Aligning Semantic Graphs for Textual Inference and Machine Reading

Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Cer,
Daniel Ramage, Chloé Kiddon, Christopher D. Manning

{mcdm, grenager, wcmac, cerd, dramage, loeki, manning}@stanford.edu

Abstract

This paper presents our work on *textual inference* and situates it within the context of the larger goals of machine reading. The textual inference task is to determine if the meaning of one text can be inferred from the meaning of another and from background knowledge. Our system generates *semantic graphs* as a representation of the meaning of a text. This paper presents new results for aligning pairs of semantic graphs, and proposes the application of *natural logic* to derive inference decisions from those aligned pairs. We consider this work as first steps toward a system able to demonstrate broad-coverage text understanding and learning abilities.

Introduction

This paper outlines some of our recent work on the task of robust textual inference, wherein systems aim to determine whether a hypothesis text follows from another text and general background knowledge. In particular, it focuses on improving alignments between the two texts and applying ideas from natural logic to inference. But beyond that, it outlines ways in which such work relates to the more general goals of machine reading.

In order to understand texts, a machine reading system must provide (1) facilities for extracting meaning from natural language text, (2) a semantic representation language, for storing meanings internally, and (3) facilities for working with stored meanings, to answer questions or to derive further consequences. We also want such a system to be robust and open-domain, and to degrade gracefully in the presence of semantic representations which may be incomplete, inaccurate, or incomprehensible. Traditional knowledge representation & reasoning approaches (KR&R) fail in that respect because they use lambda calculus composition for meaning extraction, first order logic for meaning representation, and theorem provers for inference. At the other extreme, traditional information extraction (IE) systems go straight from input to output without any internal representation of semantics, often severely limiting types of semantic relations they can understand. An effective system must account for additional semantic relations such as equivalence, inference, and contradiction.

Textual inference work initially addresses the goal of machine reading by providing tools for producing a semantic representation from arbitrary text and for doing inference upon such representations. We present the *semantic graphs* we use as a representation for storing meanings of text that has been read, and examine approximate inference mechanisms based on alignment of semantic graphs and feature-based classification of proposed graph alignments. This approach enables weak but broad coverage inference.

Two points of departure of this work from machine reading are that machine reading emphasizes *unsupervised learning* and *synthesis* of information, neither of which we currently use very much. One answer is to say that there is more to machine reading than textual entailment, but we will also briefly outline how our work addresses the underlying issues by different means.

Our work focuses on the use of broad-coverage, supervised probabilistic components coupled with hand-built lexical knowledge sources such as WordNet (although they are supplemented by resources built using unsupervised methods, such as Latent Semantic Analysis). Rather than using unsupervised learning, this setup instead satisfies the goals of machine reading by being universal: structures like syntactic phrases and semantic roles can be applied to any text. They are not task and relation specific like IE frames.

Secondly, our system does no real knowledge synthesis, but rather does on-the-fly information acquisition and checking. This may seem a defect, but while looking on Google for the old quote *Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.* (Samuel Johnson, 1775, from Boswell's *Life of Johnson*), we came across an interesting discussion on a blog[1] about how what has happened in the world is that rather than having centralized human or machine knowledge management repositories, with global ontologies, what has been successful in the information age is people foraging for the information that they need when they need it on the web. The textual inference model is really about information foraging: how to supplement information retrieval with the understanding tools which will enable inferences to be

[1] *http://jeremy.zawodny.com/blog/archives/000765.html*

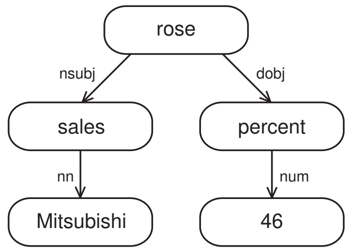

Figure 1: Typed dependency tree for "Mitsubishi sales rose 46 percent".

rose	→	fell
sales	→	sales
Mitsubishi	→	Mitsubishi_Motors_Corp.
percent	→	percent
46	→	46
Alignment score: -0.8962		

Figure 2: Alignment for the Mitsubishi example.

drawn from texts.

