Semantic and Logical Inference Model for Textual Entailment

Dan Roth
University of Illinois
at Urbana-Champaign
Urbana, IL 61801
danr@cs.uiuc.edu

Mark Sammons
University of Illinois
at Urbana-Champaign
Urbana, IL 61801
mssammon@uiuc.edu

Abstract

We compare two approaches to the problem of Textual Entailment: SLIM, a compositional approach modeling the task based on identifying relations in the entailment pair, and BoLI, a lexical matching algorithm. SLIM's framework incorporates a range of resources that solve local entailment problems. A search-based inference procedure unifies these resources, permitting them to interact flexibly. BoLI uses WordNet and other lexical similarity resources to detect correspondence between related words in the Hypothesis and the Text. In this paper we describe both systems in some detail and evaluate their performance on the 3rd PASCAL RTE Challenge. While the lexical method outperforms the relation-based approach, we argue that the relation-based model offers better long-term prospects for entailment recognition.

1 Introduction

We compare two Textual Entailment recognition systems applied to the 3rd PASCAL RTE challenge. Both systems model the entailment task in terms of determining whether the Hypothesis can be "explained" by the Text.

The first system, BoLI (Bag of Lexical Items) uses WordNet and (optionally) other word similarity resources to compare individual words in the Hypothesis with the words in the Text.

The second system, the Semantic and Logical Inference Model (SLIM) system, uses a relational model, and follows the model-theory-based approach of (Braz et al., 2005).

SLIM uses a suite of resources to modify the original entailment pair by augmenting or simplifying either or both the Text and Hypothesis. Terms relating to quantification, modality and negation are detected and removed from the graphical representation of the entailment pair and resolved with an entailment module that handles basic logic.

In this study we describe the BoLI and SLIM systems and evaluate their performance on the 3rd PASCAL RTE Challenge corpora. We discuss some examples and possible improvements for each system.

2 System Description: Bag of Lexical Items (BoLI)

The BoLI system compares each word in the text with a word in the hypothesis. If a word is found in the Text that entails a word in the Hypothesis, that word is considered "explained". If the percentage of the Hypothesis that can be explained is above a certain threshold, the Text is considered to entail the Hypothesis. This threshold is determined using a training set (in this case, the Development corpus), by determining the percentage match for each entailment pair and selecting the threshold that results in the highest overall accuracy.

BoLI uses an extended set of stopwords including auxiliary verbs, articles, exclamations, and discourse markers in order to improve the distinction between Text and Hypothesis. Negation and modality are not explicitly handled.

The BoLI system can be changed by varying the comparison resources it uses. The available

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 107–112,
Prague, June 2007.

resources are: WordNet-derived (Fellbaum, 1998) synonymy, meronymy, membership, and hypernymy; a filtered version of Dekang Lin's word similarity list (Lin, 1998) (only the ten highest-scored entries for each word); and a resource based on a lexical comparison of WordNet glosses.

We tried three main versions; one that used the four WordNet- derived resources ($BoLI$); a second that adds to the first system the Dekang Lin resource ($BoLI_D$); and a third that added to the second system the Gloss resource ($BoLI_G$). We ran them on the Development corpus, and determined the threshold that gave the highest overall score. We then used the highest-scoring version and the corresponding threshold to determine labels for the Test corpus. The results and thresholds for each variation are given in table 1.

3 System Description: Semantic and Logical Inference Model (SLIM)

The SLIM system approaches the problem of entailment via relations: the goal is to recognize the relations in the Text and Hypothesis, and use these to determine whether the Text entails the Hypothesis. A word in the Hypothesis is considered "covered" by a relation if it appears in that relation in some form (either directly or via abstraction). For the Text to entail the Hypothesis, sufficient relations in the Hypothesis must be entailed by relations in the Text to cover the underlying text.

The term "Relation" is used here to describe a predicate-argument structure where the predicate is represented by a verb (which may be inferred from a nominalized form), and the arguments by strings of text from the original sentence. These constituents may be (partially) abstracted by replacing tokens in some constituent with attributes attached to that or a related constituent (for example, modal terms may be dropped and represented with an attribute attached to the appropriate predicate).

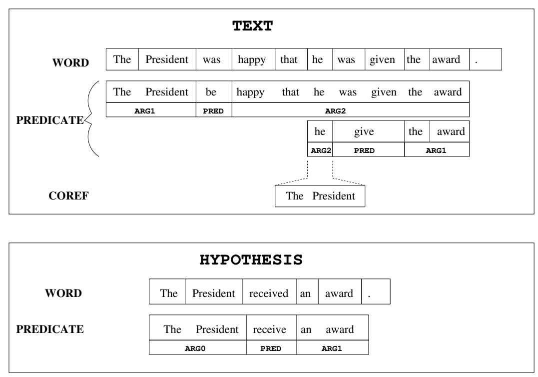
Relations may take other relations as arguments. Examples include "before" and "after" (when both arguments are events) and complement structures.

