges represent syntactic dependencies among constituents, for each anchor $a \in A$ the syntactic structure of ch and ct is checked, and a related *syntactic similarity* $ss(ch, ct) \in [0, 1]$ is evaluated. In order to obtain ss, it must be firstly defined the set of edges E_{ch} coming out from ch (in Figure 1 example, $E_{killed} = \{V\_sog, V\_obj\}$) and the corresponding set of connected nodes l_{ch} (e.g. $l_{killed} = \{[the\_red\_cat], [the\_mouse]\}$). In the same way, E_{ct} and l_{ct} are defined (e.g. $E_{devour} = \{V\_sog, V\_obj, V\_PP\}$ and $l_{ct} = \{[the\_carmine\_cat], [the\_mouse], [in\_the\_garden]\}$). A^L is defined as the set of anchors that contain overlapping linked constituents, that is, constituents linked with the same syntactic dependency to ch and ct respectively (for example, $a = ([the\_red\_cat], [the\_carmine\_cat], 0.95) \in A^L$, as the two constituents are both linked to *killed* and *devour* via a V_sog edge). ss is defined as:

$$ss(ch, ct) = \frac{\sum\limits_{a \in A^L} sm_a}{|l_{ch}|}$$

Syntactic similarity, defined by f_D, will capture how much similar the syntactic structure accompanied to two constituents (i.e., the edges of the graphs) are, by considering both their syntactic properties (i.e., the common dependencies) and the semantic properties of the constituents to which they are linked (i.e., the similarity sm_a of the anchor of the linked constituents).

3.3 Graph Similarity Measure

Both semantic (sm) and syntactic (ss) similarity (derived respectively from f_C and f_D) must be taken into consideration to evaluate the overall graph similarity measure, as the former captures the notion of node subsumption, and the latter the notion of edge subsumption. For each pair (ch, ct) belonging to the set of anchors A a global similarity is evaluated as:

$$S(ch, ct) = \delta * sm(ch, ct) + (1 - \delta) * ss(ch, ct)$$

where δ is a manually tuned parameter. The overall graph similarity is thus estimated as the average similarity of the anchors $a \in A$ over total number of anchors:

$$\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H) = \frac{\sum\limits_{A} S(ch, ct)}{|A|}$$

It is possible to predict if an entailment relation holds between H and T couple, verifyng $\mathcal{E}(\mathcal{XDG}_T, \mathcal{XDG}_H)$ against a manually tuned threshold t.

4 Applying SVM to Evaluate Parameters

As clear from the previous sections, our measure depends on many parameters (α, β, γ, and δ). These parameters may be evaluated by a machine learning algorithm such as SVM. Due to the basic assumption that H should be a S-V-O sentence, feature spaces can be easily set. In order to comparatively evaluate the importance of different features we defined these feature sets: the features $\mathcal{G}$ related to the graph equivalence measure, i.e. $\mathcal{G}$ = { S_{sim}, S_{simsub}, S_{ss}, V_{sm}, V_{ss}, O_{sim}, O_{simsub}, O_{ss} }; the features $\mathcal{A}$ related to the number of commonly anchored dependencies within constituents to the graph equivalence

		D1	D3	D4	D5	D6
$\mathcal{L}$		51.16(±3.98)	-	-	-	-
$\mathcal{L}$,$\mathcal{T}$,$\mathcal{G}$	$\beta = 0.5$	-	55.28(±2.44)	56.14(±2.51)	56.40(±2.71)	56.72(±2.92)
$\mathcal{L}$,$\mathcal{T}$,$\mathcal{G}$	$\beta = 1$	-	56.37(±2.45)	57.14(±2.94)	57.37(±3.45)	57.12(±3.56)
$\mathcal{L}$,$\mathcal{T}$,$\mathcal{G}$,$\mathcal{A}$	$\beta = 1$	-	-	57.20(±3.01)	57.42(±3.36)	57.12(±3.38)

Table 1: Preliminary analysis on the develpment set using SVM

measure, i.e. $\mathcal{A}$ = { $|l_{ch}|$, $|l_{ct}|$}; $\mathcal{T}$ that are the features related to the textual entailment subtasks (CD, MT, etc.) Feature values are defined in Sec. 3. A final and less complex feature set is $\mathcal{L}$ that represents the percentage of H tokens and of H lemmas in common with T.

5 Results and preliminary evaluation

Before submitting the two runs of the two systems we estimated the parameters over the development set. For the first system referred as *rule-based* we set the parameters at the best value, i.e. $\alpha = 0.85$, $\gamma = 0.85$, and $\delta = 0.5$. Moreover, the *threshold* for predicting a true entailment relation has been set to $t = 0.65$. For the second system referred as *SVM-based* the experiments reported in Tab. 1 have been carried out. The table reports the accuracy of the classifier over the different parameterizations. Rows represent different feature spaces and when necessary the value of the parameter β. Columns represent different degree of the SVM type 1 polynomial kernel. For these preliminary experiments α and γ have been set respectively to 1 and 0.85. This preliminary setting of α, β, and γ seems to be in contrast with the aim of using SVM to estimate the measure parameters, but it is necessary to establish the initial set A of anchors over with values of the features may be computed. These experiments have been made in 3-fold cross validation repeated 10 times. The development set has been randomly divided 10 times (with a pseudo-random function and with 10 fix seeds). The results are reported as mean and standard deviation over 30 runs. All the feature spaces are better than the baseline feature space $\mathcal{L}$. We submitted the system that had the best result in this investigation.

Results over the competition test set are reported in Table 5. As expected by the preliminary analysis over the two development set results are not extremely high. Some trend has been somehow respected. The precision of the *SVM-based* is higher than the precision of the *rule-based* approach. However, it loses many points with respect to the preliminary evaluations, more than the expected standard deviation. The recall of the method is instead in line with the preliminary experiments. On this final set the accuracy of the *rule-based* approach has been higher of the *SVM-based* approach as happened on the development set. Further analysis are needed to better explain these results.

measure	*rule-based*	*SVM-based*
cws	0.5574	0.5591
accuracy	0.5245	0.5182
precision	0.5265	0.5532
recall	0.4975	0.1950
f	0.5116	0.2884

	rule-based		*SVM-based*	
TASK	cws	$accuracy$	cws	$accuracy$
CD	0.8381	0.7651	0.7174	0.6443
IE	0.4559	0.4667	0.4632	0.4917
MT	0.5914	0.5210	0.4961	0.4790
QA	0.4408	0.3953	0.4571	0.4574
RC	0.5167	0.4857	0.5898	0.5214
PP	0.5583	0.5400	0.5768	0.5000
IR	0.4405	0.4444	0.4882	0.4889

Table 2: Competition results

References

Roberto Basili and Fabio Massimo Zanzotto. 2002. Parsing engineering and empirical robustness. *Natural Language Engineering*, 8/2-3.

Horst Bunke and Kim Shearer. 1998. A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3-4):255–259.

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Learning Methods for Text Understanding and Mining*, Grenoble, France.

An Inference Model for Semantic Entailment in Natural Language

Rodrigo de Salvo Braz **Roxana Girju** **Vasin Punyakanok**
Dan Roth **Mark Sammons**
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, 61801, USA
{braz, girju, punyakan, danr, mssammon}@cs.uiuc.edu

Semantic entailment is the problem of determining if the meaning of a given sentence entails that of another. This is a fundamental problem in natural language understanding that provides a broad framework for studying language variability and has a large number of applications. We present a principled approach to this problem that builds on inducing rerepresentations of text snippets into a hierarchical knowledge representation along with a sound inferential mechanism that makes use of it to prove semantic entailment.

1 General Description of Our Approach

Given two text snippets S (source) and T (target) (typically, but not necessarily, S consists of a short paragraph and T, a sentence) we want to determine if $S \models T$, which we read as "*S entails T*" and, informally, understand to mean that *most people would agree that the meaning of S implies that of T*. Somewhat more formally, we say that S *entails* T when some representation of T can be "matched" (modulo some meaning-preserving transformations to be defined below) with some (or part of a) representation of S, at some level of granularity and abstraction.

The approach consists of these components:

KR: A Description Logic based hierarchical knowledge representation, EFDL, into which we rerepresent the surface level text representations, augmented with induced syntactic and semantic parses and word and phrase level abstractions.

KB: A knowledge base consisting of syntactic and semantic rewrite rules, written in EFDL.

Subsumption: An extended subsumption algorithm which determines subsumption between EFDL expressions (representing text snippets or rewrite rules). "Extended" here means that the basic unification operator is extended to support several word level and phrase level abstractions.

First a set of machine learning based resources are used to induce the representation for S and T. The entailment algorithm then proceeds in two phases: (1) it incrementally generates new representations of the original representation of the source text S and (2) it makes use of an (extended) subsumption algorithm to check whether any of the alternative representations of the source entails the representation of the target T. The subsumption algorithm is used in both phases in slightly different ways.

Figure 1 provides an example of the representation of two text snippets along with a sketch of the extended subsumption to decide the entailment.

2 Hierarchical Knowledge Representation

Our semantic entailment approach relies heavily on a hierarchical representation of natural language sentences, defined formally over a domain $\mathcal{D} = \langle \mathcal{V}, \mathcal{A}, \mathcal{E} \rangle$ which consists of a set $\mathcal{V}$ of typed elements, a set $\mathcal{A}$ of attributes of elements, and a set $\mathcal{E}$ of relations among elements. We use a Description-Logic inspired language, *Extended Feature Description Logic (EFDL)*, an extension of (Cumby and Roth, 2003). As described there, expressions in the language have an equivalent representation as *concept graphs*, and we refer to the latter representation here for comprehensibility.

Nodes in the concept graph represent elements – words or (multiple levels of) phrases. *Attributes* of nodes represent properties of elements. Examples of attributes include {LEMMA, WORD, POS, MAINVERB, PHTYPE, PHHEAD, NETYPE, SRLTYPE {ARG0, ... ARGM}, NEG}. The first three are word

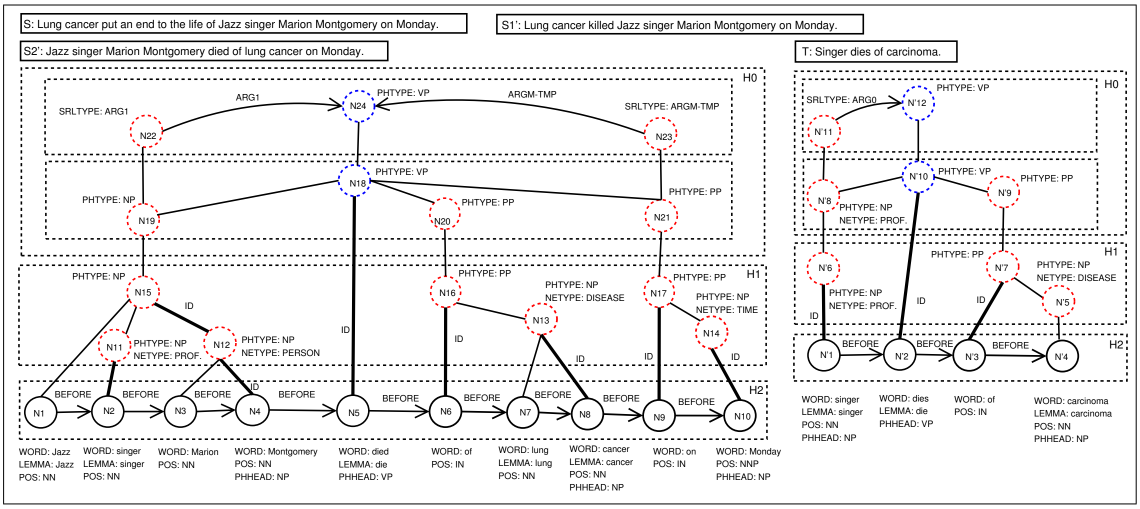


Figure 1: Example of *Re-represented Source & Target* pairs as concept graphs. The original source sentence S generated several alternatives including S_1' and the sentence in the figure (S_2'). Our algorithm was not able to determine entailment of the first alternative (as it fails to match in the extended subsumption phase), but it succeeded for S_2'. The dotted nodes represent phrase level abstractions. S_2' is generated in the first phase by applying the following chain of inference rules: #1 (genitives): “Z’s W → W of Z”; #2: “X put end to Y’s life → Y die of X”. In the extended subsumption, the system makes use of WordNet hypernymy relation (“*lung cancer*” IS-A “*carcinoma*”) and NP-subsumption rule (“*Jazz singer Marion Montgomery*’” IS-A “*singer*”). The rectangles encode the hierarchical levels (H_0, H_1, H_2) at which we applied the extended subsumption. Also note that in the current experiments we don’t consider noun plurals and verb tenses.

level, the next three are phrase level, NETYPE is the named entity of a phrase, SRLTYPE is the set of semantic arguments as defined in PropBank (Kingsbury et al., 2002) and NEG is a negation attribute.

Relations (roles) between two elements are represented by labeled edges between the corresponding nodes. Examples of roles include: $\{$BEFORE, ID, MOD, $\{$ARG0, . . . ARGM$\}\}$; BEFORE indicates the order between two individuals, ID and MOD represent a *contains* relation between a word and a phrase where the word, respectively, is or is not the head.

Concept graphs are used both to describe sentence representations and rewrite rules. Details are omitted here; we just mention that the expressivity of these differ - the body and head of rules are simple chain graphs, for inference complexity reasons. Restricted expressivity is an important concept in Description Logics (Baader et al., 2003), from which we borrow several ideas and nomenclature.

Concept graph representations are induced via state of the art machine learning based resources that include a tokenizer, a lemmatizer, a part-of-speech tagger, a syntactic parser, a semantic parser, a named entity recognizer, and a name coreference system. Rewrite rules were filtered from a large collection of paraphrase rules developed in (Lin and Pantel, 2001) and compiled into our language; a number of non-lexical rewrite rules were generated manually. Currently, our knowledge base consists of approximately 300 inference rules.

The most significant aspect of our knowledge representation is its **hierarchy**. It is defined over a set of typed elements that are partitioned into several classes in a way that captures levels of abstraction and is used by the inference algorithm to exploit these inherent properties of the language. The hierarchical representation provides flexibility – rewrite rules can depend on a level higher than the lexical one, as in: [W/PHTYPE=NP] of [Z/PHTYPE=NP] → Z’s W. Most importantly, it provides a way to abstract over variability in natural language by supporting inference at a higher than word level, and thus supports the inference process in recovering from inaccuracies in lower level representations. Consider the following example in which processing at the semantic parse level exhibits identical structure, despite significant lexical level differences.

S: “[*The bombers*]/A0 *managed* [*to enter* [*the embassy build-*

ing]/A1]/A1."[1]

T: "[*The terrorists*]/A0 *entered* [*the edifice*]/A1."

On the other hand, had the phrase *failed to enter* been used instead of *managed to enter* , a NEG attribute associated with the main verb would prevent this inference. Note that failure of the semantic parser to identify the semantic arguments A0 and A1 will not result in a complete failure of the inference, as described in the next section: it will result in a lower score at this level that the optimization process can compensate for (in the case that lower level inference occurs).

3 Inference Model and Algorithm

An exact subsumption approach that requires the representation of T be entirely embedded in the representation of S'_i is unrealistic. Natural languages allow words to be replaced by synonyms, modifier phrases to be dropped, etc., without affecting meaning. We define below our notion of extended subsumption, computed given two representations, which is designed to exploit the hierarchical representation and capture multiple levels of abstractions and granularity of properties represented at the sentence, phrase, and word-level.

Nodes in a concept graph are grouped into different hierarchical sets denoted by $H = \{H_0, \ldots, H_j\}$ where a lower value of j indicates higher hierarchical level (more important nodes). This hierarchical representation is derived from the underlying concept graph and plays an important role in the definitions below. We say that S'_i entails T if T can be *unified into* S'_i. The significance of definitions below is that we define unification so that it takes into account both the hierarchical representation and multiple abstractions.

Let $V(T)$, $E(T)$, $V(S'_i)$, and $E(S'_i)$ be the sets of nodes and edges in T and S'_i, respectively. Given a hierarchical set H, a *unification* is a 1-to-1 mapping $U = (U_V, U_E)$ where $U_V : V(T) \mapsto V(S'_i)$, and $U_E : E(T) \mapsto E(S'_i)$ satisfying:

1.$\forall(x, y) \in U$: x and y are in the same hierarchical level.

2.$\forall(e, f) \in U_E$: their sinks and sources must be unified accordingly. That is, for n_1, n_2, m_1, and m_2 which are the sinks and the sources of e and f respectively, $(n_1, m_1) \in U_V$ and $(n_2, m_2) \in U_V$.

Let $\mathcal{U}(T, S'_i)$ denote the space of all unifications from T to S'_i. In our inference, we assume the existence of a unification function G determining the cost of unifying pairs of nodes or pairs of edges. G may depend on language and domain knowledge, e.g. synonyms, name matching, and semantic relations. When two nodes or edges cannot be unified, G returns infinity. This leads to the definition of *unifiability*.

Definition 3.1 *Given a hierarchical set H, a unification function G, and two concept graphs S'_i and T, we say that T is* unifiable *to S'_i if there exists a unification U from T to S'_i such that the cost of unification defined by*

$$D(T, S'_i) = \min_{U \in \mathcal{U}(T, S'_i)} \sum_{H_j} \sum_{(x,y) \in U | x,y \in H_j} \lambda_j G(x, y)$$

is finite, where λ_j are some constants s.t. the cost of unifying nodes at higher levels dominates those of the lower levels.

Because top levels of the hierarchy dominate lower ones, nodes in both graphs are checked for subsumption in a top down manner. The levels and corresponding processes are:

`Hierarchy set` H_0 corresponds to sentence-level nodes, represented by the verbs in the text. The inherent set of attributes is {PHTYPE, MAINVERB, LEMMA}. In order to capture the argument structure at sentence-level, each verb in S'_i and T has a set of edge attributes {ARG$_i$, PHTYPE$_i$}, where ARG$_i$ and PHTYPE$_i$ are the semantic role label and phrase type of each argument i of the verb considered.

For each verb in S'_i and T, check if they have the same attribute set and argument structure at two abstraction levels:

1) The semantic role level (SRL attributes). eg: ARG0 verb ARG1 : *[Contractors]*/ARG0 *build [houses]*/ARG1 *for $100,000.*

2) The syntactic parse level (parse tree labels). Some arguments of the verb might not be captured by the semantic role labeler (SRL); we check their match at the syntactic parse level. eg: NP verb NP PP : *[Contractors]*/NP *build [houses]*/NP *[for $100,000]*/ PP.

At this level, if all nodes are matched (modulo functional subsumption), the cost is 0, otherwise it is infinity.

[1] The verbs "*manage*" and "*enter*" share the semantic argument "[*the bombers*]/A0".

`Hierarchy set` H_1 corresponds to phrase-level nodes and represents the semantic and syntactic arguments of the H_0 nodes (verbs). If the phase-level nodes are recursive structures, all their constituent phrases are H_1 nodes. For example, a complex noun phrase consists of various base-NPs. Base-NPs have edges to the words they contain.

The inference procedure recursively matches the corresponding H_1 nodes in T and S'_i until it finds a pair whose constituents do not match. In this situation, a *Phrase-level Subsumption* algorithm is applied. The algorithm is based on subsumption rules that are applied in a strict order (as a decision list) and each rule is assigned a confidence factor.

The algorithm makes sure two H_1 nodes have the same PHTYPE, but allows other attributes such as NETYPE to be optional. Each unmatched attribute results in a uniform cost.

`Hierarchy set` H_2 corresponds to word-level nodes. The attributes used here are: {WORD, LEMMA, POS}. Unmatched attributes result in a uniform cost.

We solve the subsumption problem by formulating an equivalent Integer Linear Programming (ILP) problem of which details is omitted. Despite the fact that this optimization problem is NP hard, commercial packages have very good performance on sparse problems such as this one (Xpress-MP,).

4 Experimental Evaluation

The system needs to establish a confidence threshold to decide, for each example, if the entailment is true of false. The system searches for the optimal threshold per task on the development set. It took about 50 minutes to run the system on the development set and 2 hours on the test. Table 1 shows the system's accuracy on the development set.

	All	**Task**						
		CD	**IE**	**IR**	**MT**	**PP**	**QA**	**RC**
System	64.8	74.0	35.0	62.0	87.5	63.8	84.0	49.0
Test	56.1	77.3	50.0	52.2	53.3	50.0	50.0	51.4

Table 1: System's performance obtained for each experiment on the Pascal corpora and its subtasks.