Textual inference system description

The textual inference task first appeared latent within the field of question answering (Pasca & Harabagiu 2001; Moldovan *et al.* 2003), and then received focus within the PASCAL Recognizing Textual Entailment (RTE) Challenge (Dagan, Glickman, & Magnini 2005; Bar-Haim *et al.* 2006), and related work within the U.S. Government AQUAINT program.

Our textual inference system employs a three-stage architecture in which alignment and entailment determination are two separate phases, preceded by a linguistic analysis. The alignment phase aims to assess the congruity between the hypothesis *H* and the text *T*, i.e., how well *H* can be embedded within *T*. Although early work on textual inference based the entailment decision solely on the quality of the alignment, we have found that the existence of negation, intensional contexts, and other common linguistic phenomena make alignment quality an unreliable indicator of inferrability. Consider the following hypothesis and text: *Arafat targeted for assassination* and *Sharon denies Arafat is targeted for assassination*. The hypothesis graph is completely embedded in the text graph, but it would be incorrect to conclude that there is entailment. To remedy this, in the third phase of our textual inference system we examine high-level semantic features of the proposed graph alignment, including indicators of such phenomena, to make the entailment decision.

Linguistic analysis phase. The goal of the first stage is to create for both text and hypothesis *semantic graphs*, which can be viewed as structured linguistic representations that contain as much information as possible about semantic content. As basis for the semantic graph, we use *typed dependency graphs*, in which each node is a word and labeled edges represent grammatical relations between words. Figure 1 gives the typed dependency graph for the sentence *Mitsubishi sales rose 46 percent*. The semantic graph for a sentence contains thus a node for each word of the sentence, each node being embellished with metadata generated by a toolkit of linguistic processing tools, including word lemmas, part of speech tags, canonicalization of quantitative expressions, and named entity recognition. The graph also contains labeled edges of multiple types. Chief among these are the directed edges representing the grammatical relations. These are derived using a set of deterministic handcoded rules defining patterns over the parser tree (de Marneffe, MacCartney, & Manning 2006), output by the Stanford parser (Klein & Manning 2003). To ensure correct parsing, we preprocess the sentences to collapse named entities and collocations into new tokens. Additional edges in the semantic graph represent imputed dependencies which result from "collapsing" pairs of surface-level dependencies; semantic relations derived from a semantic role labeling subsystem; and coreference links generated by a coreference module. Our code architecture ensures that each linguistic processing tool outputs a consistent analysis between the text and the hypothesis.

Alignment phase. In the second stage, our objective is to find an alignment from the hypothesis graph to the text graph which best represents the *support* of the hypothesis in the text, if any. The existence of a "good" alignment does not imply that the hypothesis is entailed; instead, the alignment is used as a piece of evidence upon which the entailment decision can be based. Formally, a (word-level) alignment is a map from the words in the hypothesis to the words in the text, or to *no word*, if the word has no support in the text. In our current system, we use one-to-one alignments.[2] Figure 2 gives the alignment for the following text-hypothesis pair:
T: Mitsubishi Motors Corp.'s new vehicle sales in the US fell 46 percent in June.
H: Mitsubishi sales rose 46 percent. (*FALSE*)

We define a measure of alignment quality, and a procedure for identifying high-scoring alignments. The different procedures we have explored are detailed below. The 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

[2]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.

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.

Entailment determination phase. The last stage consists in the determination of entailment, depending on the semantic graphs, as well as on the alignment between them. While a good alignment between semantic graphs is an important clue to the semantic relation between two sentences, it is not the whole story. Two sentences can exhibit a high degree of lexical and structural similarity, yet fail to be equivalent, entailing, or even consistent, due to the occurrence of negation, conditionals, modality, and many other semantic phenomena. Consider, for example:

(1) Hamas captured several hostages.

