Semantic Inference at the Lexical-Syntactic Level for Textual Entailment Recognition

Roy Bar-Haim†,**Ido Dagan**†, **Iddo Greental**‡, **Idan Szpektor**† and **Moshe Friedman**†
†Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel
‡Linguistics Department, Tel Aviv University, Ramat Aviv 69978, Israel
{barhair,dagan}@cs.biu.ac.il,greenta@post.tau.ac.il,
{szpekti,friedmm}@cs.biu.ac.il

Abstract

We present a new framework for textual entailment, which provides a modular integration between knowledge-based exact inference and cost-based approximate matching. Diverse types of knowledge are uniformly represented as entailment rules, which were acquired both manually and automatically. Our proof system operates directly on parse trees, and infers new trees by applying entailment rules, aiming to strictly *generate* the target *hypothesis* from the source *text*. In order to cope with inevitable knowledge gaps, a cost function is used to measure the remaining "distance" from the hypothesis.

1 Introduction

According to the traditional formal semantics approach, inference is conducted at the logical level. However, practical text understanding systems usually employ shallower lexical and lexical-syntactic representations, augmented with partial semantic annotations. Such practices are typically partial and quite ad-hoc, and lack a clear formalism that specifies how inference knowledge should be represented and applied. The current paper proposes a step towards filling this gap, by defining a principled semantic inference mechanism over parse-based representations.

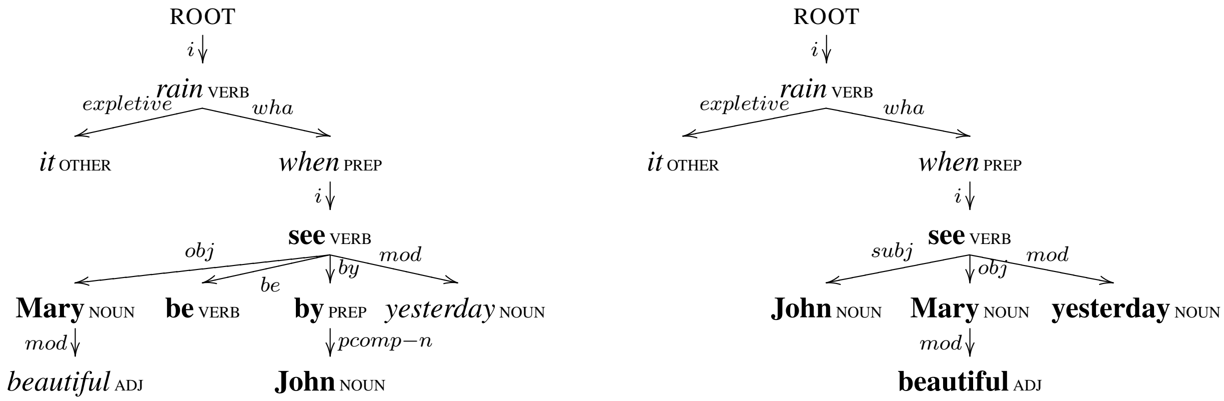
Within the textual entailment setting a system is required to recognize whether a hypothesized statement h can be inferred from an asserted text t. Some inferences can be based on available knowledge, such as information about synonyms and paraphrases. However, some gaps usually arise and it is often not possible to derive a complete "proof" based on available inference knowledge. Such situations are typically handled through approximate matching methods.

This paper focuses on knowledge-based inference, while employing rather basic methods for approximate matching. We define a proof system that operates over syntactic parse trees. New trees are derived using entailment rules, which provide a principled and uniform mechanism for incorporating a wide variety of manually and automatically-acquired inference knowledge. Interpretation into stipulated semantic representations, which is often difficult to obtain, is circumvented altogether. Our research goal is to explore how far we can get with such an inference approach, and identify the scope in which semantic interpretation may not be needed. For a detailed discussion of our approach and related work, see (Bar-Haim et al., 2007).

2 Inference Framework

The main contribution of the current work is a principled semantic inference mechanism, that aims to generate a target text from a source text using entailment rules, analogously to logic-based proof systems. Given two parsed text fragments, termed *text* (t) and *hypothesis* (h), the inference system (or *prover*) determines whether t entails h. The prover applies entailment rules that aim to transform t into h through a sequence of intermediate parse trees. For each generated tree p, a heuristic cost function is employed to measure the likelihood of p entailing h.

