Light-Weight Entailment Checking for Computational Semantics

Christof Monz Maarten de Rijke

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam, Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands
E-mail: christof,mdr@science.uva.nl

Abstract

Inference tasks in computational semantics have mostly been tackled by means of first-order theorem proving tools. While this is an important and welcome development, it has some inherent limitations. First, generating first-order logic representations of natural language documents is hampered by the lack of efficient and sufficiently robust NLP tools. Second, the computational costs of deploying first-order logic theorem proving tools in real-world situations may be prohibitive. And third, the strict yes/no decisions delivered by such tools are not always appropriate.

In this paper we report on an approach to inference in semantics that works on very minimal representations which can easily be generated for arbitrary domains. Moreover, our approach is computationally efficient, and provides graded outcomes instead of strict yes/no decisions. Our approach is fully implemented, and a preliminary evaluation of the approach is discussed in the paper.

1 Introduction

It has been observed that automated first-order inference systems can serve some of the needs of natural language processing, and, in particular, of discourse processing [Blackburn *et al.*, 1999]. For instance, first-order theorem proving and model generation can be used to implement pragmatic constraints on presupposition projection. The core issues here can be successfully tackled in terms of consistency checks, informativity checks, and minimality checks. Each of these tasks admits a natural interpretation within automated reasoning systems [Gardent and Webber, 2000].

It is safe to say that this emphasis on deploying first-order logic tools for a relatively small number of discourse related tasks characterizes much of today's work in computational semantics [Monz and de Rijke, 2000]. The aim of the present paper is to argue for a much broader view of inference in semantics. First of all,

we want to emphasize that there is a whole spectrum of reasoning tools from which semanticists can choose. These range from theorem proving in first-order logic to satisfiability checking in restricted logics (such as modal and description logics) to various forms of abductive reasoning to model checking in temporal logic to probabilistic reasoning.

Second, there is a wide variety of domains in which such tools can be used to perform semantical tasks, including discourse processing, dialogue processing, and information processing.

Third, the choice of one particular reasoning service over another should be guided by a variety of concerns, including performance, as well as the ability to efficiently and robustly generate representations of text or speech that can be fed to a reasoning service. By and large, computational semanticists seem to have ignored the latter requirement. While some may argue that adequate performance is unfeasible given current techniques, our take on this matter is that semanticists should get to work with, and try to get the most out of, currently available NLP techniques, without waiting, say, for the perfect parser.

The fourth point we want to get across is that, ultimately, the choice of one particular representation and inference mechanism over others should be decided upon by means of proper testing and evaluation techniques.

Our general agenda, then, is to pursue computational semantics using currently available techniques, and to determine not only how far we can get, but also to what extent moving to richer representations and deeper levels of analysis pays off. This is similar to the trade-off between partial and full parsing: although partial parsing is not as reliable as full parsing, it is often used because it is more efficient, robust, and often provides sufficient information. Attempts to use deeper representations should then be motivated by the fact that more shallow ways of building representations are unable to provide the information needed.

To make matters concrete, we focus on *entailment checking* and its use in computational semantics. In Section 2 we briefly discuss areas in which the need for efficient entailment checking arises. In Section 3 we list a number of criteria for selecting a method for checking entailment, and we zoom in on one particular choice, based on ideas from information retrieval. In Sections 4 and 5 we describe experiments that we carried out to provide a preliminary evaluation of our method for entailment checking, and in Section 6 we conclude and outline future work.

2 Background: Informativity

Informativity is about determining whether a piece of information (e.g., a reading or an utterance or a piece of text) is already entailed by its local context. Informativity is an often used notion in natural language processing and understanding. For instance, [Blackburn *et al.*, 1999] show that informativity can be treated as an entailment problem: a piece of new information NEW is informative with respect a discourse context OLD and general world knowledge KB just in case the impli-

Figure 1: Two Segmentized Documents (Topic 6).