(2) Hamas took multiple prisoners.

(3) Hamas took no prisoners.

(4) Hamas took many prisoners, witnesses said.

(5) If Hamas took prisoners, Israel will retaliate.

Equipped with appropriate lexical resources, the alignment module will be able to find good alignments between the semantic graph for (1) and those for each of (2) through (5). But while (1) entails (2), it contradicts (3), and is compatible with either the truth or falsity of (4) and (5).

The last stage therefore aims to identify global features of such semantic phenomena. The entailment problem is reduced to a representation as a vector of 72 features. These features are meant to capture salient patterns of entailment and non-entailment, with a particular attention to contexts which reverse or block monotonicity, such as negations and quantifiers. We describe here some of the features. For further details, we refer the reader to (MacCartney *et al.* 2006).

We have polarity features which indicate the presence (or absence) of linguistic markers of negative polarity contexts in both semantic graphs, such as negation (*not*), downward-monotone quantifiers (*no*, *few*), restricting prepositions (*without*, *except*) and superlatives (*tallest*).

Other features deal with factive, implicative and non-factive verbs which carry semantic presuppositions giving rise to (non-)entailments such as *The gangster tried to escape* $\not\models$ *The gangster escaped*. In that context, negation influences some patterns of entailment and needs to be taken into account: *The gangster managed to escape* $\models$ *The gangster escaped* while *The gangster didn't manage to escape* $\not\models$ *The gangster escaped*.

Structure features aim to determine that the syntactic structure of the text and the hypothesis do not match, as in the following example: *Jacques Santer succeeded Jacques Delors as president of the European Commission in 1995* $\not\models$ *Jacques Delors succeeded Jacques Santer in the presidency of the European Commission*.

Some features recognize (mis-)matches between numbers, dates, times and persons. Our normalization of number and date expressions, and the inference rules on these, allow us to recognize that "more than 2,000" entails "100 or more": *More than 2,000 people lost their lives in the devastating Johnstown Flood* $\models$ *100 or more people lost their lives in a ferry sinking.*

We can use techniques from supervised machine learning to learn a statistical classifier as we have a data set of examples that are labeled for entailment. We use a logistic regression classifier with a Gaussian prior for regularization. The relatively small size of existing training sets can lead to overfitting problems. We address this by keeping the feature dimensionality small, and using high regularization penalties in training. As well as learning weights based on development data, we also have hand-set weights guided by linguistic intuition.

Improving alignment

Because the inference determination phase heavily relies on the alignment one, it is important to get the best possible alignments. In this section we report progress on three tasks we have undertaken to improve the alignment phase: (1) the construction of manually aligned data which enables automatic learning of alignment models, and effectively decouples the alignment and inference development efforts, (2) the development of new search procedures for finding high-quality alignments, and (3) the use of machine learning techniques to automatically learn the parameters of alignment scoring models.

Manual alignment annotation

Gold-standard alignments help us to evaluate and improve both parts of the system independently: we can use them in order to train and evaluate alignment models, and they can be used to evaluate how well the inference system performs when run using the manually assigned alignments rather than the automatically generated ones. We built a web-based tool[3] to facilitate hand annotation of alignments. In the tool, each hypothesis/text pair is displayed in a tabular format with each row corresponding to a hypothesis token and each column corresponding to a text token. The tokenization used is identical to that of the Penn Treebank except for the fact that spans of text that were identified as named entities are collapsed into a single phrasal chunk. Annotators are then able to express various relationships between a pair of hypothesis/text tokens by clicking on the corresponding table cell.

