Modeling Semantic Containment and Exclusion in Natural Language Inference

Bill MacCartney
Stanford University
wcmac@cs.stanford.edu

Christopher D. Manning
Stanford University
manning@cs.stanford.edu

Abstract

We propose an approach to natural language inference based on a model of *natural logic*, which identifies valid inferences by their lexical and syntactic features, without full semantic interpretation. We greatly extend past work in natural logic, which has focused solely on semantic containment and monotonicity, to incorporate both semantic exclusion and implicativity. Our system decomposes an inference problem into a sequence of atomic edits linking premise to hypothesis; predicts a lexical entailment relation for each edit using a statistical classifier; propagates these relations upward through a syntax tree according to semantic properties of intermediate nodes; and composes the resulting entailment relations across the edit sequence. We evaluate our system on the FraCaS test suite, and achieve a 27% reduction in error from previous work. We also show that hybridizing an existing RTE system with our natural logic system yields significant gains on the RTE3 test suite.

1 Introduction

A necessary (if not sufficient) condition for true natural language understanding is a mastery of open-domain *natural language inference* (NLI): the task of determining whether a natural-language hypothesis can be inferred from a given premise. Indeed, NLI can enable more immediate applications, such as semantic search and question answering (Harabagiu and Hickl, 2006). In recent years a spectrum of approaches to robust, open-domain NLI have been explored within the context of the Recognizing Textual Entailment challenge (Dagan et al., 2005). Up to now, the most successful approaches have used fairly shallow semantic representations, relying on measures of lexical or semantic overlap (Jijkoun and de Rijke, 2005), pattern-based relation extraction (Romano et al., 2006), or approximate matching of predicate-argument structure (Hickl et al., 2006). Such methods, while robust and often effective, are at best partial solutions, unable to explain even simple forms of logical inference. For example, most shallow approaches would fail to license the introduction of *large* in the following example:

(1) Every firm saw costs grow more than expected, even after adjusting for inflation.
Every *large* firm saw costs grow.

At the other extreme, some researchers have approached NLI as logical deduction, building on work in theoretical semantics to translate sentences into first-order logic (FOL), and then applying a theorem prover or model builder (Akhmatova, 2005; Fowler et al., 2005). Regrettably, such approaches tend to founder on the myriad complexities of full semantic interpretation, including tense, aspect, causality, intensionality, modality, vagueness, idioms, indexicals, ellipsis, and many other issues. (What is the right FOL representation of (1), for example?) FOL-based systems that have attained high precision (Bos and Markert, 2006) have done so at the cost of very poor recall.

This work explores a middle way, by developing a computational model of what Lakoff (1970) called *natural logic*, which characterizes valid patterns of inference in terms of syntactic forms re-

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 521–528
Manchester, August 2008

sembling natural language as much as possible.[1] For example, natural logic might sanction (1) by observing that: in ordinary (*upward monotone*) contexts, deleting modifiers preserves truth; in *downward monotone* contexts, inserting modifiers preserves truth; and *every* is downward monotone in its restrictor NP. A natural logic system can thus achieve the expressivity and precision needed to handle a great variety of simple logical inferences, while sidestepping the difficulties of full semantic interpretation.

2 A theory of natural logic

The natural logic approach originated in traditional logic (e.g., Aristotle's syllogisms), and was revived in a formal form by van Benthem (1986) and Sánchez Valencia (1991), who proposed a natural logic based on categorial grammar to handle inferences involving containment relations and upward and downward monotonicity, such as (1). Their *monotonicity calculus* explains inferences involving even nested inversions of monotonicity, but because it lacks any representation of exclusion (as opposed to containment), it cannot explain simple inferences such as (38) and (205) in table 2, below.

Another model which arguably follows the natural logic tradition (though not presented as such) was developed by Nairn et al. (2006) to explain inversions and nestings of implicative (and factive) predicates, as in *Ed did not forget to force Dave to leave* $\models$ *Dave left*. Their *implication projection algorithm* bears some resemblance to the monotonicity calculus, but does not incorporate containment relations or explain interactions between implicatives and monotonicity, and thus fails to license *John refused to dance* $\models$ *John didn't tango*.

We propose a new model of natural logic which generalizes the monotonicity calculus to cover inferences involving exclusion, and (partly) unifies it with Nairn et al.'s model of implicatives. We (1) augment the set of entailment relations used in monotonicity calculus to include representations of exclusion; (2) generalize the concept of monotonicity to one of *projectivity*, which describes how the entailments of a compound expression depend on the entailments of its parts; and (3) describe a weak proof procedure based on composing entailment relations across chains of atomic edits.

Entailment relations. We employ an inventory of seven mutually exclusive *basic entailment relations*, defined by analogy with set relations: equivalence (*couch* $=$ *sofa*); forward entailment (*crow* $\sqsubset$ *bird*) and its converse (*European* $\sqsupset$ *French*); negation, or exhaustive exclusion (*human* ˆ *nonhuman*); alternation, or non-exhaustive exclusion (*cat* | *dog*); cover, or non-exclusive exhaustion (*animal* $\smile$ *nonhuman*); and independence (*hungry* # *hippo*), which covers all other cases. As in the monotonicity calculus, we define these relations for expressions of every semantic type: sentences, common and proper nouns, transitive and intransitive verbs, adjectives, and so on. For example, among generalized quantifiers, we find that *all* $=$ *every*, *every* $\sqsubset$ *some*, *some* ˆ *no*, *no* | *every*, *at least four* $\smile$ *at most six*, and *most* # *ten or more*.[2]

Projectivity. In order to explain the entailments of a compound expression as a function of the entailments of its parts, we categorize semantic functions according to their *projectivity class*, a concept which generalizes both Sánchez Valencia's monotonicity classes (upward, downward, and non-monotone) and the nine implication signatures of Nairn et al. The projectivity class of a function f describes how the entailment relation between $f(x)$ and $f(y)$ depends on the entailment relation between x and y. Consider simple negation (*not*). Like most functions, it projects $=$ and # without change (*not happy* $=$ *not glad* and *isn't swimming* # *isn't hungry*). As a downward monotone function, it swaps $\sqsubset$ and $\sqsupset$ (*didn't kiss* $\sqsupset$ *didn't touch*). But we can also establish that it projects ˆ without change (*not human* ˆ *not nonhuman*) and swaps | and $\smile$ (*not French* $\smile$ *not German*, *not more than 4* | *not less than 6*). By contrast, an implicative like *refuse*, though it also swaps $\sqsubset$ and $\sqsupset$ (*refuse to tango* $\sqsupset$ *refuse to dance*), projects ˆ as | (*refuse to stay* | *refuse to go*) and projects both | and $\smile$ as # (*refuse to tango* # *refuse to waltz*).

Projectivity thus allows us to determine the entailments of a compound expression recursively, by propagating entailments upward through a semantic composition tree according to the projectivity class of each node on the path to the root. For example, the semantics of *Nobody can enter with-*

[1] Natural logic should not be confused with *natural deduction*, a proof system for first-order logic.

[2] Some of these assertions assume *existential import*, i.e., that the predicates to which the quantifiers are applied have non-empty denotations. This assumption, standard in traditional logic, seems justifiable in the context of informal natural language inference (Böttner, 1988).

out a shirt might be represented by the tree (*nobody* (*can* ((*without* (*a shirt*)) *enter*))). Since *shirt* $\sqsubset$ *clothes*, and since *without* is downward monotone, we have *without shirt* $\sqsupset$ *without clothes*. Since *nobody* is also downward monotone, it follows that *Nobody can enter without a shirt* $\sqsubset$ *Nobody can enter without clothes*.

Inference. Let $x' = e(x)$ be the result of applying an *atomic edit* e (the insertion, deletion, or substitution of a subexpression) to a compound expression x. The entailment relation between x and x' is found by projecting the entailment relation generated by e upward through x's semantic composition tree. Substitutions generate relations according to the meanings of the substituends. Most deletions generate the $\sqsubset$ relation (*red socks* $\sqsubset$ *socks*). (Insertions are symmetric: they typically generate $\sqsupset$.) However, some items have special behavior. For example, deleting (or inserting) *not* generates $\hat{}$ (*not hungry* $\hat{}$ *hungry*).

If two expressions are connected by a chain of atomic edits, we can determine the entailment relation between them by composing (as in Tarskian relation algebra) the entailment relations generated by each edit. The result may be a basic entailment relation, or may be a union of such relations, with larger unions conveying less information about entailment. This possibility, coupled with the need to find a chain of atomic edits which preserves relevant entailment relations, limits the power of the proof procedure described.

Implicatives. The account of implicatives and factives given by Nairn et al. hinges on a classification of implicative and factive operators into nine *implication signatures*, according to their implications—positive (+), negative (–), or null (○)—in both positive and negative contexts. Thus *refuse* has implication signature –/○, because it carries a negative implication in a positive context (*refused to dance* implies *didn't dance*), and no implication in a negative context (*didn't refuse to dance* implies neither *danced* nor *didn't dance*).

Most of the phenomena observed by Nairn et al. can be explained within our framework by specifying, for each signature, the relation generated when an operator of that signature is deleted from a compound expression. For example, deleting signature –/○ generates $|$ (*Jim refused to dance* $|$ *Jim danced*); under negation, this is projected as $\smile$ (*Jim didn't refuse to dance* $\smile$ *Jim didn't dance*). By contrast, deleting signature ○/– generates $\sqsupset$ (*Jim attempted to dance* $\sqsupset$ *Jim danced*); under negation, this is projected as $\sqsubset$ (*Jim didn't attempt to dance* $\sqsubset$ *Jim didn't dance*).[3]

We can also account for monotonicity effects of implicative and factive operators by describing the projectivity properties of each implication signature: signatures +/–, +/○, and ○/– are upward monotone (*attempt to tango* $\sqsubset$ *attempt to dance*); signatures –/+, –/○, and ○/+ are downward monotone (*refuse to dance* $\sqsubset$ *refuse to tango*); and signatures +/+, –/–, and ○/○ are non-monotone (*think dancing is fun* # *think tangoing is fun*).

3 The NatLog system

Our implementation of natural logic, the *NatLog* system, uses a multi-stage architecture like those of (Marsi and Krahmer, 2005; MacCartney et al., 2006), comprising (1) linguistic analysis, (2) alignment, (3) lexical entailment classification, (4) entailment projection, and (5) entailment composition. We'll use the following inference as a running example:

(2) Jimmy Dean refused to move without blue jeans.
James Dean didn't dance without pants.

The example is admittedly contrived, but it compactly exhibits containment, exclusion, and implicativity. How the NatLog system handles this example is depicted in table 1.

Linguistic analysis. Relative to other NLI systems, the NatLog system does comparatively little linguistic pre-processing. We rely on the Stanford parser (Klein and Manning, 2003), a Penn Treebank-trained statistical parser, for tokenization, lemmatization, part-of-speech tagging, and phrase-structure parsing.

By far the most important analysis performed at this stage, however, is *projectivity marking*, in which we compute the effective projectivity for each token span in each input sentence. In the premise of (2), for example, we want to determine that the effective projectivity is upward monotone

[3] Factives, however, do not fit as neatly as implicatives: For example, deleting signature +/+ generates $\sqsubset$ (*Jim forgot that dancing is fun* $\sqsubset$ *dancing is fun*); yet under negation, this is projected not as $\sqsupset$, but as $|$ (*Jim didn't forget that dancing is fun* $|$ *dancing isn't fun*). The problem arises because the implication carried by a factive is not an entailment, but a presupposition. As is well known, the projection behavior of presuppositions differs from that of entailments (van der Sandt, 1992). In the current work, we set presuppositions aside.

premise	Jimmy Dean	refused to			move	without	blue	jeans
hypothesis	James Dean		did	n't	dance	without		pants
edit index	1	2	3	4	5	6	7	8
edit type	SUB	DEL	INS	INS	SUB	MAT	DEL	SUB
lex features	str_sim=0.67	implic:+/○	cat:aux	cat:neg	hyponym			hypernym
lex entrel	=	|	=	ˆ	⊐	=	⊏	⊏
projectivity	↑	↑	↑	↑	↓	↓	↑	↑
atomic entrel	=	|	=	ˆ	⊏	=	⊏	⊏
composition	=	|	|	⊏	⊏	⊏	⊏	⊏

Table 1: An example of the operation of the NatLog model.

unary operator: *without*
  pattern: `IN < /^[Ww]ithout$/`
  argument 1: projectivity ↓ on dominating PP
    pattern: `__ > PP=proj`

binary operator: *most*
  pattern: `JJS < /^[Mm]ost$/ !> QP`
  argument 1: projectivity $\not\updownarrow$ on dominating NP
    pattern: `__ >+(NP) (NP=proj !> NP)`
  argument 2: projectivity ↑ on dominating S
    pattern: `__ >> (S=proj !> S)`

Figure 1: Some projectivity operator definitions.

for *Jimmy Dean* and *refused to*, downward monotone for *move* and *without*, and upward monotone for *blue* and *jeans*. Our choice of a Treebank-trained parser (driven by the goal of broad coverage) complicates this effort, because the nesting of constituents in phrase-structure parses does not always correspond to the structure of idealized semantic composition trees. Our solution is imperfect but effective. We define a list of operator types affecting projectivity (e.g., implicatives like *refuse to*, prepositions like *without*), and for each type we specify its arity and a Tregex tree pattern (Levy and Andrew, 2006) which permits us to identify its occurrences in our Treebank parses. We also specify, for each argument position of each type, both the projectivity class and another Tregex pattern which helps us to determine the sentence span over which the operator's effect is projected. (Figure 1 shows some example definitions.) The marking process computes these projections, performs projectivity composition where needed, and marks each token span with its final effective projectivity.

Alignment. Next, we establish an *alignment* between the premise P and hypothesis H, represented by a sequence of *atomic edits* over spans of word tokens. This alignment representation is symmetric and many-to-many, and is general enough to include various other alignment representations as special cases. We define four edit types: *deletion* (DEL) of a span from P, *insertion* (INS) of a span into H, *substitution* (SUB) of an H span for a P span, and *match* (MAT) of an H span to a P span. Each edit is parameterized by the token indices at which it operates, and edit indices may "cross", permitting representation of movement. The first four lines of table 1 depict a possible alignment for our example problem.

An alignment decomposes an inference problem into a sequence of atomic inference problems, one for each atomic edit. Note that edits are ordered, and that this ordering defines a path from P to H through intermediate forms. (Edit order need not correspond to sentence order, though it does in our example.) The relative ordering of certain kinds of edits (e.g., the insertion of *not*) may influence the effective projectivity applicable for other edits; consequently, the NatLog system can reorder edits to maximize the benefit of the projectivity marking performed during linguistic analysis.

This paper does not present new algorithms for alignment; we focus instead on identifying entailment relations between aligned sentence pairs. The experiments described in sections 4 and 5 use alignments from other sources.

Lexical entailment classification. Much of the heavy lifting in the NatLog system is done by the *lexical entailment model*, which uses a classifier to predict an entailment relation for each atomic edit based solely on features of the lexical items involved, independent of context. (For example, this model should assign the entailment relation ⊐ to the edit SUB(*move*, *dance*), regardless of whether the effective projectivity at the locus of the edit is upward monotone, downward monotone, or something else.) In the case of a SUB edit, the features include:

- WordNet-derived measures of synonymy, hyponymy, and antonymy between substituends;

- other features indicating semantic relatedness: the WordNet-based Jiang-Conrath measure (Jiang and Conrath, 1997) and a feature based on NomBank (Meyers et al., 2004);
- string similarity features based on Levenshtein string-edit distance between lemmas;
- lexical category features, indicating whether the substituends are prepositions, possessives, articles, auxiliaries, pronouns, proper nouns, operator adjectives, punctuation, etc.;
- quantifier category features, which identify classes of quantifiers with similar properties;
- a feature for unequal numeric expressions

For DEL edits, we use only the lexical category features and a feature based on a custom-built resource which maps implicatives and factives to their implication signatures. (As noted in section 2, however, most DEL edits just have $\sqsubset$ as the target lexical entailment relation.) INS edits are treated symmetrically.

The model uses a decision tree classifier trained on 2,449 hand-annotated training examples (1,525 SUB edits and 924 DEL/INS edits). The decision tree is minimally pruned, and contains about 180 leaves. When tested on the training data, the classifier achieves >99% accuracy, indicating that our feature representation successfully captures nearly all relevant distinctions between examples.

Lexical features and lexical entailment relations for our example appear on lines 5 and 6 of table 1.

Entailment projection. The lexical entailment relations generated by each atomic edit can now be projected upward to determine the corresponding atomic entailment relations, that is, the entailment relations between successive intermediate forms on the path from P to H, as defined by the alignment. Strictly speaking, the effective projectivity for a particular edit should be computed based on the intermediate form upon which the edit operates, since the projectivity properties of this form can depend on preceding edits. However, the NatLog system minimizes the need to compute projectivity in intermediate forms by reordering the edits in an alignment in such a way that effective projectivity can, in most cases, simply be taken from the projectivity marking of P and H performed during the linguistic analysis stage.

The effective projectivity and resulting atomic entailment relation for each edit in our running example are depicted in lines 7 and 8 of table 1. For all (non-MAT) edits but one, the effective projectivity is upward monotone, so that the atomic entailment relation is identical with the lexical entailment relation. However, the SUB(*move*, *dance*) edit occurs in a downward monotone context, so that the lexical relation $\sqsupset$ is converted to $\sqsubset$ at the atomic level.

Entailment composition. Finally, the atomic entailment relations predicted for each edit are combined, via relation composition, to produce an overall prediction for the inference problem. Relation composition is deterministic, and for the most part follows intuitive rules: $\sqsubset$ composed with $\sqsubset$ yields $\sqsubset$; $\sqsupset$ composed with $\sqsupset$ yields $\sqsupset$; $\#$ composed with any relation yields $\#$; $=$ composed with any relation yields that relation, and so on. Composition tends to "degenerate" towards $\#$, in the sense that the composition of a chain of randomly-selected relations tends toward $\#$ as the chain grows longer. This chaining of entailments across edits can be compared to the method presented in (Harmeling, 2007); however, that approach assigns to each edit merely a probability of preserving truth, not an entailment relation.

The last line of table 1 shows the cumulative composition of the atomic entailment relations in the line above. Particular noteworthy is the fact that $|$ and $\hat{}$ compose to yield $\sqsubset$. (To illustrate: if A excludes B (*fish* $|$ *human*) and B is the negation of C (*human* $\hat{}$ *nonhuman*), then A entails C (*fish* $\sqsubset$ *nonhuman*).) The final entailment relation in this line, $\sqsubset$, is NatLog's final (and correct) answer for our example problem.

4 Evaluating on FraCaS problems

The FraCaS test suite (Cooper et al., 1996) contains 346 NLI problems, divided into nine sections, each focused on a specific category of semantic phenomena (listed in table 3). Each problem consists of one or more premise sentences, a question sentence, and one of three answers: *yes* (the union of $\sqsubset$ and $=$), *no* (the union of $|$ and $\hat{}$), or *unknown* (the union of $\sqsupset$, $\smile$, and $\#$). Table 2 shows some example problems.

To facilitate comparison with previous work, we have evaluated our system using a version of the FraCas data prepared by (MacCartney and Manning, 2007), in which multiple-premise problems (44% of the total) and problems lacking a hypothesis or a well-defined answer (3% of the total) are excluded; question sentences have been converted

§	ID	Premise	Hypothesis	Ans
1	38	No delegate finished the report.	Some delegate finished the report on time.	*no*
1	48	At most ten commissioners spend time at home.	At most ten c...s spend a lot of time at home.	*yes*
2	83	Either Smith, Jones or Anderson signed the contract.	Jones signed the contract.	*unk*
5	205	Dumbo is a large animal.	Dumbo is a small animal.	*no*
6	233	ITEL won more orders than APCOM.	ITEL won some orders.	*yes*
9	335	Smith believed that ITEL had won the contract in 1992.	ITEL won the contract in 1992.	*unk*

Table 2: Illustrative examples from the FraCaS test suite

System	#	P %	R %	Acc %
most common class	183	55.74	100.00	55.74
MacCartney07	183	68.89	60.78	59.56
NatLog	183	89.33	65.69	**70.49**

§	Section	#	P %	R %	Acc %
1	Quantifiers	44	95.24	100.00	**97.73**
2	Plurals	24	90.00	64.29	75.00
3	Anaphora	6	100.00	60.00	50.00
4	Ellipsis	25	100.00	5.26	24.00
5	Adjectives	15	71.43	83.33	80.00
6	Comparatives	16	88.89	88.89	81.25
7	Temporal	36	85.71	70.59	58.33
8	Verbs	8	80.00	66.67	62.50
9	Attitudes	9	100.00	83.33	88.89
1, 2, 5, 6, 9		108	90.38	85.45	**87.04**

Table 3: Performance on FraCaS problems (three-way classification). The columns show the number of problems, precision and recall for the *yes* class, and accuracy. Results for NatLog are broken out by section.

		guess			
		yes	*no*	*unk*	total
answer	*yes*	67	4	31	102
	no	1	16	4	21
	unk	7	7	46	60
	total	75	27	81	183

Table 4: Confusions on FraCaS data (all sections)

to declarative hypotheses; and alignments between premise and hypothesis have been automatically generated and manually corrected.

Results are shown in table 3. We achieve overall accuracy of 70.49%, representing a 27% error reduction from (MacCartney and Manning, 2007). In the section concerning quantifiers, which is both the largest and the most amenable to natural logic, all problems but one are answered correctly.[4] We also answer all but one problems correctly in the (admittedly small) section on attitudes, which involves implicatives and factives. Unsurprisingly, performance is mediocre in four sections concerning semantic phenomena (e.g., ellipsis) not relevant to natural logic and not modeled by the system. But in the other five sections (about 60% of the problems), we achieve accuracy of 87.04%, an error reduction of 61% from (MacCartney and Manning, 2007). What's more, precision is high in nearly every section: even outside its areas of expertise, the system rarely predicts entailment when none exists.

Since the NatLog system was developed with FraCaS problems in mind, these results do not constitute a proper evaluation on unseen test data. On the other hand, the system does no training on FraCaS data, and has had no opportunity to learn its biases. (Otherwise, accuracy on §4 could not fall so far below the baseline.) The system not only answers most problems correctly, but usually does so for valid reasons, particular within its areas of expertise. All in all, the results fulfill our main goal in testing on FraCaS: to demonstrate the representational and inferential adequacy of our model of natural logic.

The confusion matrix shown in table 4 reveals an interesting property of the NatLog system. The commonest confusions are those where the answer is *yes* but we guess *unknown*. This reflects both the bias toward *yes* in the FraCaS data, and the system's tendency to predict *unknown* (entailment relation #) when confused: given the composition rules for entailment relations, the system can predict *yes* only if all atomic-level predictions are either $\sqsubset$ or $=$.

5 Evaluating on RTE problems

NLI problems from the PASCAL RTE Challenge (Dagan et al., 2005) differ from FraCaS problems in several important ways. (See table 5 for examples.) Instead of textbook examples of seman-

[4] In fact, the sole exception is disputable, since it hinges on whether *many* refers to proportion (apparently, the view held by the FraCaS authors) or absolute quantity.

ID	Premise	Hypothesis	Answer
71	As leaders gather in Argentina ahead of this weekends regional talks, Hugo Chávez, Venezuela's populist president is using an energy windfall to win friends and promote his vision of 21st-century socialism.	Hugo Chávez acts as Venezuela's president.	*yes*
788	Democrat members of the Ways and Means Committee, where tax bills are written and advanced, do not have strong small business voting records.	Democrat members had strong small business voting records.	*no*

Table 5: Illustrative examples from the RTE3 development set

tic phenomena, RTE problems are more natural-seeming, with premises collected "in the wild" from newswire. The premises are much longer, averaging 35 words (vs. 11 words for FraCaS). Also, RTE aims at binary classification: the RTE *no* combines the *no* and *unk* answers in FraCaS.

Due to the character of RTE problems, we do not expect NatLog to be a good general-purpose solution to solving all RTE problems. First, most RTE problems depend on forms of inference, such as paraphrase, temporal reasoning, or relation extraction, which NatLog is not designed to address. Second, in most RTE problems, the edit distance between premise and hypothesis is relatively large. More atomic edits means a greater chance that errors made in lexical entailment classification or projection will propagate, via entailment composition, to the system's final output. Rather, in applying NatLog to RTE, we hope to make reliable predictions on a subset of RTE problems, trading recall for precision. If we succeed, then we may be able to hybridize with a broad-coverage RTE system to obtain better results than either system individually—the same strategy that was adopted by (Bos and Markert, 2006) for their FOL-based system. For this purpose, we have chosen to use the Stanford RTE system described in (de Marneffe et al., 2006). We also use the Stanford system to generate alignments when evaluating NatLog on RTE problems.

Table 6 shows the performance of NatLog on RTE3 data. Relative to the Stanford system, NatLog achieves high precision on its *yes* predictions—above 70%—suggesting that hybridizing may be effective. For comparison, the FOL-based system reported in (Bos and Markert, 2006) attained a similarly high precision of 76% on RTE2 problems, but was able to make a positive prediction in only about 4% of cases. NatLog makes positive predictions far more often—in about 25% of cases.

The Stanford system makes *yes*/*no* predictions by thresholding a real-valued *inference score*. To construct a hybrid system, we adjust the Stanford inference scores by $+\Delta$ or $-\Delta$, depending on whether or not NatLog predicts *yes*. We choose Δ by optimizing development set accuracy, while adjusting the threshold to generate balanced predictions (that is, equal numbers of *yes* and *no* predictions). As an additional experiment, we fix Δ at this value and then adjust the threshold to optimize development set accuracy, resulting in an excess of *yes* predictions. (Since this optimization is based solely on development data, its use on test data is fully legitimate.) Results for these two cases are shown in table 6. The parameters tuned on development data gave good results on test data. The optimized hybrid system attained an absolute accuracy gain of 4% over the Stanford system, corresponding to an extra 32 problems answered correctly. This result is statistically significant ($p < 0.05$, McNemar's test, 2-tailed).

System	Data	% Yes	P %	R %	Acc %
Stanford	dev	50.25	68.66	66.99	67.25
	test	50.00	61.75	60.24	60.50
NatLog	dev	22.50	**73.89**	32.38	59.25
	test	26.38	**70.14**	36.10	59.38
Hybrid, bal.	dev	50.00	70.25	68.20	68.75
	test	50.00	65.50	63.90	64.25
Hybrid, opt.	dev	56.00	69.20	75.24	**70.00**
	test	54.50	64.45	68.54	**64.50**

Table 6: Performance of various systems on RTE3 (two-way classification). The columns show the data set used (800 problems each), the proportion of *yes* predictions, precision and recall for the *yes* class, and accuracy.

The gains attributable to NatLog are exemplified by problem 788 (table 5). NatLog sanctions the deletion of a restrictive modifier and an appositive from the premise, and recognizes that deleting a negation generates a contradiction; thus it correctly answers *no*. On the other hand, there are many RTE problems where NatLog's precision works against it. For example, NatLog answers *no* to problem 71 because it cannot account for the insertion of *acts as* in the hypothesis. Fortunately, both the Stanford system and the hybrid system answer this problem correctly.

6 Conclusion

We do not claim natural logic to be a universal solution for NLI. Many important types of inference are not amenable to natural logic, including paraphrase (*Eve was let go* $\models$ *Eve lost her job*), verb alternation (*he drained the oil* $\models$ *the oil drained*), relation extraction (*Aho, a trader at UBS, ...* $\models$ *Aho works for UBS*), common-sense reasoning (*the sink overflowed* $\models$ *the floor got wet*), and so on.

Moreover, because natural logic has a weaker proof theory than FOL, some inferences lie beyond its deductive power. For example, it cannot explain inferences involving De Morgan's laws for quantifiers, as in *Not all birds fly* = *Some birds don't fly*.

However, by incorporating semantic containment, semantic exclusion, and implicativity, the model of natural logic developed in this paper succeeds in explaining a great variety of everyday patterns of inference. Ultimately, open-domain NLI is likely to require combining disparate reasoners, and a facility for natural logic inference is a good candidate to be a component of such a solution.

Acknowledgements The authors wish to thank the anonymous reviewers for their helpful comments. This work was supported in part by ARDA's Advanced Question Answering for Intelligence (AQUAINT) Program.

References

Akhmatova, Elena. 2005. Textual entailment resolution via atomic propositions. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bos, Johan and Katja Markert. 2006. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Böttner, Michael. 1988. A note on existential import. *Studia Logica*, 47(1):35–40.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

de Marneffe, Marie-Catherine, Bill MacCartney, Trond Grenager, Daniel Cer, Anna Rafferty, and Christopher D. Manning. 2006. Learning to distinguish valid textual entailments. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Fowler, Abraham, Bob Hauser, Daniel Hodges, Ian Niles, Adrian Novischi, and Jens Stephan. 2005. Applying COGEX to recognize textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Harabagiu, Sanda and Andrew Hickl. 2006. Using scenario knowledge in automatic question answering. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 32–39, Sydney.

Harmeling, Stefan. 2007. An extensible probabilistic transformation-based approach to the Third Recognizing Textual Entailment Challenge. In *ACL-07 Workshop on Textual Entailment and Paraphrasing*, Prague.

Hickl, Andrew, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with LCC's GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*.

Jijkoun, Valentin and Maarten de Rijke. 2005. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 73–76.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL-03*, Sapporo.

Lakoff, George. 1970. Linguistics and natural logic. *Synthese*, 22:151–271.

Levy, Roger and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of LREC-06*, Genoa.

MacCartney, Bill and Christopher D. Manning. 2007. Natural logic for textual inference. In *ACL-07 Workshop on Textual Entailment and Paraphrasing*, Prague.

MacCartney, Bill, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of NAACL-06*, New York.

Marsi, Erwin and Emiel Krahmer. 2005. Classification of semantic relations by humans and machines. In *ACL-05 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor.

Nairn, Rowan, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics)*, Buxton, UK.

Romano, Lorenza, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. 2006. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of EACL 2006*.

Sánchez Valencia, Victor. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

van Benthem, Johan. 1986. *Essays in logical semantics*. Reidel, Dordrecht.

van der Sandt, Rob A. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333–377.