3.1 Representation

The system compares the Text to the Hypothesis using a "blackboard" representation of the two text fragments (see figure 1). Different types of annotation are specified on different layers, all of which are "visible" to the comparison algorithm. All layers map to the original representation of the text, and each annotated constituent corresponds to some initial subset of this original representation. This allows multiple representations of the same surface form to be entertained.

Figure 1 shows some of the layers in this data structure for a simple entailment pair: the original text in the WORD layer; the relations induced from this text in the PREDICATE layer; and for the Text, a Coreference constituent aligned with the word "he" in the COREFERENCE layer. Note that the argument labels for "give" in the Text indicate that "he" is the theme/indirect object of the predicate "give".

Figure 1: "Blackboard" Representation of Entailment Pairs in SLIM



At compare-time, the coref constituent "The President" will be considered as a substitute for "he" when comparing the relation in the Hypothesis with the second relation in the Text. (The dashed lines indicate that the *coverage* of the coreference constituent is just that of the argument consisting of the word "he".) The relation comparator has access to a list of rules mapping between verbs and their argument types; this will allow it to recognize that the relation "give" can entail "receive", subject to the constraint that the agent of "give" must be the patient of "receive", and vice versa. This, together with the coreference constituent in the Text that aligns with the argument "he", will allow the system to recognize that the Text entails the Hypothesis.

3.2 Algorithm

The SLIM entailment system applies sequences of transformations to the original entailment pair in order to modify one or both members of the pair to make it easier to determine whether the Text entails the Hypothesis. The resources that make these transformations are referred to here as "operators". Each operator is required to use *Purposeful Inference*: before making a change to either entailment pair member, they must take the other member into account. For example, the conjunction expander will generate only those expansions in a text fragment that match structures in the paired text fragment more closely. This constrains the number of transformations considered and can reduce the amount of noise introduced by these operators.

Each such operator serves one of three purposes:

1. **ANNOTATE.** Make some implicit property of the meaning of the sentence explicit.
2. **SIMPLIFY/TRANSFORM.** Remove or alter some section of the Text in order to improve annotation accuracy or make it more similar to the Hypothesis.
3. **COMPARE.** Compare (some elements of) the two members of the entailment pair and assign a score that correlates to how successfully (those elements of) the Hypothesis can be subsumed by the Text.

The system's operators are applied to an entailment pair, potentially generating a number of new versions of that entailment pair. They may then be applied to these new versions. It is likely that only a subset of the operators will fire. It is also possible that multiple operators may affect overlapping sections of one or both members of the entailment pair, and so the resulting perturbations of the original pair may be sensitive to the order of application.

To explore these different subsets/orderings, the system is implemented as a search process over the different operators. The search terminates as soon as a satisfactory entailment score is returned by the comparison operator for a state reached by applying transformation operators, or after some limit to the depth of the search is reached. If entailment is determined to hold, the set of operations that generated the terminal state constitutes a proof of the solution.

3.2.1 Constraining the Search

To control the search to allow for the interdependence of certain operators, each operator may specify a set of pre- and post-conditions. Pre-conditions specify which operators must have fired to provide the necessary input for the current operator. Post-conditions typically indicate whether or not it is desirable to re-annotate the resulting entailment pair (e.g. after an operation that appends a new relation to an entailment pair member), or whether the Comparator should be called to check for entailment.

3.3 System Resources: Annotation

The SLIM system uses a number of standard annotation resources – Part-of-Speech, Shallow- and Full syntactic parsing, Named Entity tagging, and Semantic Role Labelling – but also has a number of more specialized resources intended to recognize implicit predicates from the surface representation in the text, and append these relations to the original text. These resources are listed below with a brief description of each.

Apposition Detector. Uses full parse information to detect appositive constructions, adding a relation that makes the underlying meaning explicit. It uses a set of rules specifying subtree structure and phrase labels.

Complex Noun Phrase Relation Detector. Analyzes long noun phrases and annotates them with their implicit relations. It applies a few general rules expressed at the shallow parse and named entity level.

Modality and Negation Annotator. Abstracts modifiers of relations representing modality or negation into attributes attached to the relation.

Discourse Structure Annotator. Scans the relation structure (presently only at the sentence level) to determine negation and modality of relations embedded in factive and other constructions. It marks the embedded relations accordingly, and where possible, discards the embedding relation.

Coreference Annotator. Uses Named Entity information to map pronouns to possible replacements.

Nominalization Rewriter. Detects certain common nominalized verb structures and makes the relation explicit. The present version applies a small set of very general rules instantiated with a list of embedding verbs and a mapping from nominalized to verbal forms.