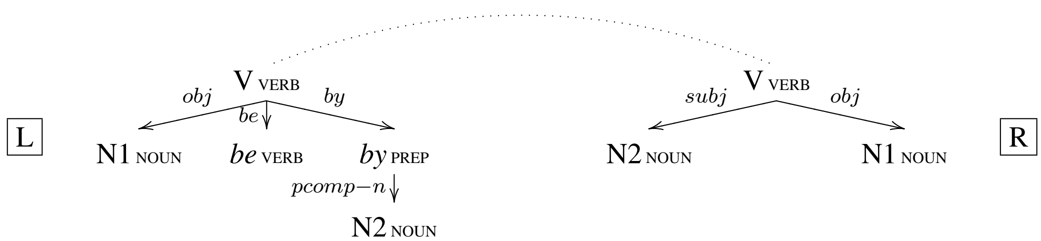
Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 131–136,
Prague, June 2007.



Source: it rained when beautiful Mary was seen by John yesterday

Derived: it rained when John saw beautiful Mary yesterday

(a) Application of passive to active transformation



(b) Passive to active transformation (substitution rule). The dotted arc represents alignment.

Figure 1: Application of inference rules. POS and relation labels are based on Minipar (Lin, 1998b)

If a complete proof is found (h was generated), the prover concludes that entailment holds. Otherwise, entailment is determined by comparing the minimal cost found during the proof search to some threshold θ.

3 Proof System

Like logic-based systems, our proof system consists of *propositions* (t, h, and intermediate premises), and *inference (entailment) rules*, which derive new propositions from previously established ones.

3.1 Propositions

Propositions are represented as dependency trees, where nodes represent words, and hold a set of features and their values. In our representation these features include the word lemma and part-of-speech, and additional features that may be added during the proof process. Edges are annotated with dependency relations.

3.2 Inference Rules

At each step of the proof an inference rule generates a *derived* tree d from a *source* tree s. A rule is primarily composed of two templates, termed *left-hand-side* (L), and *right-hand-side* (R). *Templates* are dependency subtrees which may contain *variables*. Figure 1(b) shows an inference rule, where V, $N1$ and $N2$ are common variables. L specifies the subtree of s to be modified, and R specifies the new generated subtree. Rule application consists of the following steps:

L matching The prover first tries to match L in s. L is *matched* in s if there exists a one-to-one node mapping function f from L to s, such that: (i) For each node u, $f(u)$ has the same features and feature values as u. Variables match any lemma value in $f(u)$. (ii) For each edge $u \rightarrow v$ in L, there is an edge $f(u) \rightarrow f(v)$ in s, with the same dependency relation. If matching fails, the rule is not applicable to s. Otherwise, successful matching induces *vari-*

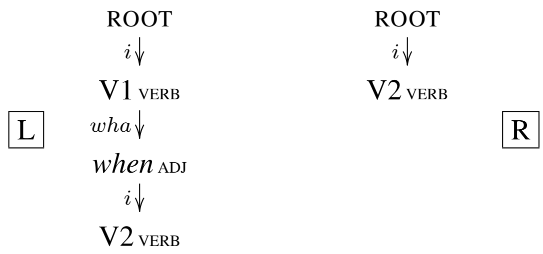


Figure 2: Temporal clausal modifier extraction (introduction rule)

able binding $b(X)$, for each variable X in L, defined as the full subtree rooted in $f(X)$ if X is a leaf, or $f(X)$ alone otherwise. We denote by l the subtree in s to which L was mapped (as illustrated in bold in Figure 1(a), left tree).

R instantiation An instantiation of R, which we denote r, is generated in two steps: (i) creating a copy of R; (ii) replacing each variable X with a copy of its binding $b(X)$ (as set during L matching). In our example this results in the subtree *John saw beautiful Mary*.

Alignment copying the *alignment* relation between pairs of nodes in L and R specifies which modifiers in l that are not part of the rule structure need to be copied to the generated tree r. Formally, for any two nodes u in l and v in r whose matching nodes in L and R are aligned, we copy the daughter subtrees of u in s, which are not already part of l, to become daughter subtrees of v in r. The bold nodes in the right part of Figure 1(b) correspond to r after alignment. *yesterday* was copied to r due to the alignment of its parent verb node.

