Learning to distinguish valid textual entailments

Marie-Catherine de Marneffe, Bill MacCartney, Trond Grenager, Daniel Cer, Anna Rafferty and Christopher D. Manning
Computer Science Department
Stanford University
Stanford, CA 94305
{mcdm, wcmac, grenager, cerd, rafferty, manning}@cs.stanford.edu

Abstract

This paper proposes a new architecture for textual inference in which finding a good alignment is separated from evaluating entailment. Current approaches to semantic inference in question answering and textual entailment have approximated the entailment problem as that of computing the best alignment of the hypothesis to the text, using a locally decomposable matching score. While this formulation is adequate for representing local (word-level) phenomena such as synonymy, it is incapable of representing global interactions, such as that between verb negation and the addition/removal of qualifiers, which are often critical for determining entailment. We propose a pipelined approach where alignment is followed by a classification step, in which we extract features representing high-level characteristics of the entailment problem, and give the resulting feature vector to a statistical classifier trained on development data.

1 Introduction

In the area of textual inference, nearly all efforts have sought to extract the maximum mileage from quite limited semantic representations, as full and open-domain natural language understanding lies far beyond current capabilities. Some have used measures of semantic overlap (Jijkoun and de Rijke, 2005), but the more interesting work has converged on a graph-alignment approach, operating on semantic graphs derived from syntactic dependency parses, and using a locally-decomposable alignment score as a proxy for strength of entailment (Haghighi et al., 2005; Salvo Braz et al., 2005). We highlight here the fundamental semantic limitations of this approach, and advocate a multi-stage architecture that addresses these issues. The three key limitations of the graph matching formulation are an *assumption of monotonicity*, an *assumption of locality*, and a *confounding of alignment and evaluation of entailment*.

Assumption of monotonicity. Most previous algorithms assume upward monotonicity: If a good match is found with a part of the text, other parts are assumed not to affect its validity. But many entailment decisions are non-monotonic. Consider variants on ID 156 in table 1. Suppose the hypothesis were *Oil prices soar*. This would allow a perfect graph match, because the hypothesis is a subgraph of the text. However, this would be incorrect because it ignores the modal operator *could*. Consider also the alternate text *Energy analysts do not confirm oil prices could soar [. . .]*.[1]

Assumption of locality. Locality is needed to allow practical search, but many entailment decisions rely on global features of the alignment, and thus do not naturally factor by nodes and edges. To take just one example, dropping a restrictive modifier preserves entailment in a positive context, but not in a negative one: *Dogs barked loudly* $\models$ *Dogs barked*, but *No dogs barked loudly* $\not\models$ *No dogs barked*.

Confounding alignment and entailment determination. In a graph-matching system, since we are embedded in a search for the lowest cost alignment, the system, rather than recognizing a non-entailment, will choose an alternate alignment. In ID 35, a graph-matching system will get a non-entailment by making the matching cost very high between *the UN Security Council* and *the "threat"*. The likely result of that is that the object of the hypothesis will align with *the UN Security Council* at the end of the text, assuming that we allow the alignment to "break" arcs.[2] The lexical alignments are

[1] This is the same problem labeled and addressed as *context* in Tatu and Moldovan (2005).

[2] Robust systems need to allow matches with imperfect arc correspondence. For instance, given *Bill went to Lyons to study*

ID	Text	Hypothesis	Entailed
5*	Scientists have discovered that drinking tea protects against heart disease by improving the function of the artery walls.	Tea protects from some diseases.	yes
35	[. . .] Ahmadinejad attacked the “threat” to bring the issue of Iran’s nuclear activity to the UN Security Council by the US, France, Britain and Germany.	Ahmadinejad attacked the UN Security Council.	no
156	Energy analysts said oil prices could soar as high as $80 a barrel and drivers in the U.S. could soon be paying $3 a gallon for gasoline, if damage reports from oil companies bear bad news.	Oil prices surged.	no
256	Brian Brohm, the Louisville quarterback, threw for 368 yards and five touchdowns as the Cardinals beat visiting Oregon State 63-27.	The quarterback threw for 413 yards and three touchdowns, and then ran to the end zone two more times.	no
484	Sir Ian Blair, the Metropolitan Police Commissioner, said, last night, that his officers were ”playing out of their socks”, but admitted that they were ”racing against time” to track down the bombers.	Sir Ian Blair works for the Metropolitan Police.	yes
532	In all, Zerich bought $422 million worth of oil from Iraq, according to the Volcker committee.	Zerich bought oil from Iraq during the embargo.	no
646*	Tokyo’s High Court has rejected an appeal for compensation by 10 Chinese survivors of Japanese germ warfare experiments during World War II.	Tokyo’s High Court approves an appeal for compensation by 10 Chinese survivors.	no

Table 1: Illustrative examples from the Pascal RTE2 data set (IDs* come from the test set).

then perfect, and the only imperfect alignment is the object arc of *attacked*. A robust inference guesser will still likely conclude that there is entailment.