Below are three examples highlighting some interesting aspects of the entailment system. In the first example, the system fails to produce the correct answer due to its inability to identify the verbal paraphrase in the long sentence. However, it successfully captures some hard entailment pairs such as those in examples 2 and 3.

S1: "*As oil prices soared to new heights after a terrorist attack in Saudi Arabia, the nation's influential oil minister tried to reassure markets yesterday that OPEC would do its best to provide adequate supplies.*"

T1: "*As oil prices soared after a terrorist attack in Saudi Arabia, the nation's oil minister tried to reassure markets that OPEC will try to provide adequate supplies.*"

S2: "*A male gorilla escaped from his cage in the Berlin zoo and sent terrified visitors running for cover, the zoo said yesterday.*"

T2: "*A gorilla escaped from his cage in a zoo in Germany.*"

S3: "*The recent G8 summit, which was first held on July 13-16, 1975, took place on Sea Island on June 8-10.*"

T3: "*The recent G8 summit took place on July 13-16, 1975.*"

In the second example, the system takes advantage of the functional subsumption and identifies the PART-OF semantic relation between "*Berlin*" and "*Germany*". The system correctly classifies the third pair as negative as it fails to match the date after it passes the argument structure test.

Acknowledgement

We thank Dash Optimization for the free academic use of their Xpress-MP software. This work was supported by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program, NSF grant ITR-IIS-0085980, and ONR's TRECC and NCASSR programs.

References

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. 2003. *Description Logic Handbook*. Cambridge.

C. M. Cumby and D. Roth. 2003. Learning with feature description logics. In S. Matwin and C. Sammut, editors, *The 12th Intl. Conference on Inductive Logic Programming (ILP)*. Springer. LNAI 2583.

P. Kingsbury, M. Palmer, and M. Marcus. 2002. Adding semantic annotation to the Penn treebank. In *Proceedings of the Human Language Technology conference (HLT).*, San Diego, CA.

D. Lin and P. Pantel. 2001. DIRT: discovery of inference rules from text. In *KDD '01*, pages 323–328.

Xpress-MP. Dash Optimization. Xpress-MP. http://www.dashoptimization.com/products.html.

Web Based Probabilistic Textual Entailment

Oren Glickman, Ido Dagan and Moshe Koppel
Computer Science Department
Bar Ilan University
Ramat Gan, Israel
{glikmao,dagan,koppel}@cs.biu.ac.il

Abstract

This paper proposes a general probabilistic setting that formalizes the notion of textual entailment. In addition we describe a concrete model for lexical entailment based on web co-occurrence statistics in a bag of words representation.

1 Introduction

This paper describes the Bar-Ilan system participating in the Recognising Textual Entailment Challenge[1]. We first propose a general probabilistic setting that formalizes the notion of textual entailment. We then describe a model, derived from the proposed probabilistic setting, for lexical entailment based on web co-occurrence statistics in a bag of words representation.

Although our proposed lexical system is relatively simple, as it doesn't rely on syntactic or other deeper analysis, it nevertheless achieved an overall accuracy of 59% and an average precision of 0.57. The system did particularly well on Comparable Documents (CD) task achieving an accuracy of 83%. The results suggest that the proposed probabilistic framework is a promising basis for improved implementations that incorporate richer information.

2 Probabilistic Textual Entailment

2.1 Motivation

In many intuitive cases, the textual entailment recognition task may be perceived as being deterministic (Dagan and Glickman, 2004). For example, given the hypothesis h_1 = "*Harry was born in Iowa*" and a candidate text t_1 that includes the sentence "*Harry's birthplace is Iowa*", it is clear that t_1 does (deterministically) entail h_1, and humans are likely to have high agreement regarding this decision. In many other texts, though, entailment inference is uncertain and has a probabilistic nature. For example, a text t_2 that includes the sentence "*Harry is returning to his Iowa hometown to get married.*" does not deterministically entail the above h_1 since Harry might have moved to Iowa as a child. Yet, it is clear that t_2 does add substantial information about the correctness of h_1. In other words, the probability that h_1 is indeed true given the text t_2 ought to be significantly higher than the prior probability of h_1 being true. More specifically, we might say that the probability p of h_1 being true should be estimated based on the percentage of cases in which someone's reported hometown is indeed his/her birthplace. Accordingly, we wouldn't accept t_2 as a definite assessment for the truth of h_1. However, in the absence of other definite information, t_2 may partly satisfy our information need for an assessment of the probable truth of h_1, with p providing a confidence probability for this inference.

In the next section we propose a concrete probabilistic setting that formalizes the above rational.

2.1 A Probabilistic Setting

Let T denote a space of possible texts, and $t \in T$ a specific text.

Meanings are captured in our model by hypotheses and their truth values. Let H denote the set of all possible *hypotheses*. A hypothesis $h \in H$ is a propositional statement which can be assigned a truth value. For now it is assumed that h is represented as a textual statement, but in principle other representations for h may fit our framework as well. (For example, h might be syntactically/semantically annotated and possibly include Prolog-style existentially quantified variables).

[1] http://www.pascal-network.org/Challenges/RTE/

A semantic state of affairs is captured by a *possible world w: H* → {0, 1}, which is defined as a mapping from *H* to {0=False, 1=True}, representing the set of *w*'s concrete truth value assignments for all possible propositions. Accordingly, *W* denotes the set of all possible worlds.

A Generative Model

We assume a probabilistic generative model for texts and possible worlds. In particular, we assume that texts are generated within the context of some state of affairs, represented by a possible world. Thus, whenever the source generates a text *t*, it generates also hidden truth assignments that constitute a possible world *w*. The hidden *w* is perceived as a "snapshot" of the (complete) state of affairs in the world within which *t* was generated.

The probability distribution of the source, over all possible texts and truth assignments $T \times W$, is assumed to reflect only inferences that are based on the generated texts. That is, we assume that the distribution of truth assignments is not bound to reflect the state of affairs in any "real" world, but only the inferences about propositions' truth that are related to the text. In particular, the probability for generating a true hypothesis *h* that is not related at all to the corresponding text is determined by some prior probability P(*h*), which is not bound to reflect *h*'s prior in the "real" world. For example, *h*="Paris is the capital of France" might have a prior smaller than 1 and might well be false when the generated text is not related at all to Paris. In fact, we may as well assume that P(*h*) = 1 only for logical tautologies. On the other hand, we assume that the probability of *h* being true (generated within *w*) would be higher than the prior when the corresponding *t* does contribute information that supports *h*'s truth.

We define two types of events over the probability space for $T \times W$:

I) For a hypothesis *h*, we denote as Tr_h the random variable whose value is the truth value assigned to *h* in the world of the generated text. Correspondingly, $\text{Tr}_h=1$ is the event of *h* being assigned a truth value of 1 (True).

II) For a text *t*, we use *t* to denote also the event that the generated text is *t* (as usual, it is clear from the context whether *t* denotes the text or the corresponding event).

Textual entailment relationship

We say that *t* probabilistically entails *h* (denoted as $t \Rightarrow h$) if *t* increases the likelihood of *h* being true, that is, if $\text{P}(\text{Tr}_h = 1| t) > \text{P}(\text{Tr}_h = 1)$ -- or equivalently if the pointwise mutual information, $\text{I}(\text{Tr}_h=1,t)$, is greater than 1.

Entailment confidence

Once *knowing* that $t \Rightarrow h$, we are further interested in a probabilistic confidence value for *h* being true given *t*, which corresponds to $\text{P}(\text{Tr}_h = 1| t)$.

3 Lexical Entailment Models

The proposed setting above provides the necessary grounding for probabilistic modeling of textual entailment. As modeling the full extent of the textual entailment problem is a long term research goal, we focus here on identifying when the lexical elements of a textual hypothesis *h* are inferred from a given text *t,* even if the relations between these concepts may not be entailed from *t*.

To model lexical entailment we first assume that the meanings of the individual (content) words in a hypothesis $h=\{u_1, \ldots, u_m\}$ can be assigned truth values. A possible interpretation for these truth values, common in formal semantics tradition, is that lexical concepts are assigned existential meanings. For example, for a given text *t*, $\text{Tr}_{\text{acquired}}=1$ if it can be inferred in *t*'s state of affairs that an acquisition event exists (occurred). It is important to note though that this is one possible interpretation. We only assume that truth values are defined for lexical items, but do not explicitly annotate or evaluate this sub-task.

Given this setting, a hypothesis is assumed to be true if and only if all its lexical components are true. When estimating the entailment probability we assume that the truth probability of a term in a hypothesis *h* is independent of the truth of the other terms in *h*, obtaining:

$$\text{P}(Tr_h = 1 | t) = \prod_{i=1}^{m} \text{P}(Tr_{u_i} = 1 | t) \qquad (1)$$

In order to estimate $\text{P}(\text{Tr}_u=1|v_1, \ldots, v_n)$ for a given word *u* and text $t=\{v_1, \ldots, v_n\}$, we further assume that the majority of the probability mass comes from a specific entailing word in *t*:

$$\text{P}(Tr_u = 1 | t) = \max_{v \in t} \text{P}(Tr_u = 1 | T_v) \qquad (2)$$

where T_v denotes the event that a generated text contains the word *v*. This corresponds to expecting that each word in *h* will be entailed from a specific word in *t* (rather than from the accumulative context of t as a whole). Alternatively, one can view (2) as inducing an alignment between the terms in *h* to the terms in the *t,* somewhat similar to alignment models in statistical MT (Brown et al., 1993).

Thus we propose estimating the entailment probability based on lexical entailment probabilities from (1) and (2) as follows:

$$\mathrm{P}(Tr_h = 1 | t) = \prod\nolimits_{u \in h} \max\nolimits_{v \in t} \mathrm{P}(Tr_u = 1 | T_v) \qquad (3)$$

3.1 Web-based Estimation of Lexical Entailment Probabilities

We perform unsupervised empirical estimation of the lexical entailment probabilities, $\mathrm{P}(\mathrm{Tr}_u{=}1|T_v)$, based on word co-occurrence frequencies from the web. Following our proposed probabilistic model (cf. Section 2.1), we assume that the web is a sample generated by a language source. Each document represents a generated text and a (hidden) possible world. Given that the possible world of the text is not observed we do not know the truth assignments of hypotheses for the observed texts. We therefore further make the simplest assumption that all hypotheses stated verbatim in a document are true and all others are false and hence $\mathrm{P}(\mathrm{Tr}_u{=}1|T_v) = \mathrm{P}(T_u\,|T_v)$. This simple co-occurrence probability, which we denote as lexical entailment probability – *lep*(*u*,*v*), is easily estimated based on maximum likelihood counts:

$$lep\ (u, v) \approx \mathrm{P}\ (T_u \mid T_v) \approx \frac{n_{u,v}}{n_v} \qquad (4)$$

where n_v is the number of documents containing word *v* and $n_{u,v}$ is the number of documents containing both *u* and *v*. The corresponding counts were achieved by performing queries to a web search engine.

The lexical entailment probability is derived from (4) and (5) above as follows:

$$P(Tr_h = 1 | t) = \prod\nolimits_{u \in h} \max\nolimits_{v \in t} lep(u, v) \qquad (5)$$

4 Experimental Setting

The text and hypotheses of all pairs in development set and test set where tokenized by the following simple heuristic – split at white space and remove any preceding or trailing of these characters: ([{)]}"'`.,;:-!?. A stop list was applied to remove frequent tokens. Counts were obtained using the *AltaVista* search engine[2], which supplies an estimate for the number of results (web-pages) for a given one or two token query.

We empirically tuned a threshold, λ, on the the estimated entailment probability to decide if entailment holds on not. For a pair <*t*,*h*>, we tag an example as true (i.e. entailment holds) if $p = \mathrm{P}(\mathrm{Tr}_h = 1|\ t) > \lambda$, and as false otherwise. We assigned a confidence of p to the positive examples ($p>\lambda$) and a confidence of 1-*p* to the negative ones.

The threshold was tuned on the on the 567 annotated text-hypothesis example pairs in the development set. The optimal (best cws) threshold was $\lambda = 0.005$ with a resulting cws of 0.57 and accuracy of 56%. This threshold was used to tag and assign confidence scores to the 800 pairs of the test set.

4.1 Results

The resulting accuracy on the test set was of 59% and the resulting confidence weighted score was of 0.57. Both are statistically significantly better then chance at the 0.01 level.

4.2 Analysis

Table 1 lists the accuracy and cws when computed separately for each task. As can be seen by the table the system does well on the CD and MT tasks, and quite poorly (not better than chance) on the RC, PP, IR and QA tasks.

task	accuracy	cws
Comparable Documents (CD)	0.8333	0.8727
Machine Translation (MT)	0.5667	0.6052
Information Extraction (IE)	0.5583	0.5143
Reading Comprehension (RC)	0.5286	0.5142
Paraphrase (PP)	0.5200	0.4885
Information Retrieval (IR)	0.5000	0.4492
Question Answering (QA)	0.4923	0.3736

Table 1: accuracy and cws by task

It seems as if the success of the system is attributed almost solely to its success on the CD and MT tasks. Indeed it seems as if there is something common to these two tasks, which differentiates them from the others - in both tasks high overlap of content words (or their meanings) tend to correspond to entailment.

Success and failure cases

The system misclassified 331 out of the 800 test examples. The vast majority of these mistakes

[2] http://www.av.com

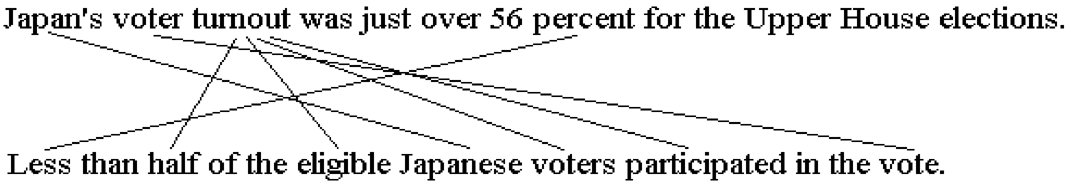

Japan's voter turnout was just over 56 percent for the Upper House elections.
Less than half of the eligible Japanese voters participated in the vote.

Figure 2: system's underlying alignment for example 1026 (RC). gold standard - false, system - false

(75%) were false positives – pairs the system tagged as true but annotated as false. It is also interesting to note that the false negative errors were more common among the MT and QA tasks while the false positive errors were more typical to the other tasks. An additional observation from the recall-precision curve (Figure 1), is that high system confidence actually corresponds to false entailment. This is attributed to an artifact of this dataset by which examples with high word overlap between the text and hypothesis tend to be biased to negative examples.

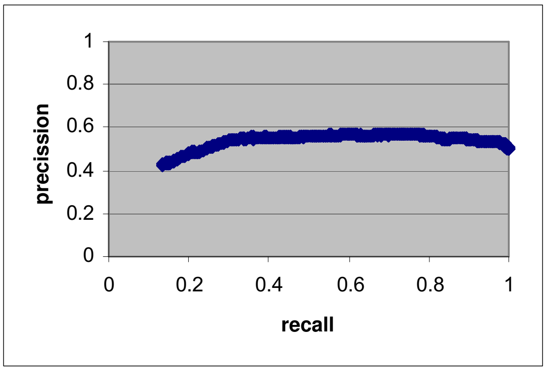


Figure 1: precision recall curve of system

In an attempt to 'look under the hood' we exmined at the underlying alignment preformed by our system on a sample of examples. Figure 2 illustrates a typical alignment. Though some of the entailing words correspond to what we believe to be the correct alignment (e.g. voter → vote, japan's → japanese), the system also finds many dubious lexical pairs (e.g. turnout → half, percent → less). Obviously, co-occurrence within documents is only one factor in estimating the entailment between words. This information should be combined with other statistical criteria that capture complementary notions of entailment, as addressed in (Geffet and Dagan, 2004), or with lexical resources such as WordNet.

In an additional experiment we tried using as a confidence score a weighted average of the lexical probabilities (rather than the product in Equation 1) using the token's *idf* as a weight, following the weighting scheme which was applied to direct word overlap in (Monz and de Rijke, 2001). This method resulted in comparable but slightly lower accuracy of 56%.

5 Conclusions

This paper described the Bar-Ilan system participating in the Recognising Textual Entailment Challenge. We proposed a general probabilistic setting that formalizes the notion of textual entailment. In addition we described a model for lexical entailment based on web co-occurrence statistics in a bag of words representation. Although our proposed lexical system is relatively simple, as it doesn't rely on syntactic or other deeper analysis; it nevertheless achieved encouraging results. The results suggest that the proposed probabilistic framework is a promising basis for improved implementations incorporating deeper types of information.

References

Ido Dagan and Oren Glickman. 2004. *Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability*. PASCAL workshop on Learning Methods for Text Understanding and Mining.

Maayan Geffet and Ido Dagan. 2004. *Feature Vector Quality and Distributional Similarity*, Coling 2004.

Christof Monz, Maarten de Rijke. 2001. Light-Weight Entailment Checking for Computational Semantics. In Proc. of the third workshop on inference in computational semantics (ICoS-3).

Textual Entailment Recognition Based on Inversion Transduction Grammars

Dekai Wu[1]

Human Language Technology Center
HKUST
Department of Computer Science
University of Science and Technology, Clear Water Bay, Hong Kong
dekai@cs.ust.hk

Abstract

The PASCAL Challenge's textual entailment recognition (RTE) task presents intriguing opportunities to test various implications of the strong language universal constraint posited by Wu's (1995, 1997) Inversion Transduction Grammar (ITG) hypothesis. The ITG Hypothesis provides a strong inductive bias, and has been repeatedly shown empirically to yield both efficiency and accuracy gains for numerous language acquisition tasks. Since the RTE challenge abstracts over many tasks, it invites meaningful analysis of the ITG Hypothesis across tasks including information retrieval, comparable documents, reading comprehension, question answering, information extraction, machine translation, and paraphrase acquisition. We investigate two new models for the RTE problem that employ simple generic Bracketing ITGs. Experimental results show that, even in the absence of any thesaurus to accommodate lexical variation between the Text and the Hypothesis strings, surprisingly strong results for a number of the task subsets are obtainable from the Bracketing ITG's structure matching bias alone.

1 Introduction

The *Inversion Transduction Grammar* or *ITG* formalism, which historically was developed in the context of translation and alignment, hypothesizes strong expressiveness restrictions that constrain paraphrases to vary word order only in certain allowable nested permutations of arguments—even across different languages (Wu, 1997). The textual entailment recognition (RTE) challenge provides opportunities for meaningful analysis of the ITG Hypothesis across a broad range of application domains.

The strong inductive bias imposed by the ITG Hypothesis has been repeatedly shown empirically to yield both efficiency and accuracy gains for numerous language acquisition tasks, across a variety of language pairs and tasks. Zens and Ney (2003) show that ITG constraints yield significantly better alignment coverage than the constraints used in IBM statistical machine translation models on both German-English (Verbmobil corpus) and French-English (Canadian Hansards corpus). Zhang and Gildea (2004) find that unsupervised alignment using Bracketing ITGs produces significantly lower Chinese-English alignment error rates than a syntactically supervised tree-to-string model (Yamada and Knight, 2001). With regard to translation rather than alignment accuracy, Zens *et al.* (2004) show that decoding under ITG constraints yields significantly lower word error rates and BLEU scores than the IBM constraints.

The present studies on the RTE challenge are motivated by the following observation: the empirically demonstrated suitability of ITG paraphrasing constraints across languages should hold, if anything, even more strongly in the monolingual case.

The simplest class of ITGs, *Bracketing ITGs*, are particularly interesting in applications like the RTE challenge, because they impose ITG constraints in language-independent fashion, and in the simplest case do not require any language-specific linguistic grammar or training. In Bracketing ITGs, the grammar uses only a single, undifferentiated non-terminal (Wu, 1995). The key modeling property of Bracketing ITGs that is most relevant to the RTE challenge is that they assign strong preference to candidate Text-Hypothesis pairs in which nested constituent subtrees can be recursively aligned with a minimum of constituent boundary violations. Unlike language-specific linguistic approaches, however, the shape of the trees are driven in unsupervised fashion by the data. One way to view this is that the trees are hidden explanatory variables. This not only provides significantly higher robustness than more highly constrained manually constructed grammars, but also makes the model widely applicable across languages in economical fashion without a large investment in manually con-

[1] The author would like to thank the Hong Kong Research Grants Council (RGC) for supporting this research in part through grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09, and Marine Carpuat for invaluable assistance in preparing the datasets and stoplist.

structed resources.