Derived tree generation by rule type Our formalism has two methods for generating the derived tree: *substitution* and *introduction*, as specified by the rule type. With *substitution* rules, the derived tree d is obtained by making a local modification to the source tree s. Except for this modification s and d are identical (a typical example is a lexical rule, such as *buy* → *purchase*). For this type, d is formed by copying s while replacing l (and the descendants of l's nodes) with r. This is the case for the passive rule. The right part of Figure 1(a) shows the derived tree for the passive rule application. By contrast, *introduction* rules are used to make inferences from a subtree of s, while the other parts of s are ignored and do not affect d. A typical example is inference of a proposition embedded as a relative clause in s. In this case the derived tree d is simply taken to be r. Figure 2 presents such a rule that derives propositions embedded within temporal modifiers. Note that the derived tree does not depend on the main clause. Applying this rule to the right part of Figure 1(b) yields the proposition *John saw beautiful Mary yesterday*.

3.3 Annotation Rules

Annotation rules add features to parse tree nodes, and are used in our system to annotate negation and modality. Annotation rules do not have an R. Instead, nodes of L may contain annotation features. If L is matched in a tree then the annotations are copied to the matched nodes. Annotation rules are applied to t and to each inferred premise prior to any entailment rule application and these features may block inappropriate subsequent rule applications, such as for negated predicates.

4 Rules for Generic Linguistic Structures

Based on the above framework we have manually created a rule base for generic linguistic phenomena.

4.1 Syntactic-Based Rules

These rules capture entailment inferences associated with common syntactic structures. They have three major functions: (i) simplification and canonization of the source tree (categories 6 and 7 in Table 1); (ii) extracting embedded propositions (categories 1, 2, 3); (iii) inferring propositions from non-propositional subtrees (category 4).

4.2 Polarity-Based Rules

Consider the following two examples:

John *knows* that Mary is here $\Rightarrow$ Mary is here.
John *believes* that Mary is here $\nRightarrow$ Mary is here.

Valid inference of propositions embedded as verb complements depends on the verb properties, and the polarity of the context in which the verb appears (positive, negative, or unknown) (Nairn et al., 2006). We extracted from the polarity lexicon of Nairn et al. a list of verbs for which inference is allowed in positive polarity context, and generated entailment

#	Category	Example: source	Example: derived
1	Conjunctions	Helena's very experienced and has played a long time on the tour.	$\Rightarrow$ Helena has played a long time on the tour.
2	Clausal modifiers	But celebrations were muted as many Iranians observed a Shi'ite mourning month.	$\Rightarrow$ Many Iranians observed a Shi'ite mourning month.
3	Relative clauses	The assailants fired six bullets at the car, which carried Vladimir Skobtsov.	$\Rightarrow$ The car carried Vladimir Skobtsov.
4	Appositives	Frank Robinson, a one-time manager of the Indians, has the distinction for the NL.	$\Rightarrow$ Frank Robinson is a one-time manager of the Indians.
5	Determiners	The plaintiffs filed **their** lawsuit last year in U.S. District Court in Miami.	$\Rightarrow$ The plaintiffs filed **a** lawsuit last year in U.S. District Court in Miami.
6	Passive	We have been approached by the investment banker.	$\Rightarrow$ The investment banker approached us.
7	Genitive modifier	Malaysia's crude palm oil output is estimated to have risen by up to six percent.	$\Rightarrow$ The crude palm oil output of Malasia is estimated to have risen by up to six percent.
8	Polarity	Yadav was forced to resign.	$\Rightarrow$ Yadav resigned.
9	Negation, modality	What we've **never** seen is actual costs come down.	What we've never $\overline{\text{seen}}$ is actual costs come down. ($\not\Rightarrow$ What we've seen is actual costs come down.)

Table 1: Summary of rule base for generic linguistic structures.

rules for these verbs (category 8). The list was complemented with a few reporting verbs, such as *say* and *announce*, assuming that in the news domain the speaker is usually considered reliable.

4.3 Negation and Modality Annotation Rules

We use annotation rules to mark negation and modality of predicates (mainly verbs), based on their descendent modifiers. Category 9 in Table 1 illustrates a negation rule, annotating the verb *seen* for negation due to the presence of *never*.

4.4 Generic Default Rules

Generic default rules are used to define default behavior in situations where no case-by-case rules are available. We used one default rule that allows removal of any modifiers from nodes.

5 Lexical-based Rules

These rules have open class lexical components, and consequently are numerous compared to the generic rules described in section 4. Such rules are acquired either lexicographically or automatically.

The rules described in the section 4 are applied whenever their L template is matched in the source premise. For high fan-out rules such as lexical-based rules (e.g. words with many possible synonyms), this may drastically increase the size of the search space. Therefore, the rules described below are applied only if L is matched in the source premise p and R is matched in h.