Supported relationships include the alignment of tokens with both *directional* and *bi-directional* semantics, as well as the alignment of antonyms. Pairs with *directional* semantics are used to capture the case where the semantics of one of the aligned tokens is intuitively a superset of the semantics of the other token in the pair. Examples[4] include pairs such as 'consumption'/'drinking', 'coronavirus'/'virus', and 'Royal Navy'/'British'. Pairs with

[3]A demonstration of the labeling tool is available at: http://josie:stanford.edu:8080/tableannotatorDemo/. As we believe that this tool should be generally useful for annotating paired texts, we plan on creating a publicly available distribution of it.

[4]Drawn from the RTE 2005 data set.

bi-directional semantics are tokens with synonymous semantics. This includes both cases were identical words and phrases in the text and the hypothesis can be aligned, and cases such as 'allow'/'grant', 'Dow'/'Dow Jones', and 'blood glucose'/'blood sugar' where the pairs have nearly synonymous meanings in context. However, while, by distinguishing between these different types of alignments, we are able to capture some limited semantics in the alignment process, the exploitation of this information is left to future work.

In the annotation process, we need to take into account both the lexical and structural levels of the sentences. Words which are semantically similar should be aligned, but there must be a trade-off between high lexical relatedness and syntactic structure. Determiners, adjectives, and numbers preceding a noun have to be aligned with the eventual determiner, adjective or number adjoined to the aligned noun. Aligning subgraphs is thus preferred to aligning words here and there in the graph. In the following example, even though *measures* is more lexically related to *measuring*, we will align it to *stretches* which is structurally related to *Milky Way*. We also align *2,000 light-years* with *100,000 light-years* which is where the contradiction lies: from the text, we will infer that the Milky Way measures 100,000 light-years and not 2,000.

T: The galaxy, measuring just 2,000 light-years across, is a fraction of the size of our own Milky Way, which stretches 100,000 light-years in diameter.

H: The Milky Way measures 2,000 light-years across.

Improving alignment search

If we want to automatically find "good" alignments, we will need both a formal scoring model which measures alignment quality as well as a search procedure for finding high scoring models. Formally, we define the score of the dataset $\mathcal{D}$ to be the sum of the scores of the individual alignments:

$$s(\mathcal{D}) = \sum_{(t,h,a)\in\mathcal{D}} s(t,h,a)$$

where h is the hypothesis and t is the text of a particular alignment problem, and a is the alignment between them. We then assume that the score $s(t,h,a)$ of an individual alignment decomposes into a sum of local scores, given by the scoring functions s_w for word pairs and s_e for edge-path pairs, as follows:

$$s(t,h,a) = \sum_{i\in h} s_w(h_i, a(i)) + \sum_{(i,j)\in e(h)} s_e((h_i,h_j),(a(h_i),a(h_j)))$$

where we use the notation $a(x)$ to refer to the word in the text which is aligned to the word x in the hypothesis under the alignment a, and $e(x)$ to refer to a function returning the set of edges in a hypothesis x. The first term is the sum of the scores of the alignments of the individual words, and the second term is the sum of the scores of the alignments of the pairs of words which are connected by an edge in the hypothesis graph.

The space of possible alignments is large: for a hypothesis with m words and a text with n words, there are $(n+1)^m$ possible alignments, making exhaustive search intractable. Informed search methods such as A* are also inefficient, since it is difficult to find heuristics which prune a significant part of the search space. Although exact search is infeasible, the search problem doesn't seem that hard. The bulk of the alignment score depends on local factors: the quality of the match between aligned words. As a consequence, we have found it easy in practice to find high-quality solutions using two approximate search techniques, *beam search* and *stochastic local search*, which we now explain.

The beam search technique is straightforward: at all steps in the search we keep at most k partial alignment candidates to explore. We begin by choosing an order in which the hypothesis words will be aligned. At each iteration of the search, we select the next hypothesis word to align, and for every partial alignment in the beam from the previous iteration, we try extending it with the current word in all possible ways. We score these new partial alignments and put them in a priority queue. Finally, we select the k partial alignments from the queue which have the highest scores, and put them into a new beam. We repeat this process for every hypothesis word, and at the end select the highest scoring alignment in the final beam.