Formally, ITGs can be defined as the restricted subset of syntax-directed transduction grammars or SDTGs Lewis and Stearns (1968) where all of the rules are either of *straight* or *inverted* orientation. Ordinary SDTGs allow any permutation of the symbols on the right-hand side to be specified when translating from the input language to the output language. In contrast, ITGs only allow two out of the possible permutations. If a rule is straight, the order of its right-hand symbols must be the same for both language. On the other hand, if a rule is inverted, then the order is left-to-right for the input language and right-to-left for the output language. Since inversion is permitted at any level of rule expansion, a derivation may intermix productions of either orientation within the parse tree. The ability to compose multiple levels of straight and inverted constituents gives ITGs much greater expressiveness than might seem at first blush.

Moreover, for reasons discussed by Wu (1997), ITGs possess an interesting intrinsic combinatorial property of permitting roughly up to four arguments of any frame to be transposed freely, but not more. This matches suprisingly closely the preponderance of linguistic verb frame theories from diverse linguistic traditions that all allow up to four arguments per frame. Again, this property emerges naturally from ITGs in language-independent fashion, without any hardcoded language-specific knowledge. This further suggests that ITGs should do well at picking out Text-Hypothesis pairs where the order of up to four arguments per frame may vary freely between the two strings. Conversely, ITGs should do well at rejecting pairs where (1) too many words in one sentence find no correspondence in the other, (2) frames do not nest in similar ways in the candidate sentence pair, or (3) too many arguments must be transposed to achieve an alignment—all of which would suggest that the sentences probably express different ideas.

As an illustrative example, in common similarity models, the following pair of sentences (found in actual data arising in our experiments below) would receive an inappropriately high score, because of the high lexical similarity between the two sentences:

> Chinese president Jiang Zemin arrived in Japan today for a landmark state visit .
>
> 江泽民 将 是 到 日本 做 国事访问 的 首位 中国 国家 主席 .
> *(Jiang Zemin will be the first Chinese national president to pay a state vist to Japan.)*

However, the ITG based model is sensitive enough to the differences in the constituent structure (reflecting underlying differences in the predicate argument structure) so that our experiments show that it assigns a low score. On the other hand, the experiments also show that it successfully assigns a high score to other candidate bisentences representing a true Chinese translation of the same English sentence, as well as a true English translation of the same Chinese sentence.

We investigate two new models for the RTE problem that employ simple generic Bracketing ITGs, both with and without a stoplist. The experimental results show that, even in the absence of any thesaurus to accommodate lexical variation between the Text and the Hypothesis strings, surprisingly strong results for a number of the task subsets are obtainable from the Bracketing ITG's structure matching bias alone.

2 Experimental Method

Each Text-Hypothesis pair of the test set was scored via the biparsing algorithm described in Wu and Fung (2005) which is essentially similar to the dynamic programming approach of Wu (1997). As mentioned earlier, biparsing for ITGs can be accomplished efficiently in polynomial time, rather than the exponential time required for classical SDTGs.

The ITG scoring model can also be seen as a variant of the approach described by Leusch *et al.* (2003), which allows us to forego training to estimate true probabilities; instead, rules are simply given unit weights (with caveats discussed in the Results section). The ITG scores can be interpreted as a generalization of classical Levenshtein string edit distance, where inverted block transpositions are also allowed. Even without probability estimation, Leusch *et al.* found excellent correlation with human judgment of similarity between translated paraphrases.

We evaluated two different versions of the Bracketing ITG based RTE models.

In the basic version, all words of the vocabulary are included among the lexical transductions, allowing exact word matches between the Text and the Hypothesis.

The second version excludes a list of 172 words from a stoplist from the lexical transductions. The motivation for this model was to discount the effect of words such as "the" or "of" since, more often than not, they could be irrelevant to the RTE task.

No significant training was performed with the available development sets. Rather, the aim was to establish foundational baseline results, to see in this first round of RTE experiments what results could be obtained with the simplest versions of the ITG models.

The RTE test set consists of 300 Text-Hypothesis string pairs, selected from various sources by human collectors. Each string pair is labeled according to the task category that the data was drawn from. These labels divide the data into seven task subsets, which we analyze individually below. While the collectors were attempting to build a representative dataset, it is difficult to make claims about distributional neutrality, due to the arbitrary nature of the example selection process.

3 Results

Across all subsets overall, the basic model produced a confidence-weighted score of 54.97% (better than chance at the 0.05 level). All examples were labeled, so precision, recall, and f-score are equivalent; the accuracy was 51.25%.

Surprisingly, the stoplisted model produced worse results. The overall confidence-weighted score was 53.61%, and the accuracy was 50.50%. We discuss the reasons below in the context of specific subsets.

As one might expect, the Bracketing ITG models performed better on the subsets more closely approximating the tasks for which Bracketing ITGs were designed: comparable documents (CD), paraphrasing (PP), and information extraction (IE). We will discuss some important caveats on the machine translation (MT) and reading comprehension (RC) subsets. The subsets least close to the Bracketing ITG models are information retrieval (IR) and question answering (QA).

3.1 Comparable Documents (CD)

The CD task definition can essentially be characterized as recognition of noisy word-aligned sentence pairs. Among all subsets, CD is perhaps closest to the noisy word alignment task for which Bracketing ITGs were originally developed, and indeed produced the best results for both of the Bracketing ITG models. The basic model produced a confidence-weighted score of 79.88% (accuracy 71.33%), while the stoplisted model produced an essentially unchanged confidence-weighted score of 79.83% (accuracy 70.00%).

The results on the RTE Challenge datasets closely reflect the larger-scale findings of Wu and Fung (2005), who demonstrate that an ITG based model yields far more accurate extraction of parallel sentences from quasi-comparable non-parallel corpora than previous state-of-the-art methods. Wu and Fung's results also use the evaluation metric of uninterpolated average precision (i.e., confidence-weighted score).

Note also that we believe the results here are artificially lowered by the absence of any thesaurus, and that significantly further improvements would be seen with the addition of a suitable thesaurus, for reasons discussed below under the MT subsection.

3.2 Paraphrase Acquisition (PP)

The PP task is also close to the task for which Bracketing ITGs were originally developed. For the PP task, the basic model produced a confidence-weighted score of 57.26% (accuracy 56.00%), while the stoplisted model produced a lower confidence-weighted score of 51.65% (accuracy 52.00%). Unlike the CD task, the greater importance of function words in determining equivalent meaning between paraphrases appears to cause the degradation in the stoplisted model.

The effect of the absence of a thesaurus is much stronger for the PP task as opposed to the CD task. Inspection of the datasets reveals much more lexical variation between paraphrases, and shows that cases where lexis does not vary are generally handled accurately by the Bracketing ITG models. The MT subsection below discusses why a thesaurus should produce significant improvement.

3.3 Information Extraction (IE)

The IE task presents a slight issue of misfit for the Bracketing ITG models, but yielded good results anyhow. The basic Bracketing ITG model attempts to align all words/collocations between the two strings. However, for the IE task in general, only a substring of the Text should be aligned to the Hypothesis, and the rest should be disregarded as "noise". We approximated this by allowing words to be discarded from the Text at little cost, by using parameters that impose only a small penalty on null-aligned words from the Text. (As a reasonable first approximation, this characterization of the IE task ignores the possibility of modals, negation, quotation, and the like in the Text.)

Despite the slight modeling misfit, the Bracketing ITG models produced good results for the IE subset. The basic model produced a confidence-weighted score of 59.92% (accuracy 55.00%), while the stoplisted model produced a lower confidence-weighted score of 53.63% (accuracy 51.67%). Again, the lower score of the stoplisted model appears to arise from the greater importance of function words in ensuring correct information extraction, as compared with the CD task.

3.4 Machine Translation (MT)

One exception to expectations is the machine translation subset, a task for which Bracketing ITGs were developed. The basic model produced a confidence-weighted score of 34.30% (accuracy 40.00%), while the stoplisted model produced a comparable confidence-weighted score of 35.96% (accuracy 39.17%).

However, the performance here on the machine translation subset cannot be directly interpreted, for two reasons.

First, the task as defined in the RTE Challenge datasets is not actually crosslingual machine translation, but rather evaluation of monolingual comparability between an automatic translation and a gold standard human translation. This is in fact closer to the problem of defining a good MT evaluation metric, rather than MT itself. Leusch *et al.* (2003 and personal communication) found that

Bracketing ITGs as an MT evaluation metric show excellent correlation with human judgments.

Second, no translation lexicon or equivalent was used in our model. Normally in translation models, including ITG models, the translation lexicon accommodates lexical ambiguity, by providing multiple possible lexical choices for each word or collocation being translated. Here, there is no second language, so some substitute mechanism to accommodate lexical ambiguity would be needed.

The most obvious substitute for a translation lexicon would be a monolingual thesaurus. This would allow matching synonomous words or collocations between the Text and the Hypothesis. Our original thought was to incorporate such a thesaurus in collaboration with teams focusing on creating suitable thesauri, but time limitations prevented completion of these experiments. Based on our own prior experiments and also on Leusch *et al.*'s experiences, we believe this would bring performance on the MT subset to excellent levels as well.

3.5 Reading Comprehension (RC)

The reading comprehension task is similar to the information extraction task. As such, the Bracketing ITG model could be expected to perform well for the RC subset. However, the basic model produced a confidence-weighted score of just 49.37% (accuracy 47.14%), and the stoplisted model produced a comparable confidence-weighted score of 47.11% (accuracy 45.00%).

The primary reason for the performance gap between the RC and IE domains appears to be that RC is less news-oriented, so there is less emphasis on exact lexical choices such as named entities. This puts more weight on the importance of a good thesaurus to recognize lexical variation. For this reason, we believe the addition of a thesaurus would bring performance improvements similar to the case of MT.

3.6 Information Retrieval (IR)

The IR task diverges significantly from the tasks for which Bracketing ITGs were developed. The basic model produced a confidence-weighted score of 43.14% (accuracy 46.67%), while the stoplisted model produced a comparable confidence-weighted score of 44.81% (accuracy 47.78%).

Bracketing ITGs seek structurally parallelizable substrings, where there is reason to expect some degree of generalization between the frames (heads and arguments) of the two substrings from a lexical semantics standpoint. In contrast, the IR task relies on unordered keywords, so the effect of argument-head binding cannot be expected to be strong.

3.7 Question Answering (QA)

The QA task is extremely free in the sense that questions can differ significantly from the answers in both syntactic structure and lexis, and can also require a significant degree of indirect complex inference using real-world knowledge. The basic model produced a confidence-weighted score of 33.20% (accuracy 40.77%), while the stoplisted model produced a significantly better confidence-weighted score of 38.26% (accuracy 44.62%).

Aside from adding a thesaurus, to properly model the QA task, at the very least the Bracketing ITG models would need to be augmented with somewhat more linguistic rules that include a proper model for *wh-* words in the Hypothesis, which otherwise cannot be aligned to the Text. In the Bracketing ITG models, the stoplist appears to help by normalizing out the effect of the *wh-* words.

4 Conclusion

The most serious omission in our experiments with Bracketing ITG models was the absence of any thesaurus model, allowing zero lexical variation between the Text and Hypothesis. This forced the models to rely entirely on the Bracketing ITG's inherent tendency to optimize structural match between hypothesized nested argument-head substructures. What we find highly interesting is the perhaps surprisingly large effect obtainable from this structure matching bias alone, which already produces good results on a number of the subsets.

We plan to remedy the absence of a thesaurus as the obvious next step. This can be expected to raise performance significantly on all subsets.

References

Gregor Leusch, Nicola Ueffing, and Hermann Ney. A novel string-to-string distance measure with applications to machine translation evaluation. In *Machine Translation Summit*, New Orleans, 2003.

P. M. Lewis and R. E. Stearns. Syntax-directed transduction. *Journal of the Association for Computing Machinery*, 15:465–488, 1968.

Dekai Wu and Pascale Fung. Inversion Transduction Grammar constraints for mining parallel sentences from quasi-comparable corpora. In *Forthcoming*, 2005.

Dekai Wu. An algorithm for simultaneously bracketing parallel texts by aligning words. In *33rd Annual Meeting of the Association for Computational Linguistics Conference (ACL-95)*, Cambridge, MA, Jun 1995. Association for Computational Linguistics.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3), Sep 1997.

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *39th Annual Meeting of the Association for Computational Linguistics Conference (ACL-01)*, Toulouse, France, 2001. Association for Computational Linguistics.

Richard Zens and Hermann Ney. A comparative study on reordering constraints in statistical machine translation. pages 192–202, Hong Kong, August 2003.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of COLING*, Geneva, August 2004.

Hao Zhang and Daniel Gildea. Syntax-based alignment: Supervised or unsupervised? In *Proceedings of COLING*, Geneva, August 2004.

MITRE's Submissions to the EU Pascal RTE Challenge

Samuel Bayer, John Burger, Lisa Ferro,
John Henderson, Alexander Yeh

The MITRE Corporation
202 Burlington Rd.
Bedford, MA 01730, USA
{sam,john,lferro,jhndrsn,asy} @ mitre.org

Abstract

We describe MITRE's two submissions to the RTE Challenge, intended to exemplify two different ends of the spectrum of possibilities. The first submission is a traditional system based on linguistic analysis and inference, while the second is inspired by alignment approaches from machine translation. We also describe our efforts to build our own entailment corpus. Finally, we discuss our investigations and reflections on the strengths and weaknesses of the evaluation itself.

1 Background

The MITRE Corporation has a long-standing interest in both cutting-edge and practical approaches to text understanding. We believe that progress in task-independent text understanding requires an evaluation that pushes the research forward appropriately, and a substantial portion of our effort has been devoted to an in-depth exploration of using standard reading comprehension tests for this purpose (Hirschman et al., 1999). We have discovered, however, that the availability of such corpora is limited, their construction is expensive, and reading comprehension tests in general tend to be limited in their diagnostic ability.

In this context, the RTE (Recognizing Textual Entailment) Challenge appeals to us due to its generality, its simple structure, and the possibility that it might be significantly less expensive to develop the appropriate test corpora and sufficient training corpora, for those systems that require such. We also suspect that RTE techniques will be applicable to a broad range of problems.

For the challenge, MITRE developed two systems. We hypothesize that a successful RTE system will include elements of traditional approaches based on explicit linguistic analysis and inference, alongside robust, statistical approaches that leverage a range of simple, reliably extractable features. To clarify the shortcomings of each approach alone, and to help focus on how they might support each other, we implemented a system at each end of the continuum. System 1 is our traditional system, and System 2 is our statistical system.

2 System 1

System 1 is a baseline traditional system constructed using explicit modeling of linguistic analysis. The system processes both the Hypothesis and the Text using a MITRE-built tokenizer and sentence segmenter, the Ratnaparkhi (1996) POS tagger, the University of Sussex's Morph morphological analyzer (Minnon et al., 2001), the CMU Link Grammar parser (Sleator & Temperley, 1993), and a MITRE-built dependency analyzer and Davidsonian logic generator. The Text and Hypothesis are then compared using the University of Rochester's EPILOG event-oriented probabilistic inference engine (Schubert & Hwang, 2000). Very little additional semantic knowledge is exploited, beyond a few added inference rules and simple word lists for semantic classification. Due to its currently impoverished knowledge base, the system fails to prove entailment for virtually all of the RTE data, and thus labels almost all of the data as non-entailing.

The results of System 1 on the test set are shown in Figure 1. Due to parse failures and other problems, the system failed to convert 213 of the 800 test pairs into the event logic, and so we made a partial submission for the other 587 test pairs. During development, pairs marked true were slightly more accurate than pairs marked false. This led us to a simplistic confidence scheme of 1.0 for true results and 0.5 for false results

System 1 currently has just two rules. One is intended to handle certain modals, e.g., *can run* does not entail *run*. This rule has no effect on the test set. The other rule handles some appositive cases. This other rule accounts for 2 of the correctly labeled Trues and 1 of the incorrectly labeled Trues.

Partly because System 1 has very few inference rules, about half of the correctly marked true pairs were pairs where the hypothesis is a simple subset of the text (e.g., *Rover is a big dog* entails *Rover is a dog*). However, this subset property of the inference engine also caused 6 of the 10 pairs incorrectly marked true; *Rover is not a dog* should not entail *Rover is a dog*, but System 1 thinks it does, due in part to our flat semantic representation (our modal rule was an attempt to address a small subset of these cases).

As we continue to work on this problem, we plan to exploit multiple potential sources of additional information: both explicit information sources like WordNet (Fellbaum, 1998) and information extracted from large background corpora such as Gigaword (Graff, 2003). We're also planning to synthesize this approach with the radically different approach found in System 2.

3 System 2

Statistical machine translation models inspire MITRE's second RTE system. These models are designed to find correspondences between pairs of sentences, and we believe that they can provide a stable starting point for capturing information needed to predict entailment. System 2 treats entailment data as an aligned translation corpus, and performs its prediction based on a combination of metrics intended to measure translation quality.

		System 1	**System 2**
Pairs processed		587	800
Correctly	T	11/285	231/400
Labeled	F	292/302	238/400
Accuracy		0.52	0.59
Precision		0.52	0.59
Recall		0.04	0.58
F-measure		0.07	0.58
CWS		0.50	0.62

Figure 1: System results

All but one of these metrics come from libparis, a library of string similarity metrics assembled by MITRE. Some of these metrics are inspired by MT evaluation, and some are standard string-matching algorithms (Gusfield, 1997). Additionally, we used an MT alignment score, on which we now focus our discussion.

Statistical MT explicitly models the probability that a sentence *F* in a source language will translate to a target language sentence *E*. Following Brown et al. (1993), most statistical MT models decompose this probability into many probabilities relating individual word-pairs in the two sentences. There are also mechanisms in the models for explaining spurious words in the source and target, which align with nothing.

Figure 2 shows an alignment example from the training data described below. We see that most of the source words either align with their identical counterparts or disappear. Additionally, *surrounded* aligns with *engulf*, and *Bushehr* with *Iran*. In general, we only hope that the MT models capture this sort of synonymy and paraphrase; we do not expect that these simple word associations can represent any complicated inference.

MT models must be trained from a corpus of *F*-*E* pairs, typically larger by orders of magnitude than the development set provided for the RTE evaluation. For this volume of data, we turned to the Gigaword newswire corpus (Graff, 2003), hypothesizing that newspaper headlines are often en-

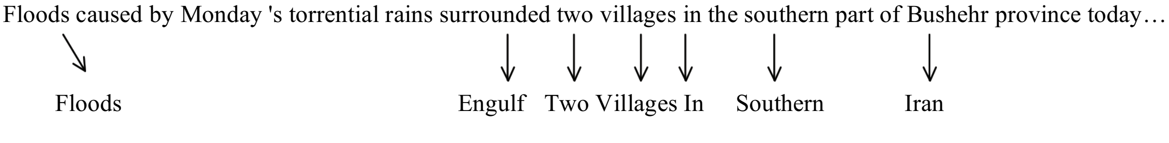

Floods caused by Monday 's torrential rains surrounded two villages in the southern part of Bushehr province today…

Figure 2: GIZA++ alignment for a training pair

differing	equal
heroism	gallantry
spaceflight	spacecraft
railmen	railworkers
procrastination	timing
hirsute	hair
engulf	surround
outplay	defeats
mountaineer	climber

Figure 3: A subset of the top word alignments acquired by GIZA++

tailed by the corresponding lead paragraph. We hoped that this noisy corpus would be a suitable training set for learning RTE alignments.

To test our hypothesis, we manually judged approximately 1000 of the lead-headline pairs from Gigaword for entailment (see Section 4 for a discussion of inter-annotator reliability). From this sample, we estimate that 60% of the headlines in the Gigaword corpus are entailed by the lead paragraph. We attempted to refine the data to acquire a smaller but less noisy corpus by training a document classifier (SVMlight: Joachims, 2002) to identify articles that exhibited the lead-entails-headline quality. Like those classifiers used to predict genre or topic, this training included the entire articles with bag-of-words features. We experimented with active learning techniques, and finally derived a 100,000-document subset of Gigaword with approximately 75% lead-entails-headline purity.