We propose that all three problems can be resolved in a multi-stage architecture, where the alignment phase is followed by a separate phase of entailment determination. Compared to previous work, we emphasize structural alignment, and seek to ignore issues like polarity and quantity, which can be left to a subsequent entailment decision: the scoring function is designed to encourage antonym matches, and ignore the negation of verb predicates. Given a good alignment, the determination of entailment reduces to a simple classification decision. The classifier can use hand-set weights, or it can be trained to minimize a relevant loss function using standard techniques from machine learning. The classifier is built over features designed to pattern valid and invalid inference. Because we already have a complete alignment, the classifier’s decision can be conditioned on arbitrary *global* features of the aligned graphs, and it can detect failures of monotonicity.

2 System

Our system has three stages: linguistic analysis, alignment, and entailment determination.

French farming practices, we would like to be able to conclude that *Bill studied French farming* despite the small structural mismatch.

2.1 Linguistic analysis

Our goal in this stage is to compute linguistic representations of the text and hypothesis that contain as much information as possible about their semantic content. We use *typed dependency graphs*, which contain a node for each word and labeled edges representing the grammatical relations between words.

Our approach is to parse the input sentences, and to convert the ouput to a typed dependency graph, using a set of deterministic hand-coded rules defining patterns over the phrase structure tree (Marneffe et al., 2006). We use the Stanford parser (Klein and Manning, 2003), a statistical syntactic parser trained on the Penn TreeBank. To ensure correct parsing, we preprocess the sentences to collapse named entities (identified by a CRF-based NER system) and collocations (derived from multiword expressions in WordNet (Fellbaum, 1998)) into new tokens. The nodes in the final dependency graph are annotated with their associated word, part-of-speech (given by the parser), lemma (given by a finite-state transducer described by Minnen et al. (2001)) and named-entity tag (given by the NER tagger).

2.2 Alignment

The purpose in the second stage is to find a good partial alignment between the graphs representing the hypothesis and the text. An alignment consists of a mapping from each node in the hypothesis graph to a single node in the text graph, or to null.[3]

We define a measure of alignment quality, and a procedure for identifying high scoring alignments. We choose a locally decomposable scoring function, such that the score of an alignment is the sum of the local node and edge alignment scores. We use an incremental beam search, combined with a node ordering heuristic, to do approximate global search in the large space of possible alignments. We have experimented with several alternative search techniques, and found that the solution quality is not very sensitive to the specific search procedure used.

Our scoring measure is designed to favor alignments which align semantically similar subgraphs, irrespective of polarity. For this reason, nodes receive high alignment scores when the words they represent are semantically similar. Synonyms and antonyms receive the highest score, and unrelated words receive the lowest. Our hand-crafted scoring metric takes into account the word, the lemma, and the part of speech, and searches for word relatedness using a range of external resources, including WordNet, precomputed latent semantic analysis matrices, and special-purpose gazettes. Alignment scores also incorporate local edge scores, which are based on the shape of the paths between nodes in the text graph which correspond to adjacent nodes in the hypothesis graph. Preserved edges receive the highest score, and longer paths receive lower scores.

2.3 Entailment determination

In the final stage of processing, we make a decision about whether or not the hypothesis is entailed by the text, conditioned on the typed dependency graphs, as well as the best alignment between them. Because we have a data set of examples that are labeled for entailment, we can use techniques from supervised machine learning to learn a statistical classifier. The class probabilities given by the classifier can be used to give confidence estimates for computing the average precision.

We use a logistic regression classifier with a Gaussian prior for regularization. The relatively small size of the training set can lead to overfitting problems. We address this by keeping the feature dimensionality small, and using high regularization penalties in training. As well as setting weights based on development data, we also have hand-set weights guided by linguistic intuition. A notable fact about our Pascal system is that using hand-set weights does not perform much worse than automatic weight setting. This is partly because the number of weights is modest, but also reflects that many of our parameters are for special purpose features that are sparsely exemplified in the development data, and which can easily receive completely wrong values (i.e., positive rather than negative weight) when fit to the limited development data.