The stochastic search technique operates instead on a *complete state formulation* of the search problem, and is based on *Gibbs sampling*, a well-known *Markov Chain Monte Carlo* technique. Our Gibbs sampler works as follows: we initialize the algorithm with the complete alignment which maximizes the greedy word pair scores, and we score it. Then, in each step of the search, we select a hypothesis word, and generate all possible alignments that result from aligning that word to a word in the passage. We score each of these alignments, and treating the scores as log probabilities, create a normalized distribution over these possible successors. We then sample a successor alignment from this distribution, and repeat. This Gibbs sampler is guaranteed to give us samples from the posterior distribution over alignments defined implicitly by the scoring function. However, we are interested in finding a maximum of the function, so we modify the basic Gibbs sampler with the *soft-max* function, parameterized by a temperature parameter which we can decrease according to a *cooling schedule*.

A comparison of the two search techniques shows that the stochastic search outperforms the beam search over a wide range of parameter values on the hand-set alignment weights. In table 1 we show the results for beam search of width 100 and stochastic search for 50 iterations on the RTE2_dev dataset. These runs are representative of beam widths ranging from 10 to 1500 and of stochastic searches with iterations ranging from 10 to 200.

Learning alignment models

In the previous section we defined a model for scoring candidate alignments, in which the scoring function is factored into the sum of scores of word alignments s_w and scores of edge alignments s_e. In previously published versions of the system we manually specified these scoring functions in a

	Correctly aligned	
	Individual words	Text/hypothesis pairs
Beam	4098	184
Stochastic	4260	202

Table 1: Results for beam search (width = 100) and stochastic search (50 iterations). In RTE2_dev, there are 5824 words and 800 text/hypothesis pairs.

	Correctly aligned	
	Individual words	Text/hypothesis pairs
Perceptron	4675	271
MIRA	4775	283

Table 2: Results for perceptron and MIRA algorithms on 10-fold cross-validation on RTE2_dev for 10 passes.

way which we believed reflected alignment quality. However, the existence of a gold standard alignment corpus described above enables the automatic learning of an alignment scoring function. More precisely, given a particular model form, it is possible to automatically select parameters which maximize some measure of performance. For both the word and edge scoring functions, we choose a linear model form in which the score is computed as the dot product of a feature vector and a weight vector:

$$s_w(h_i, t_j) = \theta_w \cdot \mathbf{f}(h_i, t_j), \text{and}$$

$$s_e((h_i, h_j), (t_k, t_\ell)) = \theta_e \cdot \mathbf{f}((h_i, h_j), (t_k, t_\ell)).$$

In selecting a learning algorithm we first must choose an objective function to minimize. We choose to minimize training set prediction error. Recent results in machine learning show the effectiveness of *online learning* algorithm for structure prediction tasks. Online algorithms iterate over the examples in the training set, and for each example they use the current weight vector to make a prediction. Then they compare the prediction to the "correct" label, and update the weight vector in a way that depends on this comparison. The *perceptron update* is very simple: when the prediction is incorrect, the weight vector is modified by adding a multiple of the difference between the feature vector of the correct label and the feature vector of the predicted label. When the prediction is correct, the weight vector is not modified. We use the adaptation of this algorithm to structure prediction, as first proposed by (Collins 2002). An alternative update rule is provided by the *MIRA update*, which attempts to make the minimal modification to the weight vector such that the score of the incorrect prediction (or predictions) for the example is lower than the score of the correct label (Crammer & Singer 2001). For this reason it is called an "ultraconservative algorithm".

We compare the performance of the perceptron and MIRA algorithms on 10-fold cross-validation on the RTE2_dev dataset. Both algorithms improve with each pass over the dataset. Most improvement is within the first five passes. Table 2 shows runs for both algorithms over 10 passes through the dataset. MIRA consistently outperforms perceptron learning.