We used the GIZA++ toolkit (Och & Ney, 2003) to induce alignment models on the paired leads and headlines from the Gigaword subset. Some indicative word correspondences found by the model are shown in Figure 3. When applied to our (held-out) manually judged Gigaword data, these models could predict headline entailment with roughly 80% accuracy (compared to the base rate of 60% in that development set).

Unfortunately the alignment scores alone were next to useless for the RTE development data, predicting entailment correctly only slightly above chance. This is presumably because the negative instances in the RTE data are designed to have substantial conceptual overlap between the text and hypothesis, while the negative Gigaword instances frequently have little overlap.

At this point, we combined the alignment models with the libparis metrics described earlier. We first trained an SVM classifier on the RTE development data, using these features, but cross-validation experiments showed this to be unpromising as well—the data appeared to be far from linearly separable. In the end, we combined all the features using a simple *k*-nearest-neighbor classifier that chose, for each test pair, the dominant truth value among the five nearest neighbors in the development set. Results are shown in Figure 1.

4 The Corpus and the Evaluation

The RTE evaluation, while promising, faces a number of challenges as it matures.

First is the issue of the feasibility of the task. Based on our investigations, the task appears to be quite difficult for humans. When tested on 10 pairs from each of the seven application scenarios in the dev2 training set, our human judge achieved an agreement rate of 91% (64/70) compared to the given truth values. While this number might seem impressive, it is less so when one considers that the training data was already considerably simplified from a real-world application. According to the Task Definition, the *T-H* pairs were hand-crafted, and any pairs "for which there was disagreement among the judges were discarded." Thus, the 91% agreement is somewhat troubling.

We also attempted to determine the degree to which paraphrases played a role. Two of our researchers independently reviewed all the TRUE entailments of the dev2 set, and determined that 94% (131/140) were mere paraphrases (*John murdered Bill* → *Bill was killed by John*), as opposed to classic entailments (*Bill is dead*). During this process, we uncovered many cases where we disagreed with the given truth value on the grounds of synonymy (e.g., *in bloody clothes* → *covered in blood*). We also identified potential disagreements about the extent to which world knowledge is allowed to play a role. For instance, pair 102 (*domestic threat* → *threat of attack*) is more convincing if one understands the implications of *al Qaeda* and *September 11, 2001* mentioned in the text.

In the process of building our own training corpus (see Section 3), we conducted additional inter-judge studies. Even after one trial phase and with a supplementary set of guidelines in hand, the judges achieved only 81% inter-annotator agreement. While a portion of this disagreement is due to the messiness of the data (e.g., bylines and date lines mis-zoned into the headlines), the more egregious difficulty was that our judges found they had irreconcilable differences in meaning interpretation. For example, in the following lead-headline pair, one judge did not think that *safe operation* entailed (meant the same thing as) *operates smoothly*, and one did.

- □ *As of Saturday, Shanghai's Hongqiao Airport has performed safe operation for some 2,600 consecutive days, setting a record in the country.*
- □ *Shanghai's Hongqiao Airport Operates Smoothly*

It's hard to imagine how annotation guidelines would resolve this disagreement. This leads us, obviously, to wonder how an evaluation like this might be designed to ensure more consistent human judgment. It also suggests that if the organizers pre-clean the development corpus in future RTE evaluations as they did for this evaluation, it would be quite useful for them to report the percentage of pairs eliminated.

In addition to the challenges of interannotator agreement, it isn't clear what a "representative" corpus would look like. The RTE development corpus is clearly constructed to stress-test a range of legitimate and illegitimate inferences, but it is not clear how to balance these. It is unclear exactly how this technology will be used, and so it is equally unclear which issues might be more vs. less important to represent in an evaluation. Even in the cases where RTE data has been drawn from "naturally occurring" corpora, such as multiple, parallel translations, it's unclear how RTE technology would be applied to those corpora.

5 Conclusion and Future Work

It's been said, about difficult challenges like RTE, that one should be aware of the temptation to climb a tree in order to get to the moon (Dreyfus, 1979); i.e., short-term solutions can be initially superior, but are frequently dead ends. MITRE's two entries illustrate this dilemma quite clearly. System 1 is the rocket ship with nothing inside: fiendishly difficult to get off the ground, and unable to fly until a wide number of things work fairly well. System 2, on the other hand, is a tree. Our challenge, as we move forward, is to figure out how to leverage the strengths and potential of both.

Acknowledgments

This paper reports on work supported by the MITRE Sponsored Research Program. We would also like to extend our thanks to Fabrizio Morbini at the University of Rochester, without whose diligent support of EPILOG this work would not have been possible.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer, 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2).

Hubert Dreyfus, 1979. *What Computers Can't Do: A Critique Of Artificial Reason*. Harper and Row.

Christiane Fellbaum, 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

David Graff, 2003. *English Gigaword.* http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

Dan Gusfield, 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press.

Lynette Hirschman, Marc Light, Eric Breck and John D. Burger, 1999. Deep Read: A reading comprehension system. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*.

Thorsten Joachims, 2002. *Learning to Classify Text Using Support Vector Machines*. Kluwer.

Guido Minnen, John Carroll and Darren Pearce, 2001. Applied morphological processing of English. *Natural Language Engineering,* 7(3).

Franz Josef Och and Hermann Ney, 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1)

Adwait Ratnaparkhi, 1996. A maximum entropy part-of-speech tagger. *Proceedings of the Empirical Methods in Natural Language Processing Conference.*

Lenhart K. Schubert and Chung Hee Hwang, 2000. Episodic Logic meets Little Red Riding Hood: A comprehensive, natural representation for language understanding, in L. Iwanska and S.C. Shapiro (eds.), *Natural Language Processing and Knowledge Representation*. MIT/AAAI Press.

Daniel Sleator and Davy Temperley, 1993. Parsing English with a link grammar. *Third International Workshop on Parsing Technologies.*

Ben Wellner, Lisa Ferro, Warren Greiff and Lynette Hirschman, 2005. Evaluating language understanding through reading comprehension tests. *Natural Language Engineering* (to appear).

Can Shallow Predicate Argument Structures Determine Entailment?

Alina Andreevskaia, Zhuoyan Li and Sabine Bergler

Abstract

The CLaC Lab's system for the PASCAL RTE challenge explores the potential of simple general heuristics and a knowledge-poor approach for recognising paraphrases, using NP coreference, NP chunking, and two parsers (RASP and Link) to produce Predicate Argument Structures (PAS) for each of the pair components. WordNet lexical chains and a few specialised heuristics are used to establish semantic similarity between corresponding components of the PAS from the pair. We discuss the results and potential of this approach.

1 Introduction

Establishing entailment relationships between two statements is important for many NLP tasks (Szpektor et al., 2004) and the problem has attracted considerable interest in the research community. Most current work relies on the analysis of corpora - single or parallel - using machine learning and statistical methods (Lin and Pantel, 2001), (Chklovski and Pantel, 2004), (Dagan and Glickman, 2004), (Shinyama and Sekine, 2004), (Barzilay and Lee, 2003) to induce entailment-specific knowledge. In contrast, we approach the textual entailment problem using general mechanisms and strategies based uniquely on simplified predicate argument structure (PAS) and lexical chains (built using WordNet (WN) (Fellbaum, 1998)). This paper describes the results we achieved with this simple approach and suggests extensions to improve system performance.

2 System overview

The CLaC Lab's system for the PASCAL textual entailment challenge is based on systems our laboratory developed for text summarization. The environment is implemented in the GATE architecture (Cunningham et al., 2002) and provides tagging, NP chunking, and knowledge-poor fuzzy NP coreference resolution (Bergler et al., 2003), (Bergler et al., 2004), (Witte and Bergler, 2003). The flexible GATE architecture allows for the creation of modular components that can be used in different combinations depending on the task. For the purposes of the textual entailment resolution we extended the coreference system to incorporate verb groups, added full parsing, and included a few specialized heuristics for particular problems that were encountered in the PASCAL RTE challenge development set.

2.1 Main strategy

Two main types of information were used to assess the relatedness between the two parts of the pair: PAS and lexical chains. We use simplified shallow PASs that cover only the verb, its subject and object (if there was one) as in Figure 1.

PASs were extracted using the results of two parsers - the Link parser (Sleator and Temperley, 1993) and RASP parser (Briscoe and Carroll, 2002). One of these two parsers can be set as default, the second to be used only when the default parser doesn't produce a parse. If

Original Sentence:
<h>Two-thirds of the Scottish police force will be deployed at the happening.</h>
Constructed PAS:
<s:[Two-thirds of the Scottish police force] v:[will be deployed] a:[<p:[at] a:[the happening]>]>

Figure 1: Predicate argument structure

both parsers are given equal priority the system chooses for each sentence the parser that produces more PASs. Lexical chains were built using WN synsets. Different thresholds were tested. The smaller values mean closer relationships, 0 being the distance between members of the same synset.

```
Algorithm CLaC PASCAL
(* true: entailment detected, false otherwise )
Use the coreference resolution system to
    produce coference chains both for $t$ and $h$
    separately and for the pair as a unit
for each pair
    for each sentence
        Extract Noun Phrases and Verb Groups
        Select a parse among parses from two
            parsers with weighted scheme
        Determine the PAS based on the parsing,
            NP chunking and verb grouping results
Apply cardinality filter
    for each numeric value from $h$
        if there is no corresponding cardinality value in $t$
        then return false
Apply Predicate Argument Structure comparison
    Transform passive constructions into active ones
    for each PAS pair
        Compute WN distance for verbs in $t$ and $h$
        if WN distance <= threshold
            if both PASs are in comparable structures[1]
            if there is coreference between corresponding parts[2]
            then return true
Apply Be-Heuristic
    if $h$ contains the pattern "X is Y" and
        X$\in h$ and X'$\in t$ and $\{X, X'\}$ belong to
        the same inter-sentence coreference chain
        and Y$\in h$ and Y'$\in t$ and $\{Y, Y'\}$ belong
        to the same inter-sentence coreference chain and X' corefers with Y'
    then return true
return false
```

The algorithm favors precision over recall, therefore all entailment values are set to FALSE unless the system finds compelling evidence to the contrary.

Analysis of the development data allowed us also to develop some additional heuristics to handle specific cases. For example, we have implemented a *be-heuristic* for *h*-sentences of type "X is Y" that uses coreference chains in t and between t and h to decide whether X is Y given the data in t. The development data contains many examples of this kind in the QA task, but the phenomenon was less frequent in the test data. Another heuristic consists in comparing numbers in two parts of the pair to ensure that cases like pair 768 (Figure 2) from the development set do not produce false positives. This heuristic is applied as an initial filter before coreference chains are built.

<t>A small bronze bust of Spencer Tracy sold for £174,000.</t>
<h>A small bronze bust of Spencer Tracy made £180,447.</h>

Figure 2: Cardinality filtering example

2.2 Results

We submitted two runs, Table 1 presents the results of both runs, where RASP was the main parser and the Link parser used only as backup, RUN1 used a WN distance threshold of 1.

[1] *comparable structure* means they both have subject(s) and/or argument(s)

[2] e.g. subjects and/or arguments of the two PASs being compared

The second run used a WN threshold of 3. Our conservative strategy lead to a low number of true-positives: 72 true-positives of 400 in the gold standard in RUN2.

Task	RUN1			
	P	R	A	Cws
All	0.57	0.15	0.52	0.51
CD	0.89	0.32	0.64	0.64
IE	0.56	0.08	0.51	0.55
MT	0.40	0.10	0.47	0.43
QA	0.23	0.04	0.45	0.47
RC	0.52	0.17	0.51	0.48
PP	0.50	0.28	0.50	0.54
IR	0.62	0.11	0.52	0.49
Task	RUN2			
	P	R	A	Cws
All	0.55	0.18	0.52	0.52
CD	0.81	0.34	0.63	0.63
IE	0.64	0.12	0.52	0.57
MT	0.37	0.10	0.47	0.43
QA	0.31	0.08	0.45	0.49
RC	0.44	0.17	0.48	0.47
PP	0.50	0.36	0.50	0.56
IR	0.64	0.16	0.53	0.49

Table 1: Results over the different categories

3 Analysis and observation

Our main interest in participating in the PASCAL RTE challenge was to experiment with simple general purpose tools such as a coreference resolution system and a parser for textual entailment recognition.

The performance of our system is low, as expected, but comparable to the results shown by other systems. Most correct TRUE assignments occur when PASs are properly extracted and there is considerable similarity between PASs of *t* and *h*. This explains also the difference between our results for different tasks. CD, for instance, gave the highest precision (0.89 in Run1, 0.92 when WN distance=1 and parsers have equal weight), since pairs are mostly made up of sentences of similar structure (Figure 3) while QA consistently gave the worst results (precision below 0.30 and accuracy below 0.50), since it includes answers derived from statements of a totally different structure (Figure 3). In general, most of our false negatives are due to not recognising similarity between two syntactically different sentences. More sophisticated PASs that include additional constituents, such as adjuncts, and specialized heuristics geared towards frequent syntactic patterns in the data, as we did for the *be-heuristic* would address these issues.

<t>In terms of music, the National Philharmonic Orchestra draws large crowds.</t>
<h>The National Philharmonic orchestra draws large crowds.</h>

Figure 3: Correctly processed pair

<t>Working with fellow Canadians Charles Best and James Collip, Banting determined that insulin was the key to treating diabetes.</t>
<h>Banting conducted research of diabetes.</h>

Figure 4: False negative

The parser influences the system's performance (Table 2), best results are obtained when the choice between the Link and RASP parsers depends on the number of PASs produced, thus increasing the chances to find comparable PASs in *t* and *h* parts of the pair. Making PASs more complex by including prepositional phrases and adjuncts can eliminate such false positives as in Figure 5.

<t>The 69-page report is also the first major product of the Betsy Lehman Center for Patient Safety and Medical Error Reduction.</t>
<h>The 69-page report is the first major product of medical errors.</h>

Figure 5: False positive

Table 3 illustrates the influence of the WN distance threshold when both parsers have equal priority and the one producing more parses is preferred. Increasing the WN distance threshold leads to increased recall but reduced precision since more PASs are considered semantically related.

Setting	P	R	A	Cws
Equal priority	.59	.13	.52	.52
RASP/Link	.55	.13	.52	.51
Link/RASP	.58	.12	.52	.51

Table 2: Post-competition runs, WN distance=0

	P	R	A	Cws
WD= 0	.59	.13	.52	.52
WD= 1	.56	.15	.52	.52
WD= 2	.55	.16	.51	.52
WD= 3	.52	.18	.51	.52

Table 3: Influence of WN distance threshold (WD)

4 Conclusion

The PASCAL RTE challenge gave us an opportunity to create and test a system that we consider as a baseline system for our future work on event coreference and analysis of comparable documents. Our simple approach based on basic PASs and coreference resolution produced the precision sightly below 0.6 (up to 0.92 for the CD task). At the same time, the recall was fairly low - 0.18 (best value being 0.36 for PP task). These numbers can be improved be applying more sophisticated PASs and by creating additional heuristics to deal with specific patterns.

References

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL 2003*, pages 16–23.

Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. 2003. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Edmonton, Canada, May 31–June 1. NIST.

Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, Yunyu Chen, Monia Doandes, and Alina Andreevskaia. 2004. Multi-ERSS and ERSS 2004. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Boston, MA, May 6–7. NIST.

E. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1499–1504, Las Palmas, Canary Islands, May 2002.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.*

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Learning Methods for Text Understanding and Mining Workshop.* January.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical database.* MIT Press.

Dekang Lin and Patrick Pantel. 2001. DIRT-discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, pages 323–328.

Yusuke Shinyama and Satoshi Sekine. 2004. Paraphrase acquisition for information extraction. In *The Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP2003) at ACL 2003.* Sapporo, Japan, July.

D. D. Sleator and D. Temperley. 1993. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies.*

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 41–48, Barcelona, Spain, July. Association for Computational Linguistics.

René Witte and Sabine Bergler. 2003. Fuzzy Coreference Resolution for Summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24. Università Ca' Foscari.

VENSES – a Linguistically-Based System for Semantic Evaluation

Rodolfo Delmonte, Sara Tonelli, Marco Aldo Piccolino Boniforti, Antonella Bristot
Department of Language Sciences
University Ca' Foscari
30124 Venice, Italy
delmont@unive.it

Emanuele Pianta
ITC-IRST
POVO - TRENTO
pianta@itc.it

Abstract

The system for semantic evaluation VENSES (Venice Semantic Evaluation System) is organized as a pipeline of two subsystems: the first is a reduced version of GETARUN, our system for Text Understanding. The output of the system is a flat list of head-dependent structures (HDS) with Grammatical Relations (GRs) and Semantic Roles (SRs) labels. The evaluation system is made up of two main modules: the first is a sequence of linguistic rule-based subcalls; the second is a quantitatively based measurement of input structures. VENSES measures semantic similarity which may range from identical linguistic items, to synonymous or just morphologically derivable. Both modules go through General Consistency checks which are targeted to high level semantic attributes like presence of modality, negation, and opacity operators, temporal and spatial location checks.

Results in cws, accuracy and precision are homogenoues for both training and test corpus and fare higher than 60%.

1. Introduction

The system for semantic evaluation VENSES (Venice Semantic Evaluation System) is organized as a pipeline of two subsystems: the first is a reduced version of GETARUN, our system for Text Understanding; the second is the semantic evaluator which was previously created for Summary and Question evaluation and has now been thoroughly revised for the new more comprehensive RTE task.

The reduced GETARUN is composed of the usual sequence of submodules common in Information Extraction systems, i.e. a tokenizer, a multiword and NE recognition module, a PoS tagger based on finite state automata; then a multilayered cascaded RTN-based parser which is equipped with an interpretation module that uses subcategorization information and semantic roles processing. Eventually, the system is equipped with a pronominal binding module that works for lexical personal, possessive and reflexive pronouns, which are substituted by the heads of their antecedents - if available. The output of the system is a flat list of head-dependent structures (HDS) with Grammatical Relations (GRs) and Semantic Roles (SRs) labels. Notable additions to the usual formalism is the presence of a distinguished Negation relation; we also mark modals and progressive mood. All other non semantic elements like auxiliaries and determiners are erased.

The evaluation system uses a strategy of rewards/penalties for T/H pairs where text entailment is interpreted in terms of semantic similarity: the closest the T/H pairs are in semantic terms the more probable is their entailment. Rewards in terms of scores are assigned for each "similar" semantic element; penalties on the contrary can be expressed in terms of scores or they can determine a local failure and a consequent FALSE decision.

The evaluation system accesses the output of GETARUN which sits on files and is totally independent of it. It is made up of two main Modules: the first is a sequence of linguistic rule-based subcalls; the second is a quantitatively based measurement of input structures. The latter is basically a count of heads, dependents, GRs and SRs, scoring only similar elements in the H/T pair. Similarity may range from identical linguistic items, to synonymous or just morphologically derivable. As to GRs and SRs they are scored higher according to whether they belong to the subset of core relations and roles, i.e. obligatory arguments, or not, that is adjuncts. Both Modules go through General Consistency checks which are targeted to high level semantic attributes like presence of modality, negation, and opacity operators, the latter ones as expressed either by the presence of discourse markers of conditionality or by a secondary level relation intervening between the main predicate and a governing higher predicate belonging to the class of non factual verbs. Two other general consistency calls regard temporal and spatial location checks which must be identical or entailed in one another, if present – but see below.

Linguistic rule-based subcalls are organized into a sequence of calls going from rules containing axiomatic-like paraphrase HDSs which are ranked higher, to rules stating conditions for similarity according to the scale of argumentality which are ranked lower. All rules address HDSs, GRs and SRs. Both Modules strive for True assessments: however, Calls 1 are then followed by Calls 2 which can output True or False according to general consistency or scoring. Modifying the scoring function may thus vary the final result dramatically: it may contribute more True decisions if relaxed, so it needs fine tuning. More experimentation is needed on much bigger data set to achieve a more general definition of this function.

2. An A-As Hybrid Parser

Our parser has been presented in detail lately in a number of papers and has achieved 90% recall on Greval Corpus and 89% recall on the XEROX-700 corpus, limited only this latter test to SUBJ/OBJ GRs. As in most robust parsers, we use a sequence or cascade of transducers: however, in our approach, since we intend to recover sentence level structure, the process goes from partial parses to full sentence parses. Sentence and then clause level parsing are crucially responsible for the right assignment of Arguments and Adjuncts (hence A-As) to a governing predicate head. This is paramount in our scheme which aims at recovering predicate-argument structures, besides performing a compositional semantic translation of each semantically headed constituent.