Reuters (31 Oct 2000 16:39 GMT)	AP (31 Oct 2000 16:25 GMT)
Reuters 1: TAIPEI (Reuters) - A Singapore Airlines plane bound for Los Angeles crashed during a typhoon at Taiwan's international airport on Tuesday, an airport police official said. **Reuters 2:** It was not immediately known how many of the 159 passengers and 20 crew were killed or injured, Civil Aeronautics Administration deputy director Chang Kuo-cheng told reporters. "The plane burst into flames and exploded shortly after takeoff," an airport police official told reporters. **Reuters 3:** Local television was reporting that over 120 injured had been taken to hospital. **Reuters 4:** The SIA Boeing 747-400 was taking off during a storm and hit by strong winds. It hit two other planes on the tarmac, including a China Airlines plane, police said. A Taiwan vice transport minister said no one was on board the other two planes. **Reuters 5:** The injured were rushed to hospital. No other details were immediately available. ⋮	**AP 1:** TAIPEI, Taiwan (AP) - A Singapore Airlines jetliner bound for Los Angeles crashed on takeoff in a storm Tuesday night and slammed into another plane on the runway, a Taiwanese official said. **AP 2:** There were 179 people on board Singapore Airlines Flight SQ006, which local media reports said was a 747. It was not immediately known how many people were hurt or killed, but local media reports said some injured people were being taken to the hospital. Strong winds seemed to have forced the plane down. There was an explosion as it struck a China Airlines plane on the runway at Taipei's Chiang Kai-shek International Airport, emergency official Wu Bi-chang said. Local media reports said the China Airlines plane was empty. **AP 3:** The crash occurred at 11:18 p.m. local time, and rescue workers were being dispatched to the scene, Wu said. Minutes later, the flashing lights of rescue vehicles were visible on the wet tarmac. Local media reports said there was a fire on the runway after the crash but that it had been extinguished. ⋮

cation $\mathrm{OLD} \wedge \mathrm{KB} \rightarrow \mathrm{NEW}$ is not valid. [Gardent and Webber, 2000] show how informativity may be used in the discourse interpretation of phenomena such as noun-noun compounds, metonymy, and definite noun phrases.

The computation of entailment relations is also important in areas of computational linguistics other than discourse processing. Let's consider two examples. The first is document summarization [Barzilay *et al.*, 1999; Mani and Bloedorn, 1999]. [Radev, 2000] describes 24 cross-document relations that can hold between segments of documents, one of which is the *entailment* relation. It can be used to compute the informativity of one text segment compared to another one. In the context of summarization this is used to avoid redundancy, i.e., if a segment entails another segment, only the entailing segment should be included in the summary.

Figure 1 shows an example of two segmentized documents, both covering a plane crash in Taiwan. A superficial glance reveals some entailments; for instance, **AP 2** is at least as informative as **Reuters 4** and as **Reuters 5**. On the other hand, it seems clear that **AP 2** does not contain the specific information mentioned in **Reuters 3**.

Formally, the task at hand may be formulated as follows: $s_{i,d}$ (the i-th segment

of document d) is *at least as informative as* $s_{j,d'}$ (the j-th segment of document d') if $s_{i,d}$ entails $s_{j,d'}$. Clearly, determining the informativity of a text segment is a task on the interface of computational semantics and inference.

Our final example of the need for entailment checking in NLP concerns concept hierarchies, that is, collections of terms organized in a hierarchical structure, where a concept A is higher in the hierarchy than a concept B if A is more general than B in the sense that all B-instances are A-instances, that is, if B entails A [Donini *et al.*, 1996]. Concept hierarchies have proved useful for a variety of purposes, including retrieval, browsing and navigation. Currently, the most common forms of concept hierarchies are the well-known categorization schemes such as Yahoo [Yahoo, 2001], and the WordNet thesaurus [WordNet, 2001], an organization of terms with synonym, antonym, hyponym/hypernym (is-a/is-a-type-of), and meronym/holonym (has-part/is-part-of) relations.

Most of these hierarchies are hand-coded, and, usually, a pair of concepts is included in the hierarchy only if the one entails the other in the traditional, strict sense of the word. There has been some work on automatically deriving thesaural relationships from texts. In recent work, Sanderson and Croft [Sanderson and Croft, 1999] report on the use of basic information retrieval techniques for this purpose; the hierarchical relation educed from a set of documents is certainly not strict logical entailment, but something like 'the child concept is a related subtopic of the parent concept.'