Towards natural logic

As we saw, by itself, an alignment model cannot easily account for the impact of semantic operators such as negation and conditionals on the entailment relations between sentences, particularly when such operators are composed in complex structures. Of course, this is where formal logic shines—but translating natural language into logical representations suitable for formal reasoning is a highly brittle and error-prone process, as years of research have demonstrated. The advantage of semantic graphs as a representation of meaning is that they remain close to the original linguistic form, and make no pretension of formality or exactness. We'd like to be able to identify the logical relation between two aligned semantic graphs—at least in the most common cases—without paying the price of embracing full logical formality.

The aim is that given a complex sentence such as *Budget airline Ryanair has pushed ahead with its 1.48bn euro takeover offer for Aer Lingus, by upping its holding in its Irish rival*, we may not be able to translate it into a formal logic representation, but we should nevertheless be able to conclude that the sentence supports the hypothesis that *There is a takeover offer for Aer Lingus.*

Natural logic

In fact, much of the theoretical foundation for such an approach has already been worked out, under the heading of *natural logic*, defined by (Lakoff 1970) as "a logic for natural language". Unlike formal logic, natural logic does not involve manipulation of formal expressions, but operates directly on the words and sentences of our ordinary language. The study of natural logic was further developed by (Benthem 1986) and (Sanchez-Valencia 1991), but has been largely ignored in NLP. This is regrettable, because it is a natural fit to the problem of textual inference.

Natural logic focuses particularly on those familiar and widespread inferences involving *monotonicity*, which reason about the consequences of widening (weakening, generalizing) or narrowing (strengthening, specializing) the concepts or constraints involved in a proposition. In its simplest applications, the heuristic which underlies natural logic is that widening a concept or constraint preserves truth, while narrowing does not.

Crucially, however, natural logic also provides an account of the impact on such inferences of *monotonicity inverters*, which serve to reverse the usual heuristic. Inversions of monotonicity may be generated not only by negation, but also by universal quantifiers like *all*; by verbs like *lack*, *fail*, or *prohibit*, whose semantics contain an inherent element of negation or restriction; by prepositions such as *without* and *except*; by adverbs such as *only*; by comparatives and superlatives; and by the antecedent of a conditional. Additional complexities arise from the nesting of monotonicity inversions (*If no toxins leak*) and differences among monotonicity inverters with respect to where they have effect (e.g., *no* vs. *all*).

We are presently engaged in developing a computational model of natural logic. Our efforts follow those of (Sanchez-Valencia 1991), which defined a formal monotonicity calculus, including the outline of an algorithm for determining the entailments of a sentence in a bottom-up, compositional fashion. However, this approach assumed the use of highly formal Lambek calculus representations as inputs; whereas we aim to build a system operating over the semantic graph representations described earlier. We are also working to extend (Sanchez-Valencia 1991) in important respects: for example, by accounting for antonyms and exclusions, and factive and non-factive verbs.

While our full natural logic system is still under development, many of the features used in the entailment determination phase of the current system are partial theories which capture natural logic heuristics. We have recently introduced several new features which extend this theme.

Lexical entailment features

At the base of a natural logic system, we need a strong lexical entailment model. If relationships between words are known, then the relationships between more complicated semantic structures, such as phrases, can be more effectively determined. Previously, alignments between words were given scores based on broad similarity measures. However, these methods just calculated naive association and disregarded the direction of entailment. In order to start making the transition to a more natural logic friendly system, where monotonicity relations between words are crucial, we have begun the foundations of a true word-word entailment system. Previously constructed features deal with synonymy and antonymy relations; these new feature being extracted take on the tasks of hypernymy relations, geographic location relations, and relations between gradable adjectives. Currently, all lexical features are being extracted from relationships in WordNet (Fellbaum 1998).

Hypernymy relations. To add hypernymy features, we first attempt to determine whether a sentence displays an upward-monotone (*positive*) or downward-monotone (*negative*) context. Then we extract hypernym/hyponym relationships from WordNet for aligned words in order to identify entailments. In upward-monotone contexts (the default), broadening a concept by replacing it with a hypernym is a valid entailment, while narrowing a concept is not. While in downward-monotone contexts, the reverse is true. The following example is not a valid entailment since in this positive context, *chronic disease* cannot be replaced by its hyponym *osteoporosis*:
T: [...] fruits and vegetables can reduce the risk of chronic disease.
H: Fruits and vegetables prevent osteoporosis. (*FALSE*)

Location relations. There are arguments in the textual entailment community on how much world knowledge should have to be known by the system. However, it seems that geographic knowledge falls under the category of common knowledge and should be covered by a lexical entailment system. For location features, holonymy relations in WordNet are extracted for locations and the directionality of entailment is preserved. If a location is part of another location, then it entails the outer location. In this instance, *Paris* is aligned to *France*, and it is determined to be a good entailment since Paris is a city in France. Since extracting holonym relations from WordNet returns chains of holonyms, *Paris* could have been aligned to *Europe* in the hypothesis and the lexical entailment would still hold.
T: The Louvre Museum in Paris opened a new individual sponsorship program in July.
H: The Louvre Museum is located in France. (*TRUE*)

Adjective gradation. Gradable adjectives in WordNet are associated to other adjectives along their gradient by the same "Similar-to" relation, without regard for which way along the gradient the relationship occurs. Adjective gradation features allow for a more finely-tuned and directional measure of adjective-adjective entailments. Intensity relationships were collected between adjectives in WordNet by applying a number of surface patterns on the adjective synset glosses of "Similar-to" adjectives. Intensity relations were also collected from relations between adjectives and their comparative and superlative forms. If an adjective is higher on the gradient than another, such as *scorching* to *hot*, then the higher adjective can entail the other adjective. However the opposite is not a valid entailment. In this instance, *high* in the text would be aligned to *highest* in the hypothesis since the two words have a high similarity score. However, this would not constitute an entailment since *highest* is a more intense variant of *high*.
T: For a western European country, the birth rate in Finland is high.
H: Finland is the European country with the highest birth rate. (*FALSE*)

Conclusion

This paper presents progress on our system for textual inference, which decouples alignment and inference stages in deciding whether one text follows from another. We have shown improvement in both stages and have outlined future research directions that might lead to systems capable of better text understanding.

We believe our system's underlying graph-based semantic representation is a reasonable place to start when building a system able to understand open-domain text. Our system embraces universality by coupling components that are not domain-specific. And when combined with inference mechanisms, this representation is a step toward the deep text understanding called for by the vision of machine reading.

References

Bar-Haim, R.; Dagan, I.; Dolan, B.; Ferro, L.; Giampiccolo, D.; Magnini, B.; and Szpektor, I. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Benthem, J. V. 1986. *Essays in logical semantics*. Dordrecht: Reidel.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP-2002*.

Crammer, K., and Singer, Y. 2001. Ultraconservative online algorithms for multiclass problems. In *Proceedings of COLT-2001*.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005*.

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

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

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

Lakoff, G. 1970. Linguistics and natural logic. In *Synthese 22*, 151–271.

MacCartney, B.; Grenager, T.; de Marneffe, M.-C.; Cer, D.; and Manning, C. D. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the North American Association for Computational Linguistics*.

Moldovan, D.; Clark, C.; Harabagiu, S.; and Maiorano, S. 2003. COGEX: A logic prover for question answering. In *NAACL 3*.

Pasca, M., and Harabagiu, S. 2001. High performance question/answering. In *SIGIR '01*, 366–374.

Sanchez-Valencia, V. 1991. *Studies on Natural Logic and Categorial Grammar*. University of Amsterdam dissertation.