The first transducer receives the input sentence split by previous processors, which is recursively/iteratively turned into a set of non-sentential level syntactic constituents - some of which can incorporate a PP headed by "of". Non-sentential level constituents, can be interspersed by heads which are subordinate clause markers, like subordinating conjuctions, or parentheticals - by punctuation, indirect interrogative clauses - by interrogative pronouns. The final output is a list of headed syntactic constituents which comprise the usual set of semantically translatable constituents, i.e., ADJP, ADVP, NP, PP, VC (Verb Cluster).

The task of the following transducer is that of creating clauses: we assume that at each sentence level only one VCluster (hence VC) can appear: we define the VC as IBAR indicating that there must be a finite or tensed verb included in it. VCs containing non-tensed verbal elements are all defined separately.

The third pass is intended to produce an improvement on the sentence-level full parse, by transducing each constituent label into a corresponding grammatical function label. The rules are taken from the inventory of LFG theory and follow its rules and principles. All attachment decisions are taken at this level of computation. In particular both PP and Relative Clauses are attached locally according to preferences and best match (but see Delmonte 2002). Finally the fourth pass has the task of splitting complex sentences into simplex ones, or clauses.

The output of the four transducers is passed to the algorithm that takes care of the creation of predicate-argument structures which has the additional task of taking into due account interclausal relations. To do that, semantic indices of governing predicates are used to assert dependencies between two adjacent clauses. This may also apply to a main clause and a clause-like adjunct like a gerundive or a participial.

Lexical information is accessed to confirm or modify previous decisions, particularly as regards OBLiques which will be interpreted as Adjuncts or Argument at this level of interpretation. We also assert Semantic Roles on the basis of lexical information (see Delmonte 1990).

To be compliant with usual Dependency Structure inventory of GRs which we also had to use for evaluation purposes, we eventually turn all predicative labels – NCOMP, ACOMP, PCOMP, VCOMP – into XCOMP. Also OBLiques are turned into IOBJect, unless they represent the passive agent by-adjunct which is assigned the GR label ARG_MOD. Then we produce flat Head-Dependent Structures.

We don't have space here to describe the Pronominal Binding module which however accesses Referential Heads at clause level and establishes possible antecedent-pronoun candidate lists which are then weighted and the best one chosen (but see Delmonte, Bianchi 1991).As an example consider Snippet 820 reported here below:

T. Clinton's new book is not big seller here.
H. Clinton's book is a big seller.

Whose structure is computed respectively as follows:
T.
be(adj-locative, here).
seller(ncmod, big).
book(ncmod-specif, 'Clinton-s_').
be(xcomp-prop, seller).
be(subj- theme_bound, book).
be(neg, not).
H.
seller(ncmod, big).
book(ncmod-specif, 'Clinton-s_').
be(xcomp-prop, seller).
be(subj-theme_bound, book).

The presence of the negation operator in the T portion of the snippet will prevent the evaluator from assessing to TRUE even though the relevant HD structures are identical.

3. The Semantic Evaluator (SE)

As said above, the SE is organized into two main modules: a quantitatively based module, and a sequence of rule-based subcalls where scoring is also taken into account when needed, to increase confidence in the decision process. The two modules must then undergo general consistency checks which have the task to ascertain the presence of possible mismatches at semantic level. In particular, these checks take care of the following semantic items:

- presence of spatiotemporal locations relatively to the same governing predicate, or a similar one as has been computed from previous modules;
- presence of opacity operators like discourse markers for conditionality having scope over the governing predicate under analysis;
- presence of quantifiers and other referentiality related determiners attached to the same nominal head in the T/H pair under analysis and chosen as relevant one by previous computation;
- presence of antonyms in the T/H pair at the level of governing predicates;
- presence of predicates belonging to the class of "doubt" expressing verbs, governing the relevant predicate shared by the T/H pair.

In some cases the General Consistency Checks have to be suspended: in particular whenever both T/H pairs contain opacity operators and negation, as for instance in,

Snippet no. 1014

The thick atmosphere of Titan makes it difficult for even the largest telescopes on Earth to see anything clearly.

Telescopes on Earth cannot see Titan clearly.

3.1 The Rule-Based Module

This Module is organized as a sequence of rule-based calls which start from exceptional cases down to default cases. Exceptional cases of Semantic Similarity are those constituted by definition-like H sentences, or simple paraphrases of the meaning expressed by the main predicate of the T text. Generally speaking, every time one such rule is fired, the T/H pair contains a conceptually complex lexical predicate and its paraphrase in conceptually simple components.

Examples of such cases are constituted by pairs like the following:

a. interview --> conduct an interview
b. pressurise --> apply pressure
c. treat --> receive treatment (provide)
d. fire → send letter of dismissal

where both a. and b. were actually present in WordNet while c. did not figure with the same predicates but rather with the one in brackets; d. was totally absent.

Definitions and paraphrases are looked up at first in the definitions made available by WordNet. In case of failure a list of some 50 manually made up axiomatic rules are accessed. Each such rule addresses main predicates in the T/H pair, together with presence of semantically relevant dependent if needed, and whenever the concept expressed by the lexically complex predicate requires it. Together with the predicates, the rules select relevant GRs and SRs when needed. In addition, more restrictions are introduced on additional arguments or adjuncts. Eventually, as is the case with all the rules, penalties are explored in terms of semantic operators of the main predicate like negation, modality and opacity inducing verbs which must either absent or be identical in the T/H pair.

The linguistically-based Module is organized into a sequence of five subcalls where the T/H pairs are checked for semantic similarity starting from sameness of main predicates to semantic approximate match.

The first subcall requires the presence of same HDs as main predicates with core arguments, i.e. the ones which have been computed as subject, object, indirect object, arg_mod (passive "by" agent adjunct), xcomp. Nonconflicting SRs are checked in all subcalls: i.e. subject-agent are allowed to match with arg_mod-agent and subject-theme_affected with object-theme_affected but not viceversa. These matches take care of what are usually referred to as lexical alternations for verb sucategorization frames, and lexical rules in LFG terms which encompass such syntactic phenomena as passive, intransitivization, ergativization, dative shift, etc.

The second subcall requires the presence of same HDs as a combination of main head and main dependent and at least another identical HD structure within the core argument subset. Other subcalls included in this group check nominalization derivational relations intervening between main predicate of T and H, which in one case is checked with edit distance measures.

The third subcall takes as input a list of "light-verbs" in semantic terms, i.e. verbs including "be", "have", "appear", and other similar copulative and locational verbs – like "live", "hold", "take_place", "participate", etc. - which are used to either make a definition, assert a property of the subject, individuate a location of the subject etc. These verbs are matched against main predicates and core arguments of the T portion, which must be identical to H. Quantitative measures are added to confirm the choice. Notable exceptions are sentences containing "be_born" predication which require specific constructions on the other member of the T/H pair.

The fourth subcall takes as input at least one identical main predicate HD non argument structure and one additional core argument or adjunct structure. Quantitative measures are added to confirm the choice.

The fifth subcall looks for different main predicates with core arguments which however must be non antonyms, non negative polarity and be synonyms. In addition, there must be at least another important identical non argument HD structure shared. Quantitative measures are added to confirm the choice. One such case is represented by

Snippet no. 1639

Lennon was murdered by Mark David Chapman outside the Dakota on Dec. 8, 1980.
Mark David Chapman killed Lennon.

Differently from what happens in real opposite meaning snippets where the SE considers SRs which must also be opposite, as in snippet 933,
Crude Oil Prices Slump
Oil prices drop

Or cases in which the snippet is rescued due to the presence of same SRs,
Snippet no. 876
Officials said Michael Hamilton was killed when gunmen opened fire and exchanged shots with Saudi security forces yesterday
Michael Hamilton died yesterday.

where DIE and KILL have opposite meaning but when KILL is used in the passive the SRs attached to their SUBJects will be identical.

3.2 The Quantitative Module

In this module all Heads, Dependents, GRs and SRs are collected for each member of the T/H pair and then they are passed to a scoring function that takes care of identical or similar members by assigning a certain score to every hit. Penalties correspond to high scores, while rewards correspond to low scores. A threshold is then set at a certain value which should encode the presence of a comparatively high number of identical/similar linguistic items.
As with previous subcalls, at the end of the computation semantic consistency and integrity is checked by collecting and comparing semantic operators, as well as performing a search of possible governing "doubt" verbs.
Generally speaking, we also treat short utterances differently from long ones. A stricter check is performed whenever an utterance has 3 or less HD structures, the reason being that in these structures some of the above mentioned subcalls would fail due to insufficient information available.

4. Evaluation and Discussion

The RTE task is a hard task: this may be partly due to the way in which it has been formulated – half of the snippets are TRUE, the other half are FALSE. It is usually the case that 10-15% mistakes are ascribable to the parser or any other analysis tool; another 5-10% mistakes will certainly come from insufficient semantic information. Whenever a system makes 20% errors this is doubled to 40% and the final result will become 60% overall Recall.
We looked into our mistakes to evaluate the import of the parser on the final Recall and we found out that: 10 snippets out of 100 TRUE ones have a wrong parse which can be regarded the main cause of the mistake. In other words only 10% of wrong results can be ascribed to bad parses. The remaing 10% is due to insufficient semantic information. In turn, this may be classified as follows:

- 80% is due to lack of paraphrases and definitions;
- 10% is due to wrong SemanticRole assignment;
- 10% is due to lack of synonym/antonym relations.

When we started working on the training corpus, verb predicates synsets made available by WordNet have been augmented by the information contained in Grady Ward's MOBY Thesaurus (http://www.dcs.shef.ac.uk/research/ilash/Moby/). Additional information has been derived from a manually reorganized version of Roget's Thesaurus, again limited though to verb predicates. We also felt we needed information related to negative polarity verb predicates which we derived from Harvard Dictionary derived from Harvard IV-4 e Laswell's dictionary on the Dynamics of Culture (http://www.wjh.harvard.edu/). The paraphrase and definition list for verb predicates taken from WordNet and transformed into HD structures was also updated in order to cover some missing cases. For instance, we had to implement a new paraphrase for the verb FIRE which is paraphrased as "send dismissal letter to" in snippet no. 783. The list of HDSs will be accessed by the Evaluator in the appropriate Module.

Test-set Results	Training-set Results
cws: 0.6257	**cws: 0.6396**
accuracy: 0.5925	**accuracy: 0.6032**
precision: 0.6242	**precision: 0.6261**
recall: 0.4650	**recall: 0.5088**
f: 0.5330	**f: 0.5614**
CD cws:0.7395 acc:0.6867	**CD cws:0.7416 acc:0.6633**
QA cws:0.5441 acc:0.5846	**QA cws:0.5719 acc:0.5444**
PP cws:0.8354 acc:0.8000	**PP cws:0.6846 acc:0.6707**
IE cws:0.6150 acc:0.5833	**IE cws:0.6192 acc:0.6000**
IR cws:0.6624 acc:0.6222	**IR cws:0.6749 acc:0.6286**
RC cws:0.5629 acc:0.5214	**RC cws:0.5422 acc:0.5243**
MT cws:0.4723 acc:0.4667	**MT cws:0.6482 acc:0.6111**

Tab.1 Results for training and test-set

5. References

Delmonte, R. 2003. Getaruns: a Hybrid System for Summarization and Question Answering. In Proc. Natural Language Processing (NLP) for Question-Answering, EACL, Budapest, ACL Columbia University, pp.21-28.

Delmonte R. 2002. GETARUN PARSER - A parser equipped with Quantifier Raising and Anaphoric Binding based on LFG, Proc. LFG2002 Conference, Athens, pp.130-153, at http://cslipublications.stanford.edu/hand/miscpubsonline.html

Delmonte, R. & D.Bianchi. 1991. Binding Pronominals with an LFG Parser, Proc. 2nd IWPT, Cancun(Messico), ACL 1991, pp. 59-72.

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 WordNet 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.

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.

- **lcs** This 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** 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

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

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), albeit 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.

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).

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 clockmaking 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*.

Figure 3: LCS features

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.

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 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.

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`.

Robust Textual Inference using Diverse Knowledge Sources

Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova
Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

We present a machine learning approach to robust textual inference, in which parses of the text and the hypothesis sentences are used to measure their asymmetric "similarity", and thereby to decide if the hypothesis can be inferred. This idea is realized in two different ways. In the first, each sentence is represented as a graph (extracted from a dependency parser) in which the nodes are words/phrases, and the links represent dependencies. A learned, asymmetric, graph-matching cost is then computed to measure the similarity between the text and the hypothesis. In the second approach, the text and the hypothesis are parsed into the logical formula-like representation used by (Harabagiu et al., 2000). An abductive theorem prover (using learned costs for making different types of assumptions in the proof) is then applied to try to infer the hypothesis from the text, and the total "cost" of proving the hypothesis is used to decide if the hypothesis is entailed.

1 Introduction

Below, we illustrate our methods with the following toy example of entailment:

```
TEXT: Chris purchased a BMW.
HYPOTHESIS: Chris bought a car.
```

Using relationships derived from syntactic dependencies, we can represent the text and hypothesis sentences equivalently as either a directed graph, or as a set of logical terms, as shown in Figure 1 and Section 3.1. In the graph, a vertex typically represents a word, but can also represent a phrase that is interpreted as a single entity. Labeled edges represent syntactic and semantic relationships tagged by various modules. The logical formula is derived by constructing a term for each node in the graph, and representing the dependency links with appropriately shared arguments. After presenting the inference methods, we show how the representations over which they work are derived from plain text.

2 Entailment by graph matching

We take the view that a hypothesis can be inferred from the text when the cost of matching the hypothesis graph to the text graph is low. For the remainder of this section, we outline a model for assigning a match cost to graphs.

For hypothesis graph H, and text graph T, a *matching* M is a mapping from the vertices of H to those of T; we allow nodes in H to map to a fictitious NIL vertex if necessary. Suppose the cost of matching M is Cost(M). Then we define the cost of matching H to T: MatchCost$(H, T) = \min_M$ Cost(M).

One simple cost model is given by the normalized sum of costs SubCost$(v, M(v))$ for substituting each vertex v in H for $M(v)$ in T:

$$\text{Cost}(M) = \frac{1}{Z} \sum_{v \in H_V} w(v)\ \text{SubCost}(v, M(v)) \qquad (1)$$

Here, $w(v)$ represents the weight or relative importance for vertex v, and $Z = \sum w(v)$ is a normalization constant. In our implementation, the weight of each vertex was based on the part-of-speech tag of the word or the type of named entity, if applicable. For hypothesis vertex v and text vertex $M(v)$, the substitution cost (in $[0, 1]$) is progressively higher for the following conditions:

- v and $M(v)$'s stem and POS / only stem match
- v is a synonym / hypernym of $M(v)$ (*WordNet*)
- v and $M(v)$'s stems are similar according to the word similarity modules (described later).

As (Punyakanok et al., 2004) demonstrated, models which also match syntactic relationships between words can outperform bag-of-words models for TREC QA answer extraction. As in (1), we can measure how relationally similar H and T are by a normalized sum of costs for substituting each edge relation (v, v') in H with the edge relation $(M(v), M(v'))$ in T. We assign a substitution cost for edge (v, v') in H based on the following conditions on path length:

- $M(v)$ is a parent/ancestor of $M(v')$
- $M(v)$ and $M(v')$ share a parent/ancestor

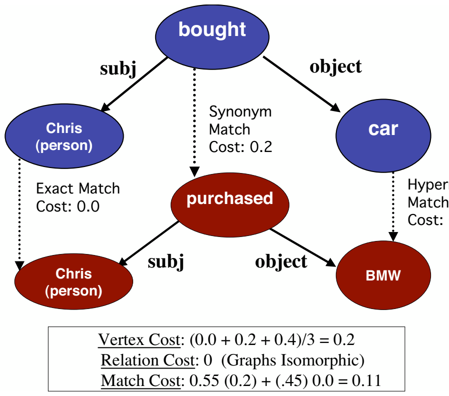


Figure 1: Example graph matching ($\alpha = 0.55$) for example pair in Section 2. Dashed lines represent mapping.

As in the vertex case we have weights for each hypothesis edge, $w(e)$, based upon the edge's label; typically subject and object relations are more important to match than others. Our final matching cost is given by a convex mixture of the vertex and relational match costs:
$\text{Cost}(M) = \alpha\text{VertexCost}(M) + (1-\alpha)\text{RelationCost}(M)$.

Notice that minimizing $\text{Cost}(M)$ is computationally hard since $\text{RelationCost}(M) = 0$ if and only if H is isomorphic to a subgraph of T. As an approximation, we can efficiently find the matching M^* which minimizes $\text{VertexCost}(\cdot)$ using the Hungarian method (Kuhn, 1955); we then perform local greedy hillclimbing search, beginning from M^*, to approximate the minimal matching.

3 Abductive theorem proving

This method works with a logical formula-like representation (Harabagiu et al., 2000) of the syntactic dependencies in the text and hypothesis sentences. The basic idea is that a hypothesis that can be logically derived from the text is entailed by it. Such a logical derivation is called a "proof" of the hypothesis.

The logical formulae capture only the syntactic dependencies in the sentences. Consequently, several entailed hypotheses that require semantic rewrites (such as "a *BMW* is a *car*") can be derived from the corresponding text formulae only by using additional assumptions in the proof. We do not use explicit logical axioms ("rules") for these assumptions; instead, each assumption that unifies one term in the hypothesis with another in the text is assigned a cost based on the judged plausibility of that assumption. This cost is computed using particular features of the assumption.

Using such a cost model, the inference procedure searches for a minimum cost proof for the hypothesis. The hypothesis is judged to be entailed from the text if it has a proof with cost below a certain learned threshold value.

We also provide a procedure to learn good costs for assumptions from a training set containing examples of entailed and non-entailed hypotheses.

3.1 Representation

For the example, the following logical representation is derived, with each number/letter representing a constant:

```
T: Chris(1) BMW(2) purchased(3,1,2)
H: Chris(x) car(y) bought(e,x,y)
```

Each predicate and each argument is also annotated with other linguistic information not shown here (such as semantic roles and named entity tags) for use in assigning costs to assumptions.

3.2 Inference

For our representation, proof steps that unify one term from the text with one term of the hypothesis suffice. We allow any pair of terms to unify with each other, with a cost assigned by the *assumption cost model*. We relax the requirements for logical unification in several ways, adding cost penalties for each such relaxation:

1. Terms with different predicates can be unified; the cost penalty is obtained using the term similarity measures (described later) and the linguistic annotations on the predicates.
2. The terms can have differing number of arguments, and the arguments of one term can be matched with those of the other term in any order. Each argument matching is assigned a cost based on the compatibility of the annotations of those arguments. A term pair might be unified in many ways corresponding to different argument matchings.
3. Constants can be unified with each other at an appropriate cost. This cost is precomputed for all constant pairs in a particular example, and is lowered for specific pairs—such as when there is possible coreference or appositive reference.

We developed a specialized abductive theorem prover to discover the minimum cost proof using uniform cost search. For our running example, the minimum cost proof unifies `BMW` with `car`, and `purchased` with `bought`, at small costs.

3.3 Learning good costs for assumptions

Given a training set of labeled text-hypothesis pairs (such as the RTE development set), we propose a learning algorithm that tries to learn good assumption costs.[1]

[1] Details are omitted here due to space constraints. See (Raina et al., 2005) for details.

4 Producing representations and similarities for inference

4.1 Syntactic processing

The first steps of the front-end deal with tokenization and parsing. Beyond this base level, the performance of the inference methods depends critically on our ability to identify similarities and differences between our fairly syntactic representations of the text and the hypothesis. This is largely dependent on being able to perform normalization and enrichment tasks that will reveal essential similarities, and on having good measures of lexical semantic similarity between words and larger units.