3 Entailment Checking

As we have just seen, there is a variety of areas in computational linguistics where entailment checking is an essential inference task. But what kind of entailment checking is appropriate? And what kind of algorithms should we use?

3.1 Three Criteria

There is a wide spectrum of methods for entailment checking that can — in principle — be used. In deciding which method to use, the following criteria are among the main ones:

1. the robustness and coverage of methods for generating representations that our system can work on,
2. the computational costs (and behavior) of the entailment checks, and
3. the type of outcomes or output that we want to have.

Below we discuss these criteria in a bit more detail; after that we present our own approach to entailment checking.

Generating Representations. Obviously, whichever tool we use for entailment checking, it needs to operate on *representations* of the input documents. Semanticists have a tendency to opt for rich representation formalisms so as to be able to capture as many relevant aspects as possible. In practice, 'rich' often means 'includes (at least) first-order logic.' What does it take to generate first-order logic representations of documents? Traditionally, this is taken to involve a number of levels, including syntactic structure, logical form, and some form of contextual interpretation [Allen, 1995]. Despite important recent advances, relevant tools (such as parsers) lack the robustness and coverage needed to efficiently generate deep semantic representations of arbitrary natural language documents. Moreover, the traditional demand that representations be precise and unambiguous may lead to representations whose size is exponential (or worse) in the size of the input document. Indeed, practicable methods for generating first-order representations are very rare.

An obvious way out is to turn to more light-weight representations that can be obtained by more shallow and more robust NLP techniques. Partial parsing or chunk parsing can be used to build such representations in an efficient and robust manner, while avoiding full disambiguation [Abney, 1996; Hobbs et al., 1996].

Bags of (stemmed) words, possibly filtered through a stopword list, are even more shallow representations of natural language documents. At the cost of giving up virtually all syntactic structure, bag-of-words representations provide very concise representations that are easy to generate, and that are typically used for large text collections [Baeza-Yates and Ribeiro-Neto, 1999]. The method for entailment checking that we propose below uses bag-of-words representations.

Computational Costs. How hard is it to reason with pieces of information in a given representation format? If one opts for first-order logic as one's representation formalism, the entailment problem is obviously undecidable. Admittedly, recent advances in first-order theorem provers and model generators do seem to make them of practical use in some classes of linguistic problems [Blackburn *et al.*, 1999]. And there is the hope that the development of test suites for, e.g., discourse understanding will allow the tuning of automated inference systems to excel at strategies that support efficient inference for semantics [Gardent and Webber, 2000]. Nevertheless, we are dealing with an undecidable problem; in practice this means that minor variations in do-able instances may cause time outs and exploding memory usage.

We see two possible replies to this problem. One is to ignore it and to accept the fact that there are problem instances for which the available computational resources are guaranteed not to suffice. The other is to stick to wide coverage but to simplify the reasoning task so that it is guaranteed to behave well on each problem instance; the latter is the approach that we adopt below.

Type of Outcomes. There is another fundamental issue here. Most reasoning methods currently used or proposed in computational semantics are based on strict, binary logical reasoning, and if they produce an outcome at all, it will be a strict yes or no. As any textbook on AI will explain, there are many reasons why approximate reasoning should be preferred in some cases. For the purposes of entailment checking in computational semantics, two reasons are particularly relevant. First, in many practical situations, such as document summarization, we simply are content with *approximate* answers, and prefer approximate answers to having no answer at all.

Second, we have found that, usually, the strict binary entailment relation only holds between text segments $s_{i,d}$ and $s_{j,d'}$ (in that order) whenever $s_{i,d}$ is a copy of $s_{j,d'}$ or an extension of a copy of $s_{j,d'}$. In other words, if only Boolean answers are allowed, the entailment relation may be too sparse (i.e., hold between too few pairs of text segments) to be of any practical use.

3.2 Our Approach

We propose a simple yet effective method for entailment checking that is based on a familiar similarity measure from information retrieval. Here are the basic ideas. First of all, we represent text segments as bags of (weighted) words. Next, to explain how weights are computed, we need to introduce a few notions. By a *topic* we mean a set of related documents; these are documents for which we need to compute entailment relations. Further, to define the weights, we use N to denote the total number of segments in the topic, and n_i for the number of segments in which the term t_i occurs. Then, the weight of a term t_i within a given topic as assigned by the equation

$$idf_i = \log\left(\frac{N}{n_i}\right) \tag{1}$$

is known as its *inverse document frequency* in an information retrieval setting [Baeza-Yates and Ribeiro-Neto, 1999]. Terms that occur in many segments (i.e., for which n_i is rather large), such as *the*, *some*, etc., receive a lower *idf*-score than terms that occur only in a few segments. The intuition behind the *idf*-score is that terms with a higher *idf*-score are better suited for discriminating the content of a particular segment from the other segments in the topic, or to put it differently, they are more content-bearing. Note, that the logarithm in (1) is only used to smoothen the differences between the scores.

Let d, d' be two documents. Given the term weights as defined in (1), we compute the *entailment score*, $entscore(s_{i,d}, s_{j,d'})$, of two segments $s_{i,d}$ in d and $s_{j,d'}$ in d' by comparing the sum of the weights of terms that appear in both segments to the sum of the weights of all terms in $s_{j,d'}$:

$$entscore(s_{i,d}, s_{j,d'}) = \frac{\sum_{t_k \in (s_{i,d} \cap s_{j,d'})} idf_k}{\sum_{t_k \in s_{j,d'}} idf_k}. \tag{2}$$

In words: how many of the content-bearing terms in $s_{j,d'}$ occur in $s_{i,d}$? Clearly, $entscore(s_{i,d}, s_{j,d'})$ varies from 0 to 1, thus providing the possibility of approximate entailment judgments.

A few remarks are in order. First, note that our entailment score is not just a notion of similarity: in general, $entscore(s_{i,d}, s_{j,d'}) \neq entscore(s_{j,d'}, s_{i,d})$.

Second, to work with $entscore$ and conclude that $s_{i,d}$ entails $s_{j,d'}$, it may not be sufficient to have a non-zero entailment score: we may need some positive 'entailment threshold.' For an example that illustrates this point, consider Figure 1 again. As we pointed out earlier, segment **AP 2** entails **Reuters 4** and **Reuters 5**. The entailment scores obtained using (2) are as follows: $entscore($**AP 2**, **Reuters 4**$) \approx 0.52$, while $entscore($**AP 2**, **Reuters 5**$) \approx 0.28$, suggesting an obvious upper-bound for the entailment threshold. In contrast, **Reuters 4** $\rightarrow$ **AP 2** does not seem to be a valid implication, suggesting that the entailment threshold should be larger than $entscore($**Reuters 4**, **AP 2**$) \approx 0.08$. The mechanism of entailment thresholds offers a large amount of flexibility for fine-tuning the entailment relation to one's purposes. See Section 5 for further discussions on this point.

Third, what kind of *logical* properties does the entailment relation that is computed using (2) enjoy? While it satisfies some properties that we usually assign to entailment relations, such as reflexivity (i.e., $entscore(s_{i,d}, s_{i,d}) = 1$ for any segment $s_{i,d}$), it also fails to satisfy some, such as transitivity. For instance, Topic 6 in our test collection contains the AP and Reuters document displayed in Figure 1, as well as a document from CNN. We found $entscore($**AP 1**, **Reuters 2**$) \approx 0.63$ and $entscore($**Reuters 2**, **CNN 5**$) \approx 0.32$, so with an entailment threshold of, say, 0.27, we would get a 'yes' for both implications, but we would get a 'no' for the implication **AP 1** $\rightarrow$ **CNN 5** as we found $entscore($**AP 1**, **CNN 5**$) \approx 0.16$, well below the subsumption threshold. Unfortunately, further explorations of the purely properties of the entailment relations computed by (2) are beyond the scope of this paper. Instead, our next task is to evaluate our approach.

4 Experimental Set-Up

In this section we describe our experimental set-up, including our test collection as well as some figures concerning the generation of representations and the computations of the entailment scores. Details of our preliminary evaluation are described in Section 5 below.