3 Feature representation

In the last stage, the entailment problem is reduced to a representation as a vector of 54 features. These features try to capture salient patterns of entailment and non-entailment, with particular attention to contexts which reverse or block monotonicity, such as negations and quantifiers.

Polarity features. These features capture the presence (or absence) of linguistic markers of negative polarity contexts in both the text and the hypothesis, such as simple negation (*not*), downward-monotone quantifiers (*no*, *few*), restricting prepositions (*without*, *except*) and superlatives (*tallest*).

Antonymy features. Entailment problems might involve antonymy, as in ID 646. We check whether an aligned pair of text/hypothesis words appear to be antonymous by consulting a pre-computed list of about 40,000 antonymous and other contrasting pairs derived from WordNet. For each antonymous pair, we generate one of 3 boolean features, indicating whether the words appear in contexts of matching polarity, only the text word is in a negative-polarity context, or only the hypothesis word does.

Adjunct features. These indicate the dropping or adding of syntactic adjuncts when moving from the text to the hypothesis. In ID 532, the hypothesis aligns well with the text, but the addition of *during the embargo* indicates non-entailment. We identify the root node of the hypothesis graph and the aligned node in the text graph. Using dependency information, we verify whether adjuncts have been added or

[3]The limitations of using one-to-one alignments are mitigated by the fact that many multiword expressions (e.g. named entities, noun compounds, multiword prepositions) have been collapsed into single nodes during linguistic analysis.

dropped. We then determine the *polarity* of the roots (negative/positive context, or restrictor of a universal quantifier) to generate features accordingly.

Modality features. These capture simple patterns of modal reasoning, as in ID 156, which illustrates the heuristic that possibility does not entail actuality. According to the occurrence (or not) of predefined modality markers, such as *must* or *maybe*, we map the text and the hypothesis to one of six modalities: *possible*, *not possible*, *actual*, *not actual*, *necessary*, and *not necessary*. The text/hypothesis modality pair is then mapped into one of the following entailment judgments: *yes*, *weak yes*, *don't know*, *weak no*, or *no*. For example:

$$(\textit{not possible} \models \textit{not actual})? \quad \Rightarrow \quad \textit{yes}$$
$$(\textit{possible} \models \textit{necessary})? \quad \Rightarrow \quad \textit{weak no}$$

Factivity features. The context in which a verb phrase is embedded may carry semantic presuppositions giving rise to (non-)entailments such as *The gangster tried to escape* $\not\models$ *The gangster escaped*. Negation influences some patterns of entailment: *The gangster managed to escape* $\models$ *The gangster escaped* while *The gangster didn't manage to escape* $\not\models$ *The gangster escaped*. To capture these phenomena, we compiled small lists of factive, implicative and non-factive verbs, clustered according to the kinds of entailments they create. We determine to which class the parent of the text aligned with the hypothesis root belongs to. If the parent is not in the list, we only check whether the embedding text is an affirmative context or a negative one. This allows us to get right an example such as ID 5, even if the verb *discover* was negated.

Quantifier features. These features are designed to capture entailment relations among simple sentences involving quantification, such as *Every company must report* $\models$ *A company must report* (or *The company*, or *IBM*). No attempt is made to handle multiple quantifiers or scope ambiguities. Each quantifier found in an alignment is mapped into one of five categories: *no*, *some*, *many*, *most*, and *all*. The *some* category includes definite and indefinite determiners and small cardinal numbers. An ordering over the categories is defined. Features are generated given the categories of both hypothesis and text.

Algorithm	Dev Set		Test Set		Test Set*	
	Acc	AvP	Acc	AvP	Acc	AvP
Hand-set	67.00	72.50	57.63	61.31	58.25	61.36
Learning	66.87	74.83	60.50	58.00	60.50	58.44

Table 2: Accuracy and average precision on the RTE2 data sets.

Number, date, and time features. These are designed to recognize (mis-)matches between numbers, dates, and times, as in ID 256. We do some normalization of date expressions and represent numbers as ranges and expressions and have a limited ability to do expression matching. In ID 256, the mismatched numbers are correctly identified (*for 413 yards* vs. *368*, *three touchdowns* vs. *five*).

Structure features. These features aim to determine that the syntactic structures of the text and hypothesis do not match, as in ID 35 were the objects of the verb *attacked* are distinct. Some other features deal with specific structures of the hypothesis: *X works for Y*, *X is located in Y*. For such hypotheses, we search the text for noun complements of the aligned subject which are cues of entailment: in ID 484, *the Metropolitan Police Commissioner* is identified as an apposition to *Sir Ian Blair*.

Alignment features. Our feature representation includes three real-valued features intended to represent the quality of the alignment: *score* is the raw score returned from the alignment phase, while *goodscore* and *badscore* try to capture whether the alignment score is "good" or "bad" by computing the sigmoid function of the distance between the alignment score and hard-coded "good" and "bad" reference values.

Conjunction features. We also use conjunctions of features, namely structure and adjunct features are used in conjunction with the alignment features.

4 Experiments and results

Table 2 shows results for our system under both ways of setting the feature weights. "Hand-set" describes experiments with weights set by hand (but the threshold set to give balanced true/false answers on the test set). "Learning" describes experiments in which feature weights, including the threshold, are trained by a logistic regression classifier on devel-

	Hand-tuned		Learning	
Task	Acc	AvP	Acc	AvP
IE	51.50	52.62	52.50	50.46
IR	53.50	58.77	61.00	60.37
QA	56.00	59.87	58.50	53.21
SUM	69.50	77.03	70.00	76.27

Table 3: Accuracy and average precision split by task on RTE2 test set.

Feature class & condition		weight
Structure	specific struct. match	3.19
Conjunction	match & root poorly aligned	2.33
Date	date match	1.98
Alignment	good score	1.52
Modal	yes	0.84
Polarity	text & hyp same neg polarity	0.76
...	...	...
Polarity	text has neg marker	–0.45
Modal	no	–0.58
Adjunct	different polarity	–1.16
Alignment	bad score	–1.22
Quantifier	mismatch	–3.12
Structure	clear mismatch	–3.21
Date	date insert	–3.43

Table 4: Learned weights for selected features. Positive weights favor entailment. Weights near 0 are omitted. Based on training on the RTE1 dev1 set and RTE2 devset.

Features excluded	Accuracy
none	67.75
Alignment score features	63.88
Structure	64.12
Modal / Number	67.38
Quantifier	67.62
Polarity	67.75
Adjunct / Antonymy / Factivity	67.88

Table 5: Ablation studies on RTE2 dev set (hand-set weights).

opment data: we used the RTE1 dev1 set (only) and the RTE2 devset. Training on more data should in principle be good, but our impression was that the style of pairs in the RTE2 set was rather different (a lot more focused on checking an individual attribute or relation, such as *X wrote Y*). We hoped that adding in a little RTE1 data would help, but feared that adding in too much would be counterproductive. In retrospect this was the right decision. If we had included all the RTE1 development and test data, our accuracy with learned weights would have been 1% lower, and the average precision 3% lower. An apparent negative to the RTE2 development set is that there are a lot of groups of pairs on one topic (e.g., the 7 pairs on the date when Cyprus was divided). We suspect that this lack of independence between examples in the development set hinders effective machine learning of parameter weights. Table 4 shows the values learned for selected feature weights. As expected, the features *date insert*, *structure clear mismatch* indicate lack of entailment while *structure match*, *date match*, *modal yes* favor entailment. Surprisingly, *date modifier insert* also indicates entailment.

A bug in the handling of circular dependency graphs crashed the final version of our system on the test set, and we reverted to a version from a week earlier for the results we submitted. The last column of table 2 shows results for the final system with that bug fixed.

Our scores are considerably higher on the development set. For the machine learned weights, this reflects classic overfitting: the software jiggles the weights to get as many training items right as possible over a sparse feature representation. But the system with hand-set weights also does much better on the development set. This is not classic overfitting but rather issues of coverage and correctness. Where the matching patterns of existing features mishandled examples in the development set or where there were errors in quantity expression processing etc., to the extent that time was available, we tried to address these issues, boosting the development set performance. While some of these changes hopefully helped test set performance too, other issues inevitably arose in the test set.

Unlike last year, we did not use per-task optimization, feeling that exploitation of it is rather unrealistic with respect to developing robust, cross-task systems. Post-hoc testing shows that use of a task feature would have boosted the average precision by over 2%, but would not have improved accuracy.

Post hoc ablation studies on the RTE2 dev set reflect the usefulness of the features (see Table 5). For this data, alignment and structure features are very helpful, but more advanced semantic features (polarity, adjunct, antonym and factivity features) seem to have no or negative effect on performance.

5 Error analysis

This section provides error analysis on the RTE2 test set. IDs given as examples are incorrectly classified.

Lexical knowledge. A lot of examples require lexical knowledge that is beyond what is currently present in our system. We give here only a few examples of items requiring such lexical knowledge that could be easily modeled into rules (e.g., *X bought Y* $\models$ *Y belongs to X*):

75 T: Three days after PeopleSoft *bought* JD Edwards in June 2003, [...]
H: JD Edwards *belongs to* PeopleSoft.

218 T: "The C. & the S." is *the brainchild of* Dave McCool.
H: Dave McCool is *the inventor of* "The C. & the S.".

379 T: David McCool took the money and *decided to start* Muzzy Lane in 2002.
H: David McCool *is the founder of* Muzzy Lane.

250 T: Walter R. Mears, *a columnist for* The Associated Press [...]
H: Walter R. Mears *writes for* The Associated Press.

However some examples require too much inferential reasoning to be correctly handled:

477 T: President Bush said Miers is the most qualified candidate for the job, and Mrs. Bush agreed: "Absolutely. Absolutely."
H: Mrs. Bush supports Miers.

Alignment. Entailment determination relies on the alignment. We end up with a reliable alignment in most cases, except when numbers are involved. We then often align wrong numbers with respect to the structure of the sentence.

198 T: Some *420 people* have been hanged in Singapore since 1991, [...]. That gives the country of *4.4 million* people the highest execution rate in the world relative to population.
H: *4.4 million* people were executed in Singapore.

247 T: [...] in the assassination of the six Jesuits and their two maids, which took place at daybreak *on the 16th of November*, as reported by president Alfredo Cristiani *on the 7th of January*.
H: The assassination of the six Jesuits and their two maids took place *on the 7th of January*.

Numbers. In the handling of numbers, we lack computation. Numbers in italics in the following examples are aligned, and we report a mismatch.

10 T: This is good news for Gaelic translators, as the EU will have to churn out official documents in this language, in addition to the *20* other official EU languages.
H: There are *21* official EU languages.

389 T: In Rwanda there were on average *8,000* victims per day for about 100 days.
H: There were *800,000* victims of the massacres in Rwanda.

Structure. Structure features failed to capture some structure (mis-)matches:

15 T: A mercenary group [...] wounded and killed an interior ministry worker and wounded five others.
H: An interior ministry worker was killed by a mercenary group.

65 T: Nguyen's lawyer, Lex Lasry, told [...]
H: Nguyen is a lawyer.

Specific hypothesis features lead to incorrect classifications. Compare ID 87 (*Salvadoran politician Hector Colindres was kidnapped* $\models$ *Hector Colindres comes from El Salvador*) with ID 331:

331 T: A group of elders visited him after his brother, Vietnam *veteran* Dan Shermock, died in July 2004, Shermock said.
H: Dan Shermock comes from Vietnam.

Expansion of word lists. Many features rely on hand-made lists of words. The expansion of these lists, namely the list of non-factive verbs (IDs 134, 575) as well as the list of roots for specific hypotheses (IDs 16, 291), would allow use to correctly classify the following examples:

134 T: Opposition leaders in India have *called* on foreign minister Natwar Singh to resign [...]
H: Natwar Singh resigned.

575 T: Microsoft *denies* that it holds a monopoly.
H: Microsoft holds a monopoly power.

16 T: The British ambassador to Egypt, Derek Plumbly, [...]
H: Derek Plumbly *resides* in Egypt.

291 T: Japan's Kyodo news agency said [...]
H: The Kyodo news agency *is based* in Japan.

References

C. Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press.

A. Haghighi, A. Ng, and C. D. Manning. 2005. Robust textual inference via graph matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-05)*.

V. Jijkoun and M. de Rijke. 2005. Recognizing textual entailment using lexical similarity. In *Proceedings of the Pascal Challenge Workshop on Recognising Textual Entailment, 2005*, pages 73–76.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*.

M.-C. de Marneffe, B. MacCartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing in English. In *Natural Language Engineering*, volume 7(3), pages 207–233.

R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. An inference model for semantic entailment in natural language. In *Proceedings of the First Pascal Challenge Workshop on Recognizing Textual Entailment*, pages 29–32.

M. Tatu and D. Moldovan. 2005. A semantic approach to recognizing textual entailment. In *HLT/EMNLP 2005*, pages 371–378.