3.4 System Resources: Simplification/Transformation

The simplification resources all demonstrate **purposeful inference**, as described in section 3.2.

Idiom Catcher. Identifies and replaces sequences of words corresponding to a list of known idioms, simplifying sentence structure. It can recognize a range of surface representations for each idiom.

Phrasal Verb Replacer. Checks for phrasal verb constructions, including those where the particle is distant from the main verb, replacing them with single verbs of equivalent meaning.

Conjunction Expander. Uses full parse information to detect and rewrite conjunctive argument and predicate structures by expanding them.

Multi-Word Expression Contractor. Scans both members of the entailment pair for compound noun phrases that can be replaced by just the head of the phrase.

3.5 System Resources: Main Comparator

All comparator resources are combined in a single operator for simplicity. This comparator uses the blackboard architecture described in 3.1.

The main comparator compares each relation in the Hypothesis to each relation in the Text, returning "True" if sufficient relations in the Hypothesis are entailed by relations in the Text to cover the underlying representation of the Hypothesis.

For a relation in the Text to entail a relation in the Hypothesis, the Text predicate must entail the Hypothesis predicate, and all arguments of the Hypothesis relation must be entailed by arguments of the Text relation. This entailment check also accounts for attributes such as negation and modality.

As part of this process, a set of rules that map between predicate- argument structures (some handwritten, most derived from VerbNet) are applied on-the-fly to the pair of relations being compared. These rules specify a mapping between predicates and a set of constraints that apply to the mappings between arguments of the predicates. For example, the agent of the relation "sell" should be the theme of the relation "buy", and vice versa.

When comparing the arguments of predicates, the system uses BoLI with the same configuration and threshold that give the best performance on the development set.

3.6 Comparison to Similar Approaches

Like (de Marneffe et al., 2005), SLIM's representation abstracts away terms relating to negation, modality and quantification. However, it uses them as part of the comparison process, not as features to be used in a classifier. In contrast to (Braz et al., 2005), SLIM considers versions of the entailment pair with and without simplifications offered by preprocessing modules, rather than reasoning only about the simplified version; and rather than formulating the subsumption (entailment) problem as a hierarchical linear program or classification problem, SLIM defers local entailment decisions to its modules and returns a positive label for a constituent only if these resources return a positive label for all subconstituents. Finally, SLIM returns an overall positive label if all words in the Hypothesis can be 'explained' by relations detected in the Hypothesis and matched in the Text, rather than requiring all detected relations in the Text to be entailed by relations in the Hypothesis.

4 Experimental Results

Table 3 presents the peformance of the BoLI and SLIM systems on the 3rd PASCAL RTE Challenge. The version of SLIM used for the Development corpus was incomplete, as several modules (Multi-word Expression, Conjunction, and Apposition) were still being completed at that time. Table 1 indicates the performance of 3 different versions of the BoLI system on the Development corpus as described in section 2.

To investigate the improvement of performance for the SLIM system relative to the available resources, we conducted a limited ablation study. Table 2 shows the performance for 3 different versions of the SLIM system on 100 entailment pairs each from the IE and QA subtasks of the Test corpus. The "full" (f) system includes all available resources. The "intermediate" (i) system excludes the resources we consider most likely to introduce errors, the Multiword Expression module and the most general Nominalization rewrite rules in the Nominalization Rewriter. The "strict" (s) system also omits the Apposition and Complex Noun Phrase

Table 1: Accuracy and corresponding threshold for versions of BoLI on the Development corpus.

TASK	Accuracy	Threshold
$BoLI$	0.675	0.667
$BoLI_D$	0.650	0.833
$BoLI_G$	0.655	0.833

Table 2: Results for different versions of SLIM on subsets of the Test and Develoment corpora.

System	**SLIM s**	**SLIM i**	**SLIM f**
Dev IE	-	-	0.650
Dev QA	-	-	0.660
Test IE	0.480	0.480	0.470
Test QA	0.680	0.710	0.710

modules. To give a sense of how well the complete SLIM system does on the Development corpus, the results for the full SLIM system on equal-sized subsets of the IE and QA subtasks of the Development corpus are also shown.

5 Discussion

From Table 3, it is clear that BoLI outperforms SLIM in every subtask.

The ablation study in Table 2 shows that adding new resources to SLIM has mixed benefits; from the samples we used for evaluation, the intermediate system would be the best balance between module coverage and module accuracy.

In the rest of this section, we analyze the results and each system's behavior on several examples from the corpus.

5.1 BoLI

There is a significant drop in performance of the BoLI from the Development corpus to the Test corpus, indicating that the threshold somewhat overfitted to the data used to train it. The performance drop when adding the gloss and Dekang Lin word similarity resources is not necessarily surprising, as these resources are clearly noisy, and so may increase similarity based on inappropriate word pairs.

In the following example, the word similarity is high, but the structure of the two text fragments gives the relevant words different overall meaning (here, that one subset of the matched words does not apply to the other):

id=26 Text: Born in Kingston-upon-Thames, Surrey, Brockwell played his county cricket for the very strong Surrey side of the last years of the 19th century.

Hypothesis: Brockwell was born in the last years of the 19th century.

From this example it is clear that in addition to the role of noise from these additional resources, the structure of text plays a major role in meaning, and this is exactly what BoLI cannot capture.

5.2 SLIM

The ablation study for the SLIM system shows a trade-off between precision and recall for some resources. In this instance, adding resources improves performance significantly, but including noisy resources also implies a ceiling on overall performance will ultimately be reached.

The following example shows the potentially noisy possessive rewrite operator permitting successful entailment:

id=19 Text: During Reinsdorf's 24 seasons as chairman of the White Sox, the team has captured Americal League division championships three times, including an AL Central title in 2000.

Transformed Text: During Reinsdorf have 24 seasons as chairman of the White Sox ...

Hypothesis: Reinsdorf was chairman of the White Sox for 24 seasons.

There are a number of examples where relaxed operators result in false positives, but where the negative label is debatable. In the next example, the apposition module adds a new relation and the Nominalization Rewriter detects the hypothesis using this new relation:

id=102 Hypothesis: He was initially successful, negotiating a 3/4 of 1 percent royalty on all cars sold by the Association of Licensed Automobile Manufacturers, the ALAM.

Transformed Text: ... Association of Licensed Automobile Manufacturers is the ALAM.

Hypothesis: The ALAM manufactured cars.

Finally, some modules did not fire as they should; for example 15, the conjunction module did not expand the conjunction over predicates. For example 24, the nominalization rewriter did not detect "plays in the NHL" from "is a NHL player". In example 35, the apposition module did not detect that "Harriet

Table 3: Results for SLIM and BoLI on the Pascal Development and Test Corpora. Results marked with an asterisk indicate not all system resources were available at the time the system was run.

Corpus	Development					Test				
Subtask	IE	IR	QA	SUM	OVERALL	IE	IR	QA	SUM	OVERALL
BoLI	0.560	0.700	0.790	0.690	0.675	0.510	0.710	0.830	0.575	0.656
SLIM	0.580*	0.595*	0.650*	0.545*	0.593*	0.485	0.6150	0.715	0.575	0.5975

Lane, niece of President James" could be rewritten.

Of course, there are also many examples where the SLIM system simply does not have appropriate resources (e.g. numerical reasoning, coreference requiring semantic categorization).

6 Conclusion

While BoLI outperforms SLIM on the PASCAL RTE 3 task, there is no clear way to improve BoLI. It is clear that for the PASCAL corpora, the distributions over word similarity between entailment pair members in positive and negative examples are different, allowing this simple approach to perform relatively well, but there is no guarantee that this is generally the case, and it is easy to create an adversarial corpus on which BoLI performs very badly (e.g., exchanging arguments or predicates of different crelations in the Text), no matter how good the word-level entailment resources are. This approach also offers no possibility of a meaningful explanation of the entailment decision.

SLIM, on the other hand, by offering a framework to which new resources can be added in a principled way, can be extended to cover new entailment phenomena in an incremental, local (i.e. compositional) way. The results of the limited ablation study support this conclusion, though the poor performance on the IE task indicates the problems with using lower-precision, higher-recall resources.

Overall, we find the results for the SLIM system very encouraging, as they support the underlying concept of incremental improvement, and this offers a clear path toward better performance.

6.1 Acknowledgements

We gratefully acknowledge the work on SLIM modules by Ming-Wei Chang, Michael Connor, Quang Do, Alex Klementiev, Lev Ratinov, and Vivek Srikumar. This work was funded by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) program, a grant from Boeing, and a gift from Google.

References

Johan Bos and Katja Markert. 2005. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

R. Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. Knowledge representation for semantic entailment and question-answering. In *IJCAI-05 Workshop on Knowledge and Reasoning for Question Answering*.

Marie-Catherine de Marneffe, Bill MacCartney, Trond Grenager, Daniel Cer, Anna Rafferty, and Christopher Manning. 2005. Learning to distinguish valid textual entailments. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

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

Sophia Katrenko and Peter Adriaans. 2005. Using maximal embedded syntactic subtrees for textual entailment recognition. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proc. of the International Conference on Machine Learning (ICML)*.

Marta Tatu, Brandon Iles, John Slavick, Adrian Novischi, and Dan Moldovan. 2005. Cogex at the second recognizing textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.