5.1 Lexical Rules

Lexical entailment rules, such as 'steal $\rightarrow$ take' and 'Britain $\rightarrow$ UK' were created based on WordNet (Fellbaum, 1998). Given p and h, a lexical rule $lemma_p \rightarrow lemma_h$ may be applied if $lemma_p$ and $lemma_h$ are lemmas of open-class words appearing in p and h respectively, and there is a path from $lemma_h$ to $lemma_p$ in the WordNet ontology, through synonym and hyponym relations.

5.2 Lexical-Syntactic Rules

In order to find lexical-syntactic paraphrases and entailment rules, such as 'X strike $Y \rightarrow X$ hit Y' and 'X buy $Y \rightarrow X$ own Y' that would bridge between p and h, we applied the DIRT algorithm (Lin and Pantel, 2001) to the first CD of the Reuters RCV1 corpus[1]. DIRT does not identify the entailment direction, hence we assumed bi-directional entailment.

We calculate off-line only the feature vector of every template found in the corpus, where each path between head nouns is considered a template instance. Then, given a premise p, we first mark all lexical noun alignments between p and h. Next, for every pair of alignments we extract the path between the two nouns in p, labeled $path_p$, and the corresponding path between the aligned nouns in h, labeled $path_h$. We then on-the-fly test whether there is a rule '$path_p \rightarrow path_h$' by extracting the stored feature vectors of $path_p$ and $path_h$ and measuring

[1] http://about.reuters.com/researchandstandards/corpus/

their similarity. If the score exceeds a given threshold[2], we apply the rule to p.

Another enhancement that we added to DIRT is template canonization. At learning time, we transform every template identified in the corpus into its canonized form[3] using a set of morpho-syntactic rules, similar to the ones described in Section 4. In addition, we apply nominalization rules such as 'acquisition of Y by $X \rightarrow X$ acquire Y', which transform a nominal template into its related verbal form. We automatically generate these rules (Ron, 2006), based on Nomlex (Macleod et al., 1998).

At inference time, before retrieving feature vectors, we canonize $path_p$ into $path_p^c$ and $path_h$ into $path_h^c$. We then assess the rule '$path_p^c \rightarrow path_h^c$', and if valid, we apply the rule '$path_p \rightarrow path_h$' to p. In order to ensure the validity of the implicature '$path_p \rightarrow path_p^c \rightarrow path_h^c \rightarrow path_h$', we canonize $path_p$ using the same rule set used at learning time, but we apply only bi-directional rules to $path_h$ (e.g. conjunct heads are not removed from $path_h$).

6 Approximate Matching

As mentioned in section 2, approximate matching is incorporated into our system via a cost function, which estimates the likelihood of h being entailed from a given premise p. Our cost function $C(p, h)$ is a linear combination of two measures: lexical cost, $C_{lex}(p, h)$ and lexical-syntactic cost $C_{lexSyn}(p, h)$:

$$C(p, h) = \lambda C_{lexSyn}(p, h) + (1 - \lambda) C_{lex}(p, h) \quad (1)$$

Let $\hat{m}()$ be a (possibly partial) 1-1 mapping of the nodes of h to the nodes of p, where each node is mapped to a node with the same lemma, such that the number of matched edges is maximized. An edge $u \rightarrow v$ in h is matched in p if $\hat{m}(u)$ and $\hat{m}(v)$ are both defined, and there is an edge $\hat{m}(u) \rightarrow \hat{m}(v)$ in p, with the same dependency relation. $C_{lexSyn}(p, h)$ is then defined as the percentage of unmatched edges in h.

Similarly, $C_{lex}(p, h)$ is the percentage of unmatched lemmas in h, considering only open-class words, defined as:

$$C_{lex}(p, h) = 1 - \frac{\sum_{l \in h} Score(l)}{\#OpenClassWords(h)} \quad (2)$$

where $Score(l)$ is 1 if it appears in p, or if it is a derivation of a word in p (according to WordNet). Otherwise, $Score(l)$ is the maximal Lin dependency-based similarity score between l and the lemmas of p (Lin, 1998a) (synonyms and hypernyms/hyponyms are handled by the lexical rules).

7 System Implementation

Deriving the initial propositions t and h from the input text fragments consists of the following steps: (i) Anaphora resolution, using the MARS system (Mitkov et al., 2002). Each anaphor was replaced by its antecedent. (ii) Sentence splitting, using mxterminator (Reynar and Ratnaparkhi, 1997). (iii) Dependency parsing, using Minipar (Lin, 1998b).