We do deterministic tokenization and then use full sentence parsing to reveal syntactic dependencies. The parser used was a variant of (Klein and Manning, 2003). The most important addition was training on an extra dozen sentences that gave the parser some exposure to topics in the news in 2005 rather than only those appearing in 1989. Exploiting headedness relations and hand-written pattern-matching rules, the parse tree is converted into a set of typed dependencies between words, representing grammatical relations (like subject and object) and other modifier dependencies, including such things as appositives, negations, and temporal modifiers. This is the basis of the graph structure in Figure 1. Various collapsings are then done to normalize and improve this dependency representation. Prepositions and possessive *'s* are changed from being vertices to relation names, and coordinations explicitly represent the conjuncts. A conditional random field (Lafferty et al., 2001) named entity recognition system is run to identify seven classes (Person, Organization, Location; Percent, Time, Money, and Date). The first three are collapsed into single nodes tagged NNP (proper noun) prior to parsing, while the latter four are grouped after parsing, but before the conversion to a dependency representation, and their values are normalized into a canonical form using hand-written regular expressions. This includes representing approximate and relative quantities (*around $40* and *less than 2 dollars*) as well as exact amounts. At the same time, we also collapse collocations, which are found in WordNet, like *back off* and *throw up* to a single node.

4.2 Additional dependencies between nodes

We augment the syntactic dependency graph with semantic role arcs using a Propbank-trained semantic role labeler (Toutanova et al., 2005). For each verb, we added edges between that verb and the head word of each of its arguments, and labeled the edges with the appropriate semantic role. This allowed us to add relations (between words) that were not captured by surface syntax, and also to classify modifying phrases as temporal, locative, and other categories. We added coreference relations between noun phrases and named entities using a maximum entropy coreference classifier modeled after (Soon et al., 2001).

Dataset	`General`		`ByTask`	
	Accuracy	CWS	Accuracy	CWS
DevSet1	64.8%	0.778	65.5%	0.805
DevSet2	52.1%	0.578	55.7%	0.661
DevSet1 + DevSet2	58.5%	0.679	60.8%	0.743
Test set	56.2%	0.620	55.2%	0.686

Table 1: Accuracy and confidence weighted score (CWS) on RTE datasets.

Task	`General`		`ByTask`	
	Accuracy	CWS	Accuracy	CWS
CD	79.3%	0.903	84.0%	0.926
IE	47.5%	0.493	55.0%	0.590
IR	56.7%	0.590	55.6%	0.604
MT	46.7%	0.480	47.5%	0.479
PP	58.0%	0.623	54.0%	0.535
QA	48.5%	0.478	43.9%	0.466
RC	52.9%	0.523	50.7%	0.480

Table 2: Accuracy and confidence weighted score (CWS) split by task on the RTE test set.

4.3 Methods for discovering term similarity

As in other work, e.g., (Moldovan et al., 2000), we relied on WordNet (Miller, 1995) heavily for lexical knowledge. The `WordNet::Similarity` module (Pedersen et al., 2004) was used to compute a symmetric similarity score between two phrases. If the queried phrases are listed as antonyms in WordNet, the match is given a very high cost in the inference procedures. Derivational forms in WordNet are used to detect nominalized events and modify the representation (e.g., *murder of police officer* entails *police officer killed*). WordNet does not include prepositions. We semi-automatically constructed a matrix of preposition similarity values using synonyms (e.g., *over* and *above*) and antonyms (e.g., *over* and *under*). Synonyms were found by grouping prepositions into clusters. Antonym pairs were added manually. Finally, we compiled a list of 206 countries and their derivatives manually (e.g., *Philippines - Filipino*), and collected a list of 276 frequently occurring acronyms in a large corpus, and recorded their expansions.

The inference procedures require considerable semantic knowledge to infer some rewrites using just phrasal dependencies; for example, *won victory in Presidential election* might entail *became President*. We attempted to discover such rewrites by looking for similarly placed phrases in a large corpus, using a backed-off modification of the similarity measure described in (Lin and Pantel, 2001).

Sometimes both of these methods are too precise. Words that are used in the same context often do not have explicit relationships between them; for instance *marathon* and *run* clearly have a semantic relationship not considered in the WordNet hierarchy. To overcome this we used `Infomap`,[2]

[2] Available at `http://infomap.stanford.edu`.

Text	Hypothesis	Our answer	Conf	Comments
A Filipino hostage in Iraq was released.	A Filipino hostage was freed in Iraq. *(TRUE)*	True	0.61	Verb rewrite is handled. Phrasal ordering does not affect cost.
The government announced last week that it plans to raise oil prices.	Oil prices drop. *(FALSE)*	False	0.69	High cost given for substituting word for its antonym.
Shrek 2 rang up $92 million.	Shrek 2 earned $92 million. *(TRUE)*	False	0.51	Collocation "rang up" is not known to be similar to "earned".
Sonia Gandhi can be defeated in the next elections in India by BJP.	Sonia Gandhi is defeated by BJP. *(FALSE)*	True	0.66	"can be" does not indicate the complement event occurs.
Fighters loyal to Moqtada al-Sadr shot down a U.S. helicopter Thursday in the holy city of Najaf.	Fighters loyal to Moqtada al-Sadr shot down Najaf. *(FALSE)*	True	0.67	Should recognize non-Location cannot be substituted for Location.
C and D Technologies announced that it has closed the acquisition of Datel, Inc.	Datel Acquired C and D technologies. *(FALSE)*	True	0.59	Failed to penalize switch in semantic role structure enough

Table 3: Analysis of results on some RTE examples.

an open-source implementation of Latent Semantic Analysis (Deerwester et al., 1990), to score words according to distributional similarity (measured using the British National Corpus). To further exploit distributional similarity, we also implemented a measure of similarity that is computed as the ratio between the number of search results from `google.com` for two phrases when queried separately and in combination.

5 Results and analysis

Our overall system is a combination of the two systems described in Sections 2 and 3. Each system produces a real number score that is normalized to have zero mean and unit variance, and then converted to a confidence value using the cumulative distribution function for a normal distribution. These individual scores are then linearly combined using logistic regression, with the weights trained on the RTE development sets. The first version (called `General`) trained one set of weights for all RTE tasks; the second version (called `ByTask`) trained separate weights per task. All parameters except the classifier weights were identical.

Table 1 reports the performance of our final classifiers on different datasets. Table 2 shows the performance separately on each task in the test set.

A random guessing baseline achieves accuracy 50% and confidence weighted score (CWS) 0.50. Our test set accuracy is only a few points above random guessing; however, the CWS is significantly higher. Thus, our predictions are well-calibrated and more robust; this is probably because our learning and classifier combination procedures maximize the likelihood of the full predicted distribution rather than just a binary accuracy value.

Table 3 has an analysis of some examples from the RTE datasets. The term similarity routines seemed most important for good performance, while many of the other modules are useful in specific cases. Many of the language resources used were sparse (e.g., antonyms in WordNet); high-recall resources would be extremely beneficial.

Acknowledgments

This work was supported by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) program.

References

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

S. Harabagiu, M. Pasca, and S. Maiorano. 2000. Experiments with Open-Domain Textual Question Answering. *COLING 2000.*

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. *ACL-2003*, 423–430.

H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML-2001.*

D. Lin and P. Pantel. 2001. DIRT: Discovery of Inference Rules from Text. *KDD 2001.*

G. Miller. 1995 WordNet: A lexical database. *Communications of the ACM*, 38(11):39-41.

D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. 2000 The structure and performance of an open-domain question answering system. *ACL-2000*, 563–570.

T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::Similarity - Measuring the relatedness of concepts. *AAAI-2004.*

V. Punyakanok, D. Roth, and W. Yih. 2004 Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics.*

R. Raina, A. Y. Ng, and C. D. Manning 2005 Robust textual inference via abduction and learning. (Unpublished manuscript).

W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

K. Toutanova, A. Haghighi, and C. D. Manning 2005. Joint learning improves semantic role labeling. *ACL-2005.*

Textual Entailment Resolution via Atomic Propositions

Elena Akhmatova
Centre for Language Technology
Division of Information and Communication Sciences
Macquarie University
North Ryde, NSW, 2109
elena@ics.mq.edu.au

Abstract

This paper presents an approach to solving the problem of textual entailment recognition and describes the computer application built to demonstrate the performance of the proposed approach. The method presented here is based on syntax-driven semantic analysis and uses the notion of atomic proposition as its main element for entailment recognition. The idea is to find the entailment relation in the sentence pairs by comparing the atomic propositions contained in the text and hypothesis sentences.

The comparison of atomic propositions is performed via an automated deduction system OTTER; the propositions are extracted from the output of the Link Parser; and semantic knowledge is taken from the WordNet database. On its current stage the system is capable to recognize basic semantically and syntactically based entailments and is potentially capable to use more external and internal knowledge to deal with more complex entailments.

1 Introduction

The variety of ways to transmit the same information is an interesting phenomenon of natural language and is an obstacle for many applications in the domain of natural language processing. Question answering, for example, has faced the fact that a possible answer to a question could be expressed in a way that is syntactically and semantically different from the question sentence, or has to be entailed from it. The paper is devoted to the phenomena of entailment. By textual entailment is understood a relationship between a coherent text T and a language expression H, which is considered as a hypothesis. T entails H if the meaning of H, as interpreted in the context of T, can be deduced from the meaning of T. By a *language expression* is understood a syntactically coherent text fragment, having a well formed fully connected syntactic analysis (Dagan and Glickman, 2004). For example,

T: Coffee boosts energy and provides health benefits.
H: Coffee gives health benefits.

is a true textual entailment that will be used as an example throughout the paper.

2 Meaning Representation

To know if a hypothesis H is entailed from a text T one should compare their meanings. We represent meaning of a sentence as a set of atomic propositions contained in it and compare the propositions in order to compare the sentences. We mean by an *atomic proposition* a minimal declarative statement (or a small idea) that is either true (T) or false (F) and whose truth or falsity does not depend on the truth or falsity of any other proposition. (*Coffee boosts energy and provides health benefits.* – propositions are: *Coffee boosts energy.* and *Coffee provides health benefits.*)

To break a sentence into its atomic propositions a syntax-driven semantic analysis of the sentence (Jurafsky and Martin, 2000) is applied, as we believe that a deep semantic and syntactical analysis is vital to solve the problem.

The implementation of the method uses an output of the parser as an input for the semantic analyser producing the output from which a first-order logic representation of the meaning can be derived.

```
        +------------------------------------------------ Xp ---------------------------------------+
        +------ Wd ----+---- Ss ----+---- Os ----+                                                  |
        |              |            |            |                                                  |
LEFT-WALL coffee.n boosts.v energy.n and provides.v health.n benefits.n.

⇒ [coffee ]subj [provides ] [health benefits ]obj
exists x exists y exists z exists za6 (Subj(x) &iq(x, 'coffee') &Pred(y)&iq(y,
'provides') & Obj(z)&iq(z, 'benefits')&attr(z, za6) & Obj(za6)& iq(za6, 'health')).

          +------------------------------------------------ Xp ---------------------------------------+
          |                                                        +------------ Op -----------+     |
          +------ Wd ----+----------------------- Ss ----------------------+    +---- AN ---+   |     |
          |              |                                                  |    |           |   |     |
   LEFT-WALL coffee.n boosts.v energy.n and provides.v health.n benefits.n.

⇒ [coffee ]subj [boosts ] [energy ]obj
exists x exists y exists z (Subj(x) &iq(x, 'coffee') & Pred(y) &iq(y, 'boosts')
& Obj(z)&iq(z, 'energy') ).
```

...
↓
linkage array
↓
propositions
↓
logic form
↓
...

Figure 1. Data of the system on different stages of analysis.

This final meaning representation is called *the logic formula* of the sentence. See figure 1 for an example of the data the system has.

A logical meaning representation of a sentence and an automatic deduction system to work with it are often used in QA applications (Moldovan et al., 2002; Moldovan et al., 2003). There exist many approaches to describe meaning by means of a logical form. Thus, a sentence *A restaurant serves meat* can have a description

exists e, x Isa(e, Serving) & Server(e, x) & Served(e, Meat) & Isa(x, Restuarant) (Jurafsky and Martin, 2000).

From our point of view, these forms are rigid and hard to produce. There are no automatic rules to understand that the event here is *serving*, and the subject of the sentence is a *server*. It's not clear how to compare two such logical representations.

As a result we use a simplified representation in this version of the system. There are three types of objects - *Subj(x), Obj(x), Pred(x)* and a meaning attaching element *iq(x, <meaning of x>)*. See fig. 1 for an example. Also, there are two variants of relationships *attr(x, y)* and *prep(x, y):*

"Somali capital" -- *Subj(x) & iq(x, 'capital') & attr(x, y) & Subj(y) & iq(y, 'somali').*

"a zoo in Berlin" - *Obj(x) & iq(x, 'zoo') & prep(x, y) & Obj(y) & iq(y, 'Berlin').*

In this case a logic formula is easy to build automatically. Semantic synonymy is expressed as an equivalence (*iq*(*x*, 'serve') <-> *iq*(*x*, 'dish')) and hyperonymy ((*iq*(*x*, 'serve') -> *iq*(*x*, 'provide')), (*iq*(*x*, 'serve') -> *iq*(*x*, 'cater'))). We can store lexical relation rules, *all x (iq(x, 'is') <-> iq(x, 'be'));* describe syntactical equivalence by means of additional logic rules, "Be X of Y -> X Y" (*director of the firm -> direct a firm*) is *all y z z1 y77 z77 ((Pred(z1) & iq(z1, 'be') & Obj(y) & iq(y, y77) & prep(y, z) & Obj(z) & iq(z, z77)) -> (Pred(y) & iq(y, y77) & Obj(z) & iq(z, z77)));* or soften some mistakes of the parser, such as prepositional attachment - *all x y y77 z z77 x77 (Pred(y) & iq(y, y77) & prep(y, x) & Obj(x) & iq(x, x77) & Obj(z) & iq(z, z77) <-> Pred(y) & iq(y, y77) & Obj(z) & iq(z, z77) & attr(z, x) & Obj(x) & iq(x, x77)).* The rules are called *knowledge rules*, as they represent knowledge of the system.

3 WordNet Relatedness

A WordNet (WordNet) relatedness algorithm used in the system was developed specially for this system, as the existing ones (Budanitsky and Hirst, 2001) are not quite right for the system. The result of its work is a relatedness score. It is used to prove the synonymy or entailment relation between words (see figure 3 for details). As it compares senses of the words, a WSD algorithm could be used (will be in future) prior to the comparison to get a more reliable score (otherwise the probability that the current word has sense *i* could be estimated as 1/*n*, where *n* – number of senses the word has). The score is calculated from the paths between the senses of the words in the graph. We use the length of a path (*take over–buy* has a length 2, *form-make-establish* has a length 3; the longer the path, the less is the relatedness); the amount of sense of the words that is on the path between these two words (two words connected with a verb *print*, for example, are more close to each other, as the words connected via *make*, because *print* has only 4 senses and make has 49); and the total number of different paths (the words which senses are connected through 10 different paths are more related then the words having only one connecting path, for example). Though we compare all words similarly now, I would like to emphasize that the following method ideally should be used only for verbs and nouns derived from them, and a different one in other cases, for, intuitively, the verbs (think about *decide* and *conclude*, and the nouns *decision* and *conclusion* derived from them) have a more generic meaning, than the nouns describing particular objects (*train*, *car*, *bus*).

4 System description

The scheme of the system is presented on the figure 2. The Link Parser 4.1a (Link Grammar) to trace the connections between the elements of sentences and a version 3.3 of OTTER (OTTER) for comparison of the atomic propositions are used now. A way to logic form is shown in section 2. After the algorithm is the following (figure 2: Otter and its input data; and figure 4): if for every proposition in the hypothesis sentence there is one in the text sentence from that it could be entailed then the sentence entailment holds, otherwise the entailment does not hold.

The same algorithm also can be used to obtain some knowledge rules from the data set: when entailment holds we want to find pairs <p1, p2> (see fig. 2) and to build knowledge rules p1->p2 to use them later. The idea of this process is the following: for every atomic proposition Y in the hypothesis find the atomic proposition in the text from which it is entailed. If there is none, find the closest (with the higher relatedness score according to WordNet (see section 7)) atomic proposition X and create a rule Prop X -> Prop Y. So, what we can learn is:

reduce the risk of diseases -> have health benefits.

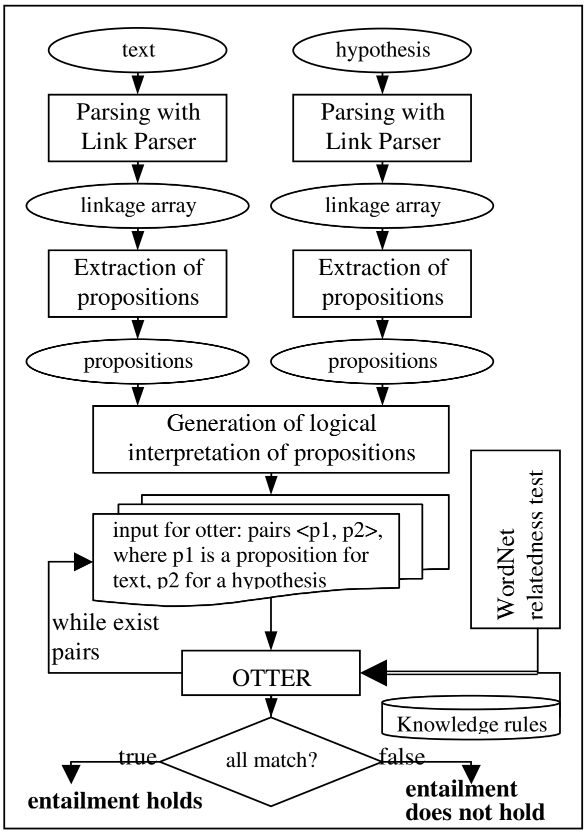
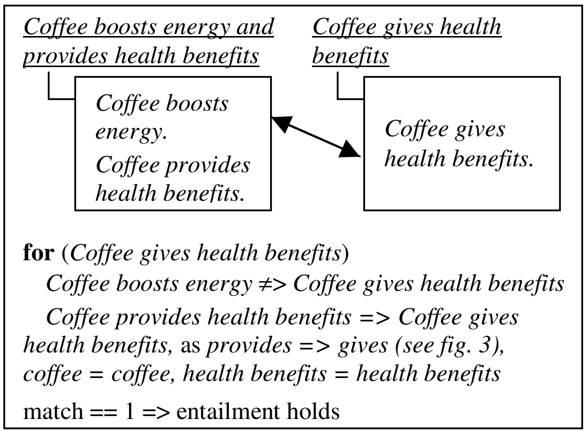


Figure 2. System architecture.



Coffee boosts energy and provides health benefits
Coffee gives health benefits
Coffee boosts energy.
Coffee provides health benefits.
Coffee gives health benefits.

for (*Coffee gives health benefits*)

Coffee boosts energy ≠> Coffee gives health benefits

Coffee provides health benefits => Coffee gives health benefits, as *provides => gives (see fig. 3), coffee = coffee, health benefits = health benefits*

match == 1 => entailment holds

Figure 4. Comparison of propositions.

provide – give **relatedness score: 0,3623**
(verb chain); maximum path length – 3 => ∀ x (*iq*(*x*, 'provide') -> *iq*(*x*, 'give')).

1. **provide#1(7)**[2259805] -- *hyperonym* -- **give#3(44)**[2136207]

...

7. **provide#6(7)**[2155855] -- *hyperonym* -- **support#2(11)**[2155507] -- *hyperonym* -- **give#3(44)**[2136207]

Figure 3. WordNet relatedness algorithm. Data and results.

5 Performance of the system

First, the examples where entailment holds and it is right.

T: *The decision is made.* - H: *The determination is made.*

As *decision* and *determination* are connected via WordNet, so we'll have a rule *all x (iq(x, 'decision') <-> iq(x, 'determination'))* .

T: *The good decision is made.* - H: *The decision is made.*

Subj(x) & iq(x,'decision') & attr(x, y) & Subj(y) & iq(y, 'good') -> Subj(x) & iq(x,'decision')

T: *The Brazilian president visited France.* - H: *The president of Brazil visited France,* and T: *The boy goes to school by bus.* - H: *The boy travels with school bus.*

A rule *all x y (attr(x, y) <-> prep(x, y))* works here.

T: *The man is a director of the company.* – H: *The man rules the company.*

"Be X of Y -> X Y" (section 4) rule is used here.

Now, the examples where entailment holds though it shouldn't:

T: *The population of France has grown during the last 3 years.* – H: *The population of Paris has grown during the last 3 years.*

T: *The gastronomic capital of France is Lyon.* – H: *The capital of France is Lyon.*

T: *The man came to the park by car.* – H: *The man came to a car park.*

It is clear now why the following two examples were recognized as TRUE entailments:

T: *A male gorilla escaped from his cage in the Berlin zoo and sent terrified visitors running for cover, the zoo said yesterday.* – H: *A gorilla escaped from his cage in a zoo in Germany.*

T: *The incident in Mogadishu, the Somali capital, came as U.S. forces began the final phase of their promised March 31 pullout.* **–** H: *The capital of Somalia is Mogadishu.*

6 Results

cws: 0.5067; accuracy: 0.5188 ;
precision: 0.6119; recall: 0.1025; f: 0.1756

task	cws	accuracy	task	cws	accuracy
CD	0.6121	0.5867	RC	0.4702	0.5214
IE	0.5519	0.5083	PP	0.5452	0.5200
MT	0.4341	0.4917	IR	0.4797	0.5111
QA	0.4649	0.4769			

Note: according to *Recognising Textual Entailment Challenge* evaluation method (Pascal Challenges).

The results are low now, as more work should be done for proposition extraction and logical representation. Also a good knowledge rule database is missing.

7 Future work

Despite not very high results we believe the proposed system has a strong potential. The main future tasks are: to make inferences inside the text sentence itself, to try reasoning with all propositions from the text, and to create an inference rule database. An attempt will be done to construct the database using sentences with inferences inside them. That is the sentences with the conjunctions *as result of*, *because*, *if*, and predicates *cause*, *follow*.

References

Alexander Budanitsky and Graeme Hirst. 2001. *Semantic Distance in WordNet: An Experimental, application-oriented evaluation of five measures.* Proceedings of the NAACL–2001 Workshop on WordNet and Other Lexical Resources, Pittsburg, PA.

Cordell Green. 1969. *Theorem-Proving by Resolution as a Basis for Question-Answering Systems*. Machine Intelligence, Chapter 11. Edinburgh University Press, pp. 183-205.

Dan Moldovan et al. 2002. *LCC Tools for Question Answering*. NIST Special Publication 200 – 251.

Dan Moldovan et al. 2003. COGEX: A Logic Prover for Question Answering. Proc. HLT-NAACL 2003, Edmonton, 2003.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Prentice-Hall. Chapter 14-15.

Ido Dagan and Oren Glickman. 2004. Probabilistic Textual Entailment: *Generic Applied Modeling of Language Variability. Learning Methods for Text Understanding and Mining*, 26 - 29 January 2004, Grenoble, France.

Link Grammar. http://www.link.cs.cmu.edu/link/.

OTTER. www-unix.mcs.anl.gov/AR/otter.

Pascal Challenges. *Recognising Textual Entailment Challenge*. http://www.pascal-network.org.

WordNet. www.cogsci.princeton.edu/~wn.

Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment

Johan Bos
School of Informatics
University of Edinburgh
`jbos@inf.ed.ac.uk`

Katja Markert
School of Computing
University of Leeds
`markert@comp.leeds.ac.uk`

Abstract

We combine two methods to tackle the textual entailment challenge: a shallow method based on word overlap and a deep method using theorem proving techniques. We use a machine learning technique to combine features derived from both methods. We submitted two runs, one using all features, yielding an accuracy of 0.5625, and one using only the shallow feature, with an accuracy of 0.5550. Our method currently suffers from a lack of background knowledge and future work will be focussed on that area.

1 Introduction

In this paper we summarise our submission to the 2004/5 Recognising Textual Entailment (RTE) challenge. In this task, given a pair of text fragments—a text (T) and an hypothesis (H)—the system has to decide whether the hypothesis is entailed by the text. The system we developed is a hybrid system, using both shallow and deep semantic analysis methods.

The shallow techniques establish a baseline performance, but also complement the deep semantic analysis. In the hybrid system, each T/H-pair is represented by feature-value vectors that are derived from either shallow or deep semantic analysis. The features used are domain-independent to increase scalability. An off-the-shelf machine learning tool was then used to derive a decision tree model from the RTE development set.

2 Shallow Semantic Analysis

The shallow semantic analysis measures only word overlap between text and hypothesis. Both text and hypothesis are tokenised and lemmatised. Each lemma in the hypothesis is assigned its inverse document frequency, using the Web as corpus, as its weight. This standard procedure allows us to assign more importance to less frequent words.

The word overlap `overlap` between text and hypothesis is initialised as zero. Should a lemma in the hypothesis also occur in the text, its weight is added to `overlap`, otherwise it is substracted. In the end `overlap` is normalised by dividing it by the sum of all weights of the lemmas in the hypothesis. This ensures that `overlap` is always a real number between 1 and -1 and also ensures independence of the length of the hypothesis.[1]

Training a decision tree on the development set with this feature alone yielded the following tree for entailment, where TRUE associates with entailment, and FALSE does not:[2]

```
overlap <= 0.161146: FALSE
overlap > 0.161146: TRUE
```

Accuracy on the development set (using 10-fold cross-validation) was 0.594 and therefore clearly beat the baseline of 0.50. In general this method overestimates the number of true entailments in the development set and achieved an F-measure of 0.672 for the class TRUE and only 0.474 for the class

[1] This word overlap measure is similar to the method used in (Monz and de Rijke, 2003) and (Saggion et al., 2004) —however, they do not substract from the overlap measure a token in the hypothesis which does not appear in the text. Hence, their scores are within 0 and 1. We experimented with this variation on the development set, but achieved slightly better performance with the scores that used substraction as well.

[2] We used Weka's J48 classifier (`http://www.cs.waikato.ac.nz/~ml/weka/`) for all experiments in this paper. We also used Weka's confidence values for confidence weighting scores.

FALSE. We submitted this baseline as Run2 and the performance on the RTE test set was as follows:

```
cws:        0.5864
accuracy:   0.5550
precision:  0.5375
recall:     0.7875
f:          0.6389
```

Although the performance is still significantly better than the baseline (5% level), it is worse than on the development set, because the level of word overlap in the test set was lower overall than in the development set. This seems to be an indicator of a different design of development and test set—using 10-fold cross-validation on the test set indicates that an `overlap` value of between -0.20 and 0.92 already indicates a TRUE value in the test set, whereas a value of over 0.92 indicates a FALSE value. The latter anomaly, which indicates that if text and hypothesis are very similar then the entailment is false, is due to the fact that there are many examples in the test set that are deliberately constructed to have a high word overlap but nevertheless be FALSE.

3 Deep Semantic Analysis

We use a robust wide-coverage CCG-parser (Bos et al., 2004) to generate fine-grained semantic representations for each T/H-pair. The semantic representation language is a first-order fragment of the DRS-language used in Discourse Representation Theory (Kamp and Reyle, 1993). To check whether an entailment holds or not, we used Vampire, a theorem prover for first-order logic (Riazanov and Voronkov, 2002), and Paradox, a finite model builder (Claessen and Sörensson, 2003).

To support the proofs we calculated background knowledge using three kinds of sources:

- Generic axioms for, for instance, the semantics of possessives, active-passives, and locations.

- Lexical knowledge that was created on the fly with an algorithm that takes as input the DRSs for the text and hypothesis, and outputs first-order axioms based on WordNet hypernyms. This algorithm also performs simple word sense disambiguation and analysis of complex concepts.

- Geographical knowledge from the CIA factbook was translated into first-order axioms.

To perform the actual search for a proof, the DRSs for T and H were translated into first-order logic. The theorem prover and model builder were used in all tasks as complementary inference engines, where the theorem prover attempts to prove the input, and the model builder tries to find a model for the negation of the input. First we checked whether the background knowledge (BK) was consistent with the text, by giving $\neg(\mathrm{BK} \wedge \mathrm{T})$ to the theorem prover. If there is a proof, indicating that the background knowledge is inconsistent, we proceed with checking for entailment without background knowledge, by giving $(\mathrm{T} \rightarrow \mathrm{H})$ to the theorem prover. Otherwise we attempt to prove $(\mathrm{BK} \wedge \mathrm{T} \rightarrow \mathrm{H})$.

Although in theory the method of finding proofs should work, in practice it does not work that well. This is mostly due to the lack of appropriate background knowledge without which many true entailments cannot be found. To overcome this problem we also used a novel way of measuring approximate entailments, relying on the model sizes computed by the model builder. Using Paradox, we computed the model size of $(\mathrm{BK}\wedge\mathrm{T})$ and that of $(\mathrm{BK}\wedge\mathrm{T}\wedge\mathrm{H})$. The underlying idea was that if the difference of these two numbers is small, it is likely to be an entailment. (In other words, the hypothesis does not introduce any or little new information.)

This deep semantic analysis proposes a number of features to describe the T/H-pairs:

`entailed`	{proof,unknown}
`inconsistent`	{proof,unknown}
`domainsize`	numeric
`domainsizeabsdif`	numeric
`domainsizereldif`	numeric
`modelsize`	numeric
`modelsizeabsdif`	numeric
`modelsizereldif`	numeric
`negation`	{yes,no}
`negationtext`	{yes,no}
`negationhypo`	{yes,no}

The features `entailed` and `inconsistent` have been discussed above. `domainsize` is the value of the domainsize of the model for both T and H, `domainsizeabsdif` is the absolute difference between the domain sizes of T and H, and `domainsizereldif` the difference relative to the model size. The `modelsize` is computed by multiplying the domain size with the number of all positive two-place predicates in the model. The features `negation`, `negationtext`, and `negationhypo` are determined by inspecting the DRSs for the presence of negation operators.

4 Combining the Methods

For the combined run we used all shallow and deep features for training a decision tree on the development set. The tree generated for the development data is displayed below:

```
entailed = proof: TRUE
entailed = unknown
|  negationhypo = yes: FALSE
|  negationhypo = no
|  |  overlap <= 0.161146: FALSE
|  |  overlap > 0.161146
|  |  |  inconsistent = proof: TRUE
|  |  |  inconsistent = unknown
|  |  |  |  domainsize <= 8
|  |  |  |  |  negation = yes: FALSE
|  |  |  |  |  negation = no
|  |  |  |  |  |  domainsize <= 6
|  |  |  |  |  |  |  domainsizeabsdif <= 0: TRUE
|  |  |  |  |  |  |  domainsizeabsdif > 0
|  |  |  |  |  |  |  |  modelsizereldif <= 0.595556: TRUE
|  |  |  |  |  |  |  |  modelsizereldif > 0.595556: FALSE
|  |  |  |  |  |  domainsize > 6: FALSE
|  |  |  |  domainsize > 8: TRUE
```

Note that not all features were used (negation in the text, relative domain size difference, model size, and absolute model size difference were not used).

We did not expect good results, as experiments using cross-validation on the development data yielded around 60% accuracy (depending on the decision tree parameters). However, on the test set, this run performed better than the baseline at the 1% level and slightly better than the shallow feature alone. The actual results on the test set are detailed below.

```
cws:       0.5931
accuracy:  0.5625
precision: 0.5530
recall:    0.6525
f:         0.5986
```

5 Error Analysis

The hybrid system was able to create semantic representations and then search for proofs for 774 of all 800 T/H-pairs in the test data, achieving a coverage of 96.8%. Only 30 proofs were found by the system, of which 23 were annotated as entailments in the gold standard. These include adequately analysed phenomena such as apposition (5x: 760, 929, 995, 1903, 1905), relative clauses (3x: 142, 1060, 1900), coordination and attachment(3x: 898, 807, 893), active-passive alternation (2x: 1007, 1897), possessives (1x: 1010), the use of background knowledge (6x: 236, 836, 1944, 1952, 1987, 1994) and more or less straightforward cases (3x: 833, 1076, 741). Note that two examples are included that were annotated as entailment, but strictly speaking they are not (Examples 893 and 236, see also Section 6).

Incorrect proofs were found for seven cases. Some of these are due to the lexical semantics of certain linguistic categories, others to a lack of background knowledge. As an example, the current system does not deal adequately with ordinals and thus finds proofs for 1617 (see below) and 2040.

Example: 1617

T: In 1782 Martin Van Buren, the first US president who was a native citizen of the United States, was born in Kinderhook, N.Y.

H: The first US president was born in Kinderhook, N.Y.

It also found a proof for 2025, where the text contained the hypothesis in an if-clause. Again this was due to an incorrect lexical semantics, and is easy to fix. More complex cases involving modifiers were 2030 and 2082 (see below). It is hard to see what kind of background knowledge can preclude proofs for such cases. (For 2030, the knowledge that Paris is the capital of France, and that each country has at most one capital, would suffice. Unfortunately our system does not select this as background knowledge because the trigger Paris is mentioned neither in the text nor in the hypothesis.)

Example: 2030

T: Lyon is actually the gastronomic capital of France.

H: Lyon is the capital of France.

Example: 2082

T: Microsoft was established in Italy in 1985.

H: Microsoft was established in 1985.

For 2055, the system correctly associated Einstein to be the subject of being the president of Israel, but it incorrectly assumed that begin invited to X is being X. A restriction on this class of modal verbs could fix this problem. (In the development data, however, there were similar cases that were annotated as entailments.)

Example: 2055

T: The fact that Einstein was invited to be the president of Israel is critical to an accurate understanding of one of the greatest individuals in modern history.

H: Einstein is the president of Israel.

Finally, background knowledge that if X is in Y, then X is located in Y, wrongly predicted an entailment for 2079. A more sophisticated lexical analysis of prepositions could improve on such examples.

Example: 2079

T: US presence puts Qatar in a delicate spot.

H: Qatar is located in a delicate spot.

In sum, the backbone of the deep semantic analysis, trying to find proofs, has a small coverage, but is reasonably accurate. Selecting more appropriate background knowledge and revising some of the lexical semantics will improve its precision. We already improved its recall by incorporating the features concerning model size differences.

6 Discussion of the entailment task

We will now discuss some observations we made on the task definition and the annotated data sets.

Task definition The current RTE dataset classified entailment as binary TRUE and FALSE. Following FRACAS, the semantic test suite in (Coopper et al., 1996), a classification that respects three values (yes, don't know, inconsistent), is probably more in its place. For instance, not only are examples 1301 and 1310 below not entailments, the hypotheses are inconsistent with the corresponding texts as well:

Example: 1301

T: The former wife of the South African president did not ask for amnesty, and her activities were not listed in the political reports submitted by the African National Congress to the Truth and Reconciliation Commission in 1996 and 1997.

H: Winny Mandela, the President's ex-wife, is requesting amnesty.

Example: 1310

T: Although the hospital insists that King Hussein is not fully free of the cancer, they are hopeful that he will recover.

H: The statement added that King Hussein has been cured completely.

In the current RTE task definition FALSE subsumes both the "don't know" and "inconsistent" values used in the FRACAS test suite.

Annotated datasets We found several cases where entailments were incorrectly annotated in our opinion. Example 236 (see below), for instance, was judged as entailment. But taking tense into account (which, incidentally, our system is currently not able to do), it is strictly speaking not a textual entailment.

Example: 236

T: Yasir Arafat has agreed to appoint a longtime loyalist as interior minister to take charge of the country's security.

H: Yasir Arafat nominated a loyalist as interior minister.

Another example is 893: the adverb *perhaps* in the text clearly expresses doubt on the date of establishment of settlements on Jakarta, and the hypothesis establishes it as a fact. This clearly is not entailment.

Example: 893

T: The first settlements on the site of Jakarta were established at the mouth of the Ciliwung, perhaps as early as the 5th century AD.

H: The first settlements on the site of Jakarta were established as early as the 5th century AD.

It would also be helpful if human agreement figures and explicit guidelines for annotation could be released for the task. For a small test, one of the authors annotated all 800 examples of the test set for entailment, using the short rules that were indicated on the entailment web page (for example, disregarding tense). Comparing to the final gold standard, now released, we had 38 differences, yielding an agreement of 95.25%. This indicated good agreement, but one has to take into account that both annotations used the indicated simplified guidelines.

References

J. Bos, S. Clark, M. Steedman, J. Curran, and J. Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proc of the 20^{th} International Conference on Computational Linguistics; Geneva, Switzerland; 2004.*

K. Claessen and N. Sörensson. 2003. New techniques that improve mace-style model finding. In *Model Computationa - Principles, Algorithms, Applications (Cade-19 Workshop)*, Miami, Florida.

R. Coopper, R. Cropuch, J. VanEijck, C. Fox, J. Van Genabith, J. Jaspars, H. Kamp, M. Pinkal, D. Milward, M. Poesio, and S. Pulman. 1996. Using the framework. fracas: A framework for computationla semantics. Technical report, Fracas Deliverable D16.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer, Dordrecht, Netherlands.

C. Monz and M. de Rijke. 2003. Light-weight entailment checking for computational semantics. In *Proc. of the 3^{rd} Workshop on Inference in Computational Semantics; 2003.*

A. Riazanov and A. Voronkov. 2002. The design and implementation of Vampire. *AI Communications*, 15(2-3).

H. Saggion, R. Gaizauskas, M. Hepple, I. Roberts, and M Greenwood. 2004. Exploring the performance of boolean retrieval strategies for open domain question answering. In *Proc. of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004.*

Applying COGEX to Recognize Textual Entailment

Abraham Fowler, Bob Hauser, Daniel Hodges, Ian Niles, Adrian Novischi, Jens Stephan
Language Computer Corporation
Richardson, Texas, 75080
{abraham,hauser,daniel,adrian,jens}@languagecomputer.com

Abstract

The PASCAL RTE challenge has helped LCC to explore the applicability of enhancements that have been made to our logic form representation and WordNet lexical chains generator. Our system transforms each T-H pair into logic form representation with semantic relations. The system automatically generates NLP axioms serving as linguistic rewriting rules and lexical chain axioms that connect concepts in the hypothesis and text. A light set of simple hand-coded world knowledge axioms are also included. Our COGEX logic prover is then used to attempt to prove entailment. Semantic relations, WordNet lexical chains, and NLP axioms all helped the logic prover detect entailment.

1 Introduction

Textual entailment occurs when one text can be inferred from the meaning/contents of another text passage. All assertions made in an entailed sentence must be made in the text passage directly, or be logically derivable from it. Our approach attempts to recognize textual entailment by determining if the hypothesis sentence can be logically derived from the text passage using a logic prover. The goal of a logic prover is to determine if some hypothetical statement can be proven given a set of other known true statements. Our logic prover operates by "reductio ad absurdum" or "proof by contradiction" (Wos, 1988). The hypothesis is negated, and if it then contradicts anything in the text or anything inferred from the text, the prover concludes that the original hypothetical statement is derivable from the text (thus, entailment exists).

A description of our system's implementation is provided in Section 2, our results and some performance analysis are included in Section 3, and some final concluding remarks are made in Section 4.

2 System Description

2.1 Logic Form Transformation

In the first stage of our system, the input text and hypothesis are converted into logic forms (Moldovan and Rus, 2001). This conversion process includes part-of-speech tagging, parse tree generation, word sense disambiguation, and semantic relations detection. In the final representation, word senses are removed from the predicates. We found that the inaccuracy of the word sense disambiguator was so great that it prevented many of our other tools from being properly utilized.

2.2 Axiom Generation

We have implemented our COGEX (Moldovan et al., 2003) logic prover into the entailment recognition system. COGEX is a modified version of the OTTER (McCune, 1994) logic prover that has been adapted for natural language processing. The prover requires a list of clauses called the "set of support" which is used to initiate the search for inferences. The set of support is loaded with the negated form of the hypothesis as well as the predicates that make up the text passage. A second list, called the usable list, contains clauses used by OTTER to generate inferences. In our system the usable list consists of all the axioms that have been generated either automatically or by hand. Axioms in our system are utilized to provide external world knowledge, knowledge of syntactic equivalence between logic form predicates, and lexical knowledge in the form of lexical chains.

2.2.1 World Knowledge Axioms

We incorporate a small common-sense knowledge base of 310 world knowledge axioms, where 80 have been manually designed based on the development set data, and 230 originate from previous projects. Currently, this data set is too small to have a significant impact, but in combination with Lexical Chains, the coverage of these axioms will grow.

2.2.2 NLP Axioms

Our NLP Axioms are linguistic rewriting rules that help break down complex logic structures and express syntactic equivalence. These axioms are automatically generated by the system through logic form and parse tree analysis. Axioms are generated to break down complex nominals and coordinating conjunctions into their components so that other axioms can be applied to the components individually to generate a larger set of inferences. Other axioms help us: (1) establish equivalence between prepositions, (2) establish equivalence between different parts of speech, (3) equate words that have multiple noun forms, and (4) equate substantives within appositions.

2.2.3 WordNet Lexical Chains

WordNet provides links between synsets. Each synset has a set of corresponding predicates for each word in the synonym set. The name of a predicate is formed by synonym word form, its part of speech, and WordNet sense. A predicate can have one or more arguments. The predicates corresponding to noun synsets usually have a single argument and the predicates corresponding to verb synsets have three arguments: event, subject, and object arguments.

A lexical chain is a chain of relations between two synsets. For each relation in the chain, the system generates an axiom using the predicates corresponding to the synsets in the relation. The axiom states that the predicate from the first synset implies the predicate from the second. For example, there is an ENTAILMENT relation between the verbs buy and pay. The system generates the following axiom for this relation:

buy_VB_1(e1,x1,x2) → pay_VB_1(e1,x1,x3)

These axioms help the logic prover infer target concepts from starting concepts when lexical chains are found between the two. Not all WordNet relations are used for generating axioms. The following three classes of relations are used: pure WordNet relations, relations created from WordNet derivational morphology, and relations extracted from WordNet glosses. A detailed description of the system as a whole can be found in (Novischi, 2005) and (Moldovan and Novischi, 2002).

2.3 Logic Prover

Once the set of support and usable lists are complete, the logic prover can begin searching for proofs. The clauses in the set of support list are weighted in the order in which they should be chosen to participate in the search. The negated hypothesis is assigned the largest weight to ensure that it will be the last clause to participate in the search. The logic prover removes the clause with the smallest weight from the set of support, and searches the usable list for new inferences that can be made. Any inferences that are produced are assigned an appropriate weight depending on what axiom they were derived from and appended to the set of support list. The logic prover continues in this fashion until the set of support list is empty. If a refutation is found, then the proof is complete. If a refutation cannot be found, then predicate arguments are relaxed. If argument relaxation fails to produce a refutation, predicates are dropped from the negated hypothesis until a refutation is found. Once a proof by refutation is found, a score for that proof is calculated by starting with an initial perfect score and deducting points for axioms that are utilized in the proof, arguments that are relaxed, and predicates that are dropped.

2.4 Scoring

The score generated by the logic prover is only a measure of the kinds of axioms used in the proof and the significance of the dropped arguments and predicates. T-H pairs with longer sentences can potentially drop more predicates, resulting in lower prover scores. Scores are normalized by first calculating the maximum penalty that can be assessed to a pair by dropping all of the hypothesis' predicates. The penalty assessed by the logic prover is then divided by the maximum drop penalty to determine the normalized score.

Due to the logic prover's relaxation techniques, it is always successful in producing a proof. The determination of whether entailment exists is made by examining the penalties assessed by the logic prover in the process of generating the proof. As more axioms are utilized and more predicates are dropped, it becomes much less likely that entailment exists between a pair. All normalized prover scores that fall below a specified threshold are considered false entailment and all scores that are above the threshold are considered true entailment. An appropriate threshold is calculated by examining the scoring output of the development data set to determine what threshold produces the highest accuracy.

The confidence score for a T-H pair in our system is measured as the distance between the normalized score and the threshold. Normalized scores that are further from the threshold will have a higher confidence score than normalized scores that are closer to the threshold. The difference between the normalized score and the threshold itself is normalized such that the resulting confidence score is a value between zero and one.

3 Performance Evaluation

Our results for the challenge are summarized in Table 1. As evidenced by these results, our system performs significantly better on T-H pairs in the comparable documents task. Due to the way T-H pairs are chosen in this task, there is often little to no information in the text of false pairs that could help us logically infer the hypothesis. This inferencing inability causes the logic prover to drop a large number of predicates and return extremely low scores for the false entailment pairs. The large difference between the true and false entailment scores allows us to easily separate the pairs.

The average scores for true and false entailment varied significantly over all of the tasks. This large variance makes it extremely difficult to choose a single threshold that can be used to detect entailment for all of the tasks. By selecting thresholds specific to each task, we were able to increase the test set's accuracy to .562. This accuracy is still considerably lower than the accuracy we received on the development set from which the thresholds were chosen.

Task	Accuracy	CWS	F-measure
test-IR	.478	.386	.472
test-CD	.780	.822	.736
test-RC	.514	.534	.558
test-QA	.485	.434	.481
test-IE	.483	.580	.603
test-MT	.542	.440	.444
test-PP	.450	.450	.585
test-all	.551	.560	.561
dev-all	.630	.639	.619

Table 1: Results for the test and development sets.

The numerous disagreements we had amongst ourselves and with the "Gold Standard" annotations leads us to believe to that the only appropriate way to calculate an upper bound for this task is to utilize the Kappa agreement metric. However, without a large set of different human annotations for the data set, it is impossible to calculate this metric.

Before evaluating the T-H pairs in the test set with our system, we manually determined how difficult it is to prove entailment in each of the true entailment T-H pairs. We established five different difficulty levels: easy, moderate, difficult, intractable, and invalid. Proofs are considered easy in cases where the entailment is simply a matter of eliminating information from the first sentence, recognizing an apposition or replacing one or two words with synonyms. Consider the following example:

Text: *A Union Pacific freight train hit five people.*
Hypothesis: *A Union Pacific freight train struck five people.*
Lexical Chain: *hit_VB* → *strike_VB*

Proofs are considered moderate when one or more inference rules are needed to derive the second sentence of the entailment pair from the first one. Consider the following example:

Text: *Satomi Mitarai died of blood loss.*
Hypothesis: *Satomi Mitarai bled to death.*
World Knowledge Axiom: *die_VB(e1,x1,x2) &*

of_IN(e1,x2) & nn_NNC(x2,x3,x4) & blood_NN(x3) & loss_NN(x4) → bleed_VB(e2,x1,x5) & to_TO(e1, x5) & death_NN(x5)

The expectation is that all entailment pairs that have been deemed easy or moderate can be handled by our current system implementation. Difficult proofs are those that cannot be handled by our theorem prover without adding substantial new functionality (coreference resolution, predicate variables in rules, etc.) or without using ad hoc rules (those not applicable beyond the case which motivates them). The following example requires very specific axioms and coreference resolution:

Text: *Israeli Prime Minister Ariel Sharon threatened to dismiss Cabinet ministers who don't support his plan to withdraw from the Gaza Strip.*

Hypothesis: *Israeli Prime Minister Ariel Sharon threatened to fire cabinet opponents of his Gaza withdrawal plan.*

We have labeled T-H pairs as intractable if we believe that entailment could not be correctly detected by an automated system. Invalid is used to indicate that, in our opinion, an entailment pair which was labeled TRUE should have been labeled FALSE . In the following pair the text does not imply that Silvio Berlusconi is Prime Minister of Italy, only that he is a prime minister with a mandate to reform Italy.

Text: *Prime Minister Silvio Berlusconi was elected March 28 with a mandate to reform Italy's business regulations and pull the economy out of recession.*

Hypothesis: *The Italian Prime Minister is Silvio Berlusconi.*

The system's performance on the T-H pairs classified as easy or moderate is significantly better than its performance on other pairs as illustrated in Table 2. Since many of the T-H pairs with the moderate classification require some external world knowledge, we suspect that with a larger knowledge base, the accuracy of the T-H pairs classified as moderate would be significantly higher.

Difficulty	**Pairs**	**Accuracy**	**CWS**
easy	81	.852	.892
moderate	122	.582	.610
difficult	126	.444	.413
intractable	1	1.000	1.000
invalid	70	.457	.501

Table 2: Results for the true entailment pairs categorized by proof difficulty

It may be possible to build a classifier to determine the inference difficulty and only return results for pairs it deems to be easy or moderate. The main difficulty with such an approach is that it is hard to classify the difficulty of an inference without knowing whether the inference is true or false. We suspect that a difficulty classifier would have trouble distinguishing difficult true entailments from easy false entailments, and vice versa.

4 Conclusion

We participated in the RTE challenge mainly as a learning experience and a test of our existing logic prover system implemented in a new way. Adding semantic relations to the logic form provided deeper semantic connectivity between concepts. This made it possible to write more abstract (more generally-applicable) world knowledge axioms. WordNet lexical chains helped to connect related concepts that used different words or different forms of the same word. And finally, based on linguistic patterns, the NLP axioms helped to link concepts that would otherwise not be connected in the logic form transformation.

References

William W. McCune, 1994. *OTTER Reference Manual and Guide*. Argonne National Laboratory, Illinois, USA, 3.0 edition, January.

Dan I. Moldovan and Adrian Novischi. 2002. Lexical chains for question answering. In *COLING*.

Dan I. Moldovan and Vasile Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Meeting of the Association for Computational Linguistics*, pages 394–401.

Dan I. Moldovan, Christine Clark, Sanda M. Harabagiu, and Steven J. Maiorano. 2003. Cogex: A logic prover for question answering. In *HLT-NAACL*.

Adrian Novischi. 2005. *Semantic Disambiguation of WordNet glosses*. Ph.D. thesis, University of Texas at Dallas.

L. Wos. 1988. *Automated Reasoning - 33 Basic Research Problems*. Prentice-Hall.

Recognizing Textual Entailment Using Lexical Similarity

Valentin Jijkoun and Maarten de Rijke
Informatics Institute, University of Amsterdam
jijkoun,mdr@science.uva.nl

Abstract

We describe our participation in the PASCAL-2005 Recognizing Textual Entailment Challenge. Our method is based on calculating "directed" sentence similarity: checking the directed "semantic" word overlap between the text and the hypothesis. We use frequency-based term weighting in combination with two different lexical similarity measures. Our best run shows 0.55 accuracy on the test data, although the difference between our two runs is not significant. We found remarkably different optimal threshold values for the development and test data. We argue that, in addition to accuracy, precision and recall are valuable measures to consider for textual entailment.

1 Introduction

Recognizing Textual Entailment Challenge, organized within the PASCAL network, is a task where systems are required to detect semantic entailment between pairs of natural language sentences. For example, the sentence *The memorandum noted the United Nations estimated that 2.5 million to 3.5 million people died of AIDS last year* is considered to logically entail the sentence *Over 2 million people died of AIDS last year.*

The organizers of the entailment challenge provided participants with development and test corpora, with 567 and 800 sentence pairs, respectively, manually annotated for logical entailment.

In this paper we describe a simple system based on lexical similarity, with two different word similarity measures. We also present our official results and a deeper analysis of the system's performance.

2 System Description

For every text/hypothesis pair (T, H), we consider each sentence a bag of words and calculate *directed sentence similarity score*. To check for entailment, we compare the score against a threshold. This method is implemented as shown in the pseudo-algorithm below.

```
let $T = (T_1, T_2, \ldots, T_n)$
let $H = (H_1, H_2, \ldots, H_m)$
let $totalSim = 0$
let $totalWeight = 0$
for $j = 1 \ldots m$ do
    let $maxSim = \max_i \mathrm{wordsim}(T_i, H_j)$
    if $maxSim = 0$ then $maxSim = -1$
    $totalSim$ += $maxSim * \mathrm{weight}(H_j)$
    $totalWeight$ += $\mathrm{weight}(H_j)$
end for
let $sim = totalSim / totalWeight$
if $sim \geq threshold$ then return TRUE
return FALSE
```

Essentially, for every word in the hypothesis we find the most similar word in the text according to the measure $\mathrm{wordsim}(w_1, w_2)$. If such a similar word exists (*maxSim* is non-zero), we add the weighted similarity value to the total similarity score. Otherwise, we subtract the weight of the word, penalizing words in the hypothesis without matching words in the text.

The threshold for the final entailment checking is selected using the development corpus of text/hypothesis pairs. The confidence of a system's decision is determined by looking at the distance between the similarity value and the threshold. For example, for positive decisions (*sim* $\geq$ *threshold*):

$$\textit{confidence} = \frac{\textit{sim} - \textit{threshold}}{1 - \textit{threshold}}.$$

The algorithm is parametrized with two functions:

- weight(w): importance of the word for the similarity identification;
- wordsim(w_1, w_2): similarity between two words, with range $[0, 1]$.

2.1 Weighting words

The weighting of words with respect to importance is based on core intuitions from research in Information Retrieval, where Inverse Document Frequency (IDF) is often used as a measure of term importance. Recently, IDF was used for the light-weight entailment checking in (Monz and de Rijke, 2001). For our experiments we used *normalized inverse collection frequency* of words, calculated on a big collection of newspaper texts. For a word w:

$$\text{ICF}(w) = \frac{\text{\# occurences of } w}{\text{\# occurences of all words}},$$

and

$$\text{weight}(w) = 1 - \frac{\text{ICF}(w) - \text{ICF}_{\min}}{\text{ICF}_{\max} - \text{ICF}_{\min}}.$$

The minimum and maximum of the inverse frequencies ($\text{ICF}_{\min}$ and $\text{ICF}_{\max}$) are used to normalize weights between 0 and 1.

2.2 Word similarity measures

We experimented with two similarity measures: Dekang Lin's dependency-based word similarity (Lin, 1998) and the measure based on lexical chains in WordNet (Hirst and St-Onge, 1998). For both measures, words were first converted to lemmas.

3 Results

We submitted two runs that differ in the word similarity measures they use: sim-lin and sim-wn. The table below summarizes the results on the test and development corpora: accuracy (A), confidence-weighted score (CWS), and also precision (P) and recall (R) for the entailment identification.

Run	A	CWS	P	R
Test corpus:				
sim-lin	55.3	55.9	53.7	75.5
sim-wn	53.6	55.3	53.4	56.5
Development corpus:				
sim-lin	61.0	64.9	57.6	81.8
sim-wn	63.4	67.4	61.6	70.6

For our two official runs, sim-lin performed significantly better than random at the 0.01 level, and sim-wn better than random at the 0.05 level.

4 Discussion

The evaluation scores are better on the development corpus than on the test corpus. This is expected since the thresholds were selected on the development corpus. However, a more detailed analysis shows that the differences between the evaluations on the test and development data are not only due to the choice of thresholds. Figure 1 shows how the performance of the system changes when the thresholds are changed from 0.1 to 0.9. We give evaluation results for both our methods and also for a simple baseline that only considers lexical overlap, without WordNet and frequency information.

Surprisingly, the performance of the system on the test corpus (thick lines) is substantially worse than on the development corpus even if optimal similarity thresholds are taken. It is not clear whether this is due to the test corpus being more "difficult," or our system overfits the development corpus in ways other than threshold selection.

Another important observation is that the optimal threshold values differ substantially for different corpora: 0.20–0.4 for the test corpus and 0.6–0.7 for the development corpus. Moreover, whereas the difference between the two similarity measures seems substantial on the development corpus, they perform very similarly on the test corpus. For these reasons, we find it impossible to tell which of the measures is

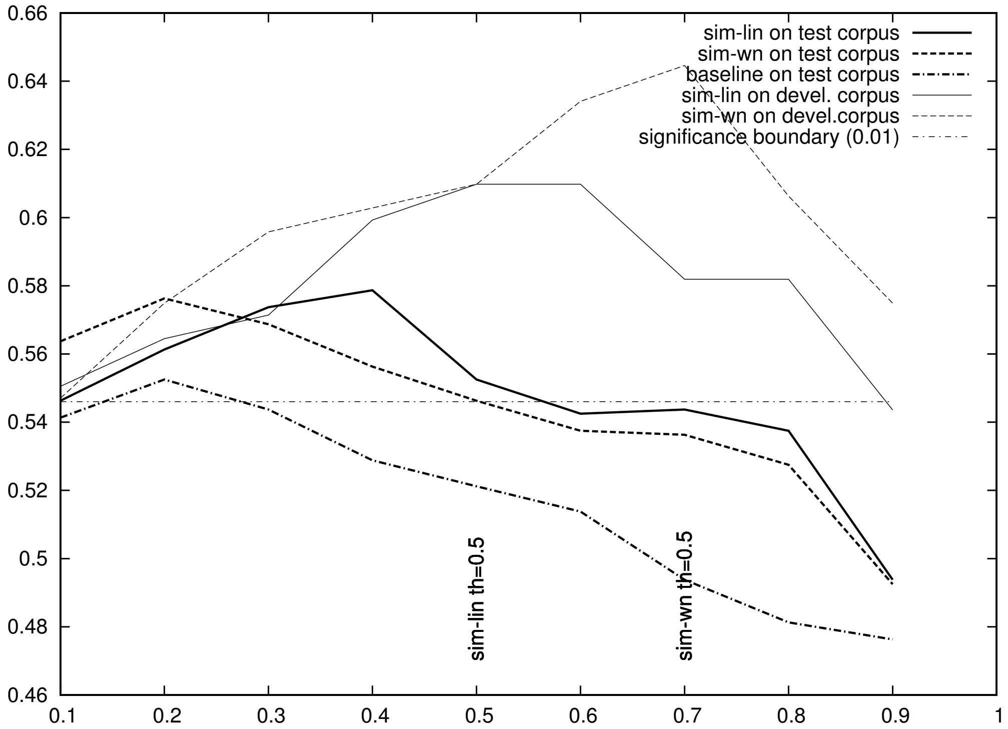


Figure 1: Performance of similarity measures with different thresholds. Thick lines show the performance on the test corpus. The thresholds optimal for the development corpus are clearly not optimal for the test corpus.

better for the task, and how to select thresholds in a robust way.

We also compared the performance of our entailment checking system on different subtasks, corresponding to different sources of the entailment pairs. The table below shows the accuracy, precision and recall for the sim-lin run for all subtasks.[1]

Subtask	A	P	R
CD	84.7	74.7	93.3
IE	55.0	95.0	52.8
MT	46.7	63.3	47.5
QA	42.3	53.9	43.8
RC	49.3	88.6	49.6
PP	42.0	80.0	45.5
IR	53.3	75.6	52.3
Overall	55.3	75.5	53.7

From the table it is clear that the overall accuracy of the system is relatively high only due to the resonable performance on the CD subtask. This particular subtask appears to be quite easy for our system, whereas on other tasks the performance is close to (or worse than) that of the random guessing. Manual examination of the entailment candidate pairs from the CD subtask shows that the pairs usually have many words in common:

T: Voting for a new European Parliament was clouded by concerns over apathy.

H: Voting for a new European Parliament has been clouded by apathy.

Entailment: TRUE, Similarity: 0.88

T: A small bronze bust of Spencer Tracy sold for $174,000.

H: A small bronze bust of Spencer Tracy made $180,447.

Entailment: FALSE, Similarity: 0.44

[1] Recall that the identifiers for the substasks have the following readings: comparable documents (CD), reading comprehension (RC), question answering (QA), information extraction (IE), machine translation (MT), and paraphrase acquisition (PP).

In the second example the similarity is substantially lower since numbers (which occur relatively rarely in our newspaper collection, and thus get higher weight) are different. We have not checked whether a simple word overlap baseline would give a reasonable performance for the CD subtask.

Note that we give precision (P) and recall (R) scores as well as accuracy. We believe that P and R help us to better understand the behavior of our algorithms in ways that accuracy does not. For instance, for all subtasks, except CD, precision is substantially higher than recall. This can be explained by the fact that our lexical similarity resources are far from complete and we are not trying to detect various complex types of paraphrasing (e.g., syntactic). Our method seems very cautious: it prefers to reject the entailment if it cannot find simple lexical evidence to support it. Although, in principle, we can tune the precision/recall balance by varying the thresholds, the experimental results on which we report in this note show that the thresholds are very corpus-specific and thus can hardly be used for this tuning.

5 Conclusions

We described our participation in the PASCAL-2005 Recognizing Textual Entailment Challenge, with a simple sentence similarity-based system that uses two different word similarity measures. Although both our runs show significant improvement over random guessing, the improvement is based only on one subtask (CD). We found that the system cannot be further tuned without overfitting, which suggests that other, deeper text features need to be explored.

Acknowledgments

Both authors were supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. In addition, Maarten de Rijke was supported by grants from NWO, under project numbers 365-20-005, 612.069.006, 220-80-001, 612.000.106, 612.000.207, 612.066.302, 264-70-050, and 017.001.190.

References

Graeme Hirst and David St-Onge. 1998. Lexical chains as representation of context for the detection and correction of malapropisms. In Fellbaum Christiane, editor, *WordNet: An electronic lexical database*. The MIT Press.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*.

Christof Monz and Maarten de Rijke. 2001. Lightweight entailment checking for computational semantics. In *Proceedings of the Workshop on Inference in Computational Semantics (ICoS-3)*.