4.1 Data Source and Preparation

For our experiments we prepared a small corpus consisting of 69 news stories from the AP news wire, BBC, CNN, *L.A. Times*, Reuters, *USA Today*, *Washington Post*, and *Washington Times*. The collection was categorized into 21 topics; this was done by hand. All documents belonging to a single topic were released on the same day and describe the same event; see Figure 1 for two documents from Topic 6.

Table 1: Statistics on the Test Collection (21 topics, 69 documents).		
	average per topic	
number of documents	3.3	docs.
document length	612	words
total length of documents	2115	words
length of longest document	783	words
length of shortest document	444	words
segments per document	16.4	
total number of segments	55.9	

The documents were segmented into paragraphs; in news stories these tend to be short, and we found that they rarely exceed 4 sentences. On average a document consisted of 16.4 segments, and a topic of a total of 55.9 segments; see Table 1.

4.2 Generating Representations

In a bag-of-words approach, the generation of representations is rather trivial, involving three fairly simple steps. First, the input is tokenized, where word boundaries are recognized and punctuation is removed. Then, each word (or token) is normalized to its lemma, where morphological information, such as inflection and plural indicators are removed. For tokenization and lemmatization we use TreeTagger [Schmid, 1994], a decision-tree-based part-of-speech tagger. Although TreeTagger assigns part-of-speech information to each word, this information is not used for further processing in the current system. The last step is to assign an *idf*-score to each word within a topic, where the *idf*-score is computed as described in Section 3.2.

4.3 Computing Entailment Scores

Now, to be able to carry out entailment checks, we need to compute entailment scores using $entscore$. Given two documents d, d' belonging to the same topic, we compute the entailment score for each pair of segments $(s_{i,d}, s_{j,d'})$. Although the pairwise computation of $entscore$ is exponential in the number of documents in the topic, it still remains computationally tractable in practice. For instance, for 4 documents (slightly more than the average case), with a total of about 1600 pairs of text segments, it takes under 10 seconds to compute all entailment scores. For 8 documents (an extreme case, which did not occur in the current collection but was artificially constructed) it takes 66 seconds; both times were measured on a 600 MHz Pentium III PC.

5 Evaluation

The literature on logic and semantics is full of examples of ideas that look interesting on paper, but perform poorly in practice. Theoretical studies do not provide an indication of the effectiveness of information processing algorithms such as the entailment scoring method that we have proposed. Hence, empirical testing is called for. Below, we first discuss the method and measures that we have used, and then we present our testing results.

5.1 Method

We compared automatically generated entailment relations to 'ideal' entailment relations. For each of the 21 topics in our test corpus we randomly selected two documents in the topic, and asked a human subject to determine all entailment relations between segments in different documents (within the same topic). Judgments were made on a scale 0–2, according to the extent to which one segment was found to entail another. Put differently, if a segment $s_{i,d}$ was found to entail segment $s_{j,d'}$ the pair ($s_{i,d}$, $s_{j,d'}$) would be rated 2, while it would be rated 0 if no entailment was found. Thus, the human judge was allowed a spectrum of entailment in the ratings 0, 1, 2.

Let's look at some examples to illustrate these ratings; in Topic 6, the AP and Reuters documents listed in Figure 1 were selected for human assessment.

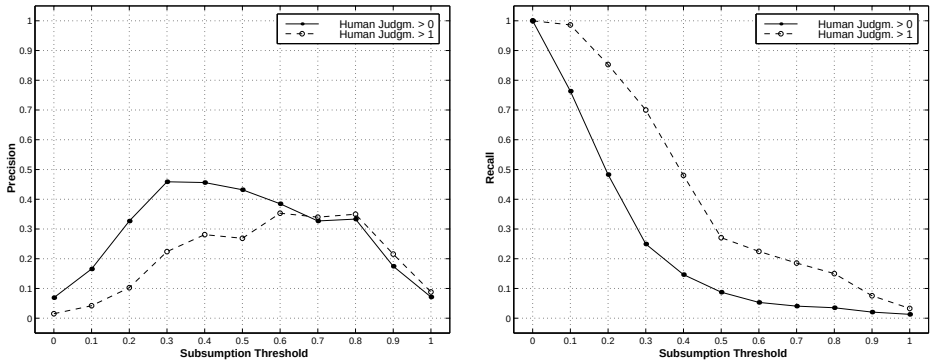
Score 2: In Topic 6, the implications **AP 2** $\rightarrow$ **Reuters 4** and **AP 2** $\rightarrow$ **Reuters 5** both scored 2, because all information in the segments on the right-hand side is present — either implicitly or explicitly — in the segment on the left-hand side.

Score 1: An implication $s_{i,d} \rightarrow s_{j,d'}$ was rated 1 whenever $s_{i,d}$ entailed a substantial subsegment of $s_{j,d'}$. For instance, **AP 1** (in Topic 6) says nearly everything expressed by **Reuters 4** except for the type of the aircraft and the fact that it was empty.

Score 0: An example where no entailment is found is given by **Reuters 4** and **AP 2**: **AP 2** contains a significant amount of information that is not present in **Reuters 4**. In such cases a score 0 was to be assigned.

Let a *potential entailment pair* be an ordered pair of text segments $(s_{i,d}, s_{j,d'})$ that may or may not stand in the entailment relation. Out of the 12083 potential entailment pairs that our human judge had to consider, 501 (4.15%) received a score of 1, and only 89 (0.73%) received a score of 2. All other potential entailment pairs received a score equal to 0.

How can we use these human assessments for measuring the performance of our entailment checking method? Let a *correct* entailment pair be a potential entailment pair $(s_{i,d}, s_{j,d'})$ for which $s_{i,d}$ does indeed entail $s_{j,d'}$ according to our human judge. Further, a *computed* entailment pair is a potential entailment pair for



(a) (b)

Figure 2: (a) Average precision with human judgments > 0 and > 1. (b) Average recall with human judgments > 0 and > 1.

which our entailment method has produced a score above the entailment threshold. *Precision* is a useful measure for determining the accuracy of our entailment checking method. It is defined as the fraction of computed entailment pairs that is correct:

$$\text{Precision} = \frac{\text{number of correct entailment pairs computed}}{\text{total number of entailment pairs computed}}.$$

Recall is used to measure the extent to which our entailment checking method is exhaustive. It is defined as the proportion of the total number of correct entailment pairs that were computed:

$$\text{Recall} = \frac{\text{number of correct entailment pairs computed}}{\text{total number of correct entailment pairs}}.$$

Observe that precision and recall depend on the entailment threshold that we use. For instance, with a very low entailment threshold, we can expect a larger number of computed entailment pairs, and hence a larger number of correctly computed entailment pairs; that is, with a low entailment threshold recall will increase.

5.2 Results

Using human judgments for the two selected documents per topic, we computed average recall and precision at 11 different entailment thresholds, ranging from 0 to 1, with .1 increments; the average was computed over all topics. The results are summarized in Figures 2 (a) and (b).

There were two ways in which we compared the computed entailment scores against the human assessments. First, we used a human rating of more than 0 to classify an entailment pair as correct; the resulting precision and recall are indicated with 'Human judgment > 0' in Figures 2 (a) and (b). Second, we required

a human rating of more than 1 for a entailment pair to be correct; the resulting measures are indicated with 'Human judgment > 1.'

There are several things worth noting about our experimental results. First, as expected, precision is higher when human judgments > 0 are used to determine the correct entailment pairs than with human judgments > 1. Further, the highest score is obtained for a entailment threshold around 0.3, thus suggesting that some value around 0.3 is the optimal entailment threshold. As expected, initially precision increases as the entailment threshold is increased, but there are drops in precision around 0.3 and again around 0.8. This may be explained as follows. For some topics, the threshold is higher than the maximum entailment score, and in such cases no entailment pairs are computed and precision for those topics drops to 0.

As to recall (Figure 2 (b)), as expected recall is higher when human judgments > 1 are used to determine the correct entailment pairs than with human judgments > 0. Moreover, recall increases as the entailment threshold decreases, thus suggesting that a entailment threshold equal to 0 is the preferred one if recall is the most important measure.

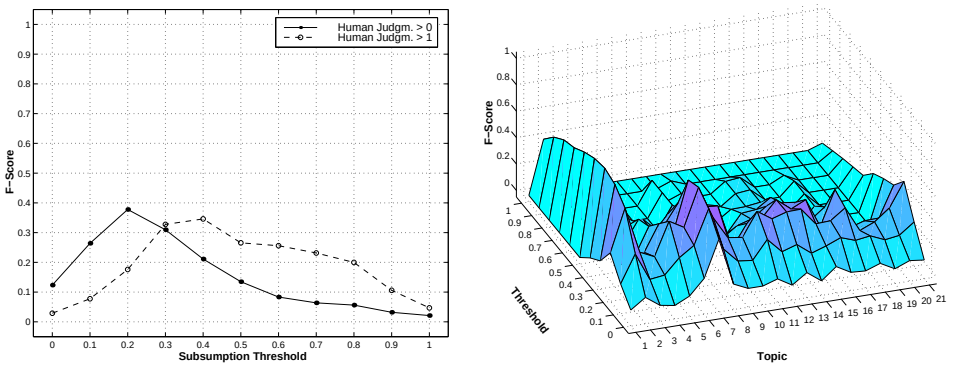
Since precision and recall suggest two different optimal entailment thresholds, there is an obvious question: what is the optimal threshold if precision and recall are equally important? In information retrieval, various methods have been suggested for generating a single performance number, which combines precision and recall aspects, to quantify the usefulness of retrieval methods. One of these is the harmonic mean F of recall and precision [Shaw Jr *et al.*, 1997] which is computed as

$$F = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}.$$

The F-score assumes a high value only when both recall and precision are high. We have plotted the average F-scores in Figure 3 (a). Observe that the optimal entailment threshold for human judgments > 0 seems to be around 0.18, and approximately 0.4 for human judgments > 1. This conforms to the intuition that a higher entailment threshold is more effective when human judgments are stricter.

In Figure 3 (b) we have plotted F-scores per topic, with human judgments > 0. The average over these curves corresponds to the solid line in (a), and, indeed, the shape of the solid line in (a) can be recognized in (b). Note that there is some variance between the topics, and that there are some clear outliers, such as Topic 1 and Topic 2.

The F-score suggests that 0.2 (0.4) is to be taken as entailment threshold for identifying entailing segments with a human judgment score of at least 1 (2). This results in an overall precision of 0.33 (0.28) and an overall recall of 0.48 (0.48). Obviously, precision and recall figures in the 30% and 40% range are not optimal, but computing entailment is a hard task, and relatively poor results should not come as a surprise. Nevertheless, even when working with extremely simple representations, our experiments indicate that almost half of all entailment relations are identified! It might appear that identifying only a third of them correctly is unsatisfactory, but, again, note that entailment relations are very sparse, viz. only



(a) (b)

Figure 3: (a) Average F-scores with human judgments > 0 and > 1. (b) F-scores across topics, with human judgments > 0.

4% of all potential entailment pairs received a human rating of 1, and even fewer, 0.7%, received a human rating of 2.

At the end of the day, we're left with the following question: Did we really compute *entailment* between text segments, or merely some kind of topic overlap? It is generally accepted that idf can assess when two text segments concern a similar subject matter. In many cases, and especially for newswires, it is to expected that similar subject matter implies alternative stories about the same events, and, therefore, that idf will be a reasonable measure of entailment. This measure may or may not suffice, depending on the task at hand — the important thing is that our method and the human judgments on which its preliminary evaluation are based, give us a baseline against which other methods can be compared.

6 Conclusion

In this paper we have argued for a very liberal view of inference for computational semantics. We have also argued that the costs of building representations should be a key concern when deciding which inference method to use for semantical tasks. We have also argued that, instead of waiting for perfect tools to arrive and deliver perfect representations, computational semanticists should try to get the most out of currently available NLP techniques.

To illustrate these ideas we proposed a simple yet effective method for entailment checking that is based on the use of idf. While the use of idf is nothing new, we believe that our scoring function and its application to measure entailments are. Our approach is computationally efficient, and provides graded outcomes instead of strict yes/no decisions. Moreover, the approach is fully implemented.

Another message of this paper is the need and importance of empirical evaluation in computational semantics. We provided a preliminary evaluation of our entailment checking method using a small corpus of 69 news stories, organized in 21 topics.

Our future work will be pursued along two main lines: enhancements to our entailment checking method, and improvements to our evaluation.

Enhancing the Method. While human judgments require a substantial effort, we believe that it is essential to have a reasonably sized test corpus so as to be able to successfully pursue our main interest, of which the present paper is only a first step: to generate various kinds of representations of natural language documents, to perform inference tasks with these representations, and to determine, by empirical means, how representation and inference are connected.

The next step along this line is to improve and extend the current entailment scoring method, first of all by using lexical semantic information in the form of WordNet synonyms and hyponyms/hypernyms. Then, we want to move to a richer level of representation such as simple argument structures, and to perform entailment checks at that level and compare the resulting precision and recall measures to those obtained in the present paper. As suggested by one of our referees, the substantial literature on text classification may provide useful input here, in particular Bayesian methods [Koller and Sahami, 1996] and pattern-recognition methods based on optimization [Joachims, 1999]. In addition, a number of ideas related to the maximal marginal relevance criterion known from document summarization [Carbonell and Goldstein, 1998] seem relevant.

Evaluation. To improve the quality of our evaluation, we have extended the size of our collection to 30 topics, and we are in the process of extending the current judgments to include the new topics, while a second human judge is currently working on scoring the extended collection. Using these additional judgments, we aim to characterize the distribution of results within and across topics.

Acknowledgments

We want to thank our referees for valuable comments and suggestions. We also want to thank Henry Chinaski for providing the human judgments on over 12.000 possible entailments.

Christof Monz was supported by the Physical Sciences Council with financial support from the Netherlands Organization for Scientific Research (NWO), project 612-13-001. Maarten de Rijke was supported by the Spinoza Project 'Logic in Action' at ILLC, the University of Amsterdam, and by a grant from NWO under project number 365-20-005.

References

[Abney, 1996] S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.

[Allen, 1995] J. Allen. *Natural Language Understanding*. Benjamin Cummings, 1995.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[Barzilay *et al.*, 1999] R. Barzilay, K. McKeown, and M. Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL'99)*, 1999.

[Blackburn *et al.*, 1999] P. Blackburn, J. Bos, M. Kohlhase, and H. de Nivelle. Inference and computational semantics. In *Proceedings IWCS-3*, 1999.

[Carbonell and Goldstein, 1998] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings SIGIR'98*, pages 335–336, 1998.

[Donini *et al.*, 1996] F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Reasoning in description logics. In *Principles of Knowledge Representation*. CSLI Publications, 1996.

[Gardent and Webber, 2000] C. Gardent and B. Webber. Automated reasoning and discourse interpretation. Claus report 113, Computational Linguistics, University of the Saarland, 2000.

[Hobbs et al., 1996] J. Hobbs et al. FASTUS: a cascaded finite-state transducer for extracting information from natural-language text. In E. Roche and Y. Schabes, editors, *Finite State Devices for Natural Language Processing*. Cambridge MA: MIT Press, 1996.

[Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*. Morgan Kaufmann Publishers, 1999.

[Koller and Sahami, 1996] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of ICML-96, 13th International Conference on Machine Learning*, pages 284–292, 1996.

[Mani and Bloedorn, 1999] I. Mani and E. Bloedorn. Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1–2):35–67, 1999.

[Monz and de Rijke, 2000] C. Monz and M. de Rijke. Inference in computational semantics. *Journal of Language and Computation*, 1:151–158, 2000.

[Radev, 2000] D. Radev. A common theory of information theory from multiple text sources, step one: Cross-document structure. In *Proceedings 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, 2000.

[Sanderson and Croft, 1999] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR'99*, pages 206–213, 1999.

[Schmid, 1994] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[Shaw Jr *et al.*, 1997] W.M. Shaw Jr, R. Burgin, and P. Howell. Performance standards and evaluations in IR test collections: Cluster-based retrieval methods. *Information Processing & Management*, 33:1–14, 1997.

[WordNet, 2001] 2001. WordNet. URL: `http://www.cogsci.princeton.edu/~wn`. Accessed May 17, 2001.

[Yahoo, 2001] 2001. Yahoo. URL: `http://www.yahoo.com`. Accessed February 23, 2001.