The proof search is implemented as a depth-first search, with maximal depth (i.e. proof length) of 4. If the text contains more than one sentence, the prover aims to prove h from each of the parsed sentences, and entailment is determined based on the minimal cost. Thus, the only cross-sentence information that is considered is via anaphora resolution.

8 Evaluation

		Full (run1)		Lexical (run2)	
Dataset	Task	Acc.	Avg.P	Acc.	Avg.P
Test	IE	0.4950	0.5021	0.5000	0.5379
Official	IR	0.6600	0.6174	0.6450	0.6539
Results	QA	0.7050	0.8085	0.6600	0.8075
	SUM	0.5850	0.6200	0.5300	0.5927
	All	**0.6112**	**0.6118**	**0.5837**	**0.6093**
Dev.	All	0.6443	0.6699	0.6143	0.6559

Table 2: Empirical evaluation - results.

The results for our submitted runs are listed in Table 2, including per-task scores. **run1** is our full system, denoted $\mathcal{F}$. It was tuned on a random sample of 100 sentences from the development set, resulting in $\lambda = 0.6$ and $\theta = 0.6242$ (entailment threshold). **run2** is a lexical configuration, denoted $\mathcal{L}$, in which $\lambda = 0$ (lexical cost only), $\theta = 0.2375$ and the only inference rules used were WordNet Lexical rules. We found that the higher accuracy achieved by $\mathcal{F}$ as compared to $\mathcal{L}$ might have been merely due to a lucky choice of threshold. Setting the threshold to its optimal value with respect to the test set resulted in an accuracy of 62.4% for $\mathcal{F}$, and 62.9% for

[2] We set the threshold to 0.01

[3] The active verbal form with direct modifiers

$\mathcal{L}$. This is also hinted by the very close average precision scores for both systems, which do not depend on the threshold. The last row in the table shows the results obtained for 7/8 of the development set that was not used for tuning, denoted *Dev*, using the same parameter settings. Again, $\mathcal{F}$ performs better than $\mathcal{L}$. $\mathcal{F}$ is still better when using an optimal threshold (which increases accuracy up to 65.3% for $\mathcal{F}$ and 63.9% for $\mathcal{L}$. Overall, $\mathcal{F}$ does not show yet a consistent significant improvement over $\mathcal{L}$.

Initial analysis of the results (based on *Dev*) suggests that the coverage of the current rules is still rather low. Without approximate matching (h must be fully proved using the entailment rules) the recall is only 4.3%, although the precision (92%) is encouraging. Lexical-syntactic rules were applied in about 3% of the attempted proofs, and in most cases involved only morpho-syntactic canonization, with no lexical variation. As a result, entailment was determined mainly by the cost function. Entailment rules managed to reduce the cost in about 30% of the attempted proofs.

We have qualitatively analyzed a subset of false negative cases, to determine whether failure to complete the proof is due to deficient components of the system or due to higher linguistic and knowledge levels. For each pair, we assessed the reasoning steps a successful derivation of h from t would take. We classified each pair according to the most demanding type of reasoning step it would require. We allowed rules that are presently unavailable in our system, as long as they are similar in power to those that are currently available. We found that while the single dominant cause for proof failure is lack of world knowledge, e.g. *the king's son is a member of the royal family*, the combination of missing lexical-syntactic rules and parser failures equally contributed to proof failure.

9 Conclusion

We defined a novel framework for semantic inference at the lexical-syntactic level, which allows a unified representation of a wide variety of inference knowledge. In order to reach reasonable recall on RTE data, we found that we must scale our rule acquisition, mainly by improving methods for automatic rule learning.

Acknowledgments

We are grateful to Cleo Condoravdi for making the polarity lexicon developed at PARC available for this research. We also wish to thank Ruslan Mitkov, Richard Evans, and Viktor Pekar from University of Wolverhampton for running the MARS system for us. This work was partially supported by ISF grant 1095/05, the IST Programme of the European Community under the PASCAL Network of Excellence IST-2002-506778, the Israel Internet Association (ISOC-IL) grant 9022 and the ITC-irst/University of Haifa collaboration.

References

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *AAAI (to appear)*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press.

Dekang Lin and Patrik Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 4(7):343–360.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL*.

Dekang Lin. 1998b. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC*.

C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. 1998. Nomlex: A lexicon of nominalizations. In *EURALEX*.

Ruslan Mitkov, Richard Evans, and Constantin Orasan. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of CICLing*.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5*.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of ANLP*.

Tal Ron. 2006. Generating entailment rules based on online lexical resources. Master's thesis, Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel.