WIEN: Wordwise Inference and Entailment Now
Or: How We Taught Machines to Recognize Natural Language Inference

Chris Billovits, Mihail Eric, Chris Guthrie
{cjbillov, meric, guthriec}@stanford.edu

June 10, 2015

Abstract

The problem of inferring textual entailment relations is a fundamental challenge in natural language understanding. Building systems with the ability to recognize entailment relationships across sentences is a crucial step in achieving complete machine-level semantic understanding. We propose a multi-label classification model, implementing a random forest classifier with a carefully engineered and selected collection of linguistic and semantic features, to tackle this problem. Our system obtains an optimal F1 score of 80.9% on the SemEval-2014 SICK dataset.

1 Introduction

The advent of such commercial natural language interfaces as Siri, Google Now, and Cortana has made it clear that full semantic language understanding is one of the great contemporary problems of artificial intelligence. Developing a system with this power is immensely difficult because of the complexity of human text. A system with full semantic understanding would need to be capable of handling all the tricky aspects of language including named entity disambiguation, coreference resolution, and relation extraction. Moreover, it would need to have a grounded knowledge base of the world, being able interpret common sense facts and draw logical inferences among linguistic statements. Recognizing textual entailment relations is a cornerstone problem in the path to achieving true semantic understanding. Given two sentences S_1 and S_2, this problem seeks to determine the relationship between the two statements. Common classes of relationships include entailment (S_2 logically follows from S_1), contradiction (S_2 is not logically consistent with S_1), and neutrality (S_2 is logically independent of S_1).

This task is performed with ease by humans, who have years of practice as agents in real world scenarios and processors of common sense knowledge, but is one of the hardest problems in automated natural language understanding. For a machine to recognize textual entailment well, it has to have a nuanced understanding of the sentences and be able to figure out both general semantics (with reference to world knowledge) and also the particular logical structure of the sentences (so that, for example, the system can tell that the addition of the word "no" takes the sentence pair from entailment to contradiction). For this work, we develop a machine-learning based model using multi-class classification with a linguistic and semantic feature set to achieve competitive performance on this task.

The remainder of our paper is structured as follows: Section 2 outlines previous work on recognizing textual entailment, Section 3 explains the methodology of our approach, Section 4 discusses our explored feature set in detail, Section 5 gives the performance of our system, Section 6 discusses further improvements that can be made to enhance our model, Section 7 outlines the insights learned from developing this system, and Section 8 gives concluding thoughts and ideas for further exploration.

2 Background

We can divide the approaches taken by prior work on recognizing textual entailment into two main categories – *statistical* approaches and *rule-based* approaches. By statistical, we mean those approaches that probabilistically model patterns of entailment from large corpora. By rule-based, we mean those approaches that seek to determine entailment by understanding the underlying natural logic relationships between sentences. While these approaches are often combined to some extent, most of the research we evaluated has a strong focus on one over the other.

An interesting note: in relation to the three levels of analysis suggested by [1] (surface, syntactic, semantic), the vast majority of approaches we analyzed (including the rule-based approaches) stayed at the surface level, with only brief forays into syntactic and semantic levels (e.g. MacCartney 2009's [8] use of semantic composition trees).

Within the statistical category of approaches, the two key choices to be made are the model's feature set and the type of machine learning employed. [9] summarizes various combinations of these approaches as they were used in the SemEval-2014 challenge.

Potential features to include (as outlined in [5]):

- Output from standard natural language processing steps, such as stemming and POS tagging
- One or more similarity metrics between sentences
- Negation features (or other syntactic features) within sentences
- Normalization/re-weighting of sentences

Within the rule-based category of approaches, we found that most research has attempted to determine entailment by establishing an optimal set of edits to transform the premise into the hypothesis, using an alignment process such as MANLI [7]. Once this transformation has been established, the particular set of edits can by analyzed via a rule-based calculus, grounded in *a priori* principles of natural logic, to determine if the hypothesis likely follows from the premise. This is the approach used with strong results by MacCartney et al.'s NatLog system in [8].

While the NatLog system has a predetermined calculus for determining entailment (so that, for example, the abstract effects of negation are hard-coded into the system), some research, such as [12], has attempted to learn the effects of semantic relations. That is, these systems still use an edit/transformation model, but make no *a priori* guess as to the effects of these transformations, choosing instead to learn these effects from established corpora.

Even more flexibly, Bowman et al. [4] have used a recursive neural network architecture to learn rules for the entire process, starting only with representations of the premise and hypothesis statements.

3 Methodology

The SICK (Sentences Including Compositional Knowledge) data set consists of supervised pairs of sentences, which are classified as entailment, contradiction, or neutral. It was developed in order to test compositional knowledge, while omitting the need for encyclopedic knowledge. Specifically, the challenge was constructed to cater to compositional distributional semantic models (CDSM). The dataset has train/dev/splits consisting of 4500, 500, and 4927 sentence pairs respectively.

Our approach sought to combine a broad set of linguistic and *ad hoc* features in a similar vein and spirit to the entrants of the SemEval-2014 challenge. Training and testing was performed on the SICK data sets in the conventional train, dev, test splits, with a gold syntactic parse as generated in Bowman et al. [4]. A summary table of methods used on SICK is included from [9], and we broadly describe our contributions in the context of these methods. To simulate a submission, we abstained from examining model and feature decisions of SemEval2014 submissions [9] when developing and testing our methodology. Similarly, we constrained ourselves to five runs on the test set, as per the entry rules for SemEval.

Our approach to creating feature sets were based jointly on semantic sentence level information and Markovized language models. The former seeks to generate phrase-level sentence denotations, and compares the denotations to generate dense clusters of features. Word vectors were incorporated to form a sentence-level CDSM. The language models seek to broaden coverage by comparing subsets of unstructured sentences. WordNet and FrameNet were used as sources of auxiliary data. Fifty-dimensional GloVe vectors [10] trained over the 2014 Wikipedia and GigaWord corpus were sourced from Stanford NLP.

We employed a pipeline to generate a sparse feature set, select the most effective subset of features, and train an ensemble classifier with ten-fold cross-validation. The source code is available on GitHub at http://github.com/mihail911/224UProject/.

3.1 Pipeline Setup

Our pipeline permitted a wide variety of supervised learners. Due to the relatively small size of the data, and potential non-linearity of the problem, support vector machines (SVM) and kernel methods appear to be the most flexible choice. Indeed, 12 / 18 entrants to SemEval2014 used an SVM or non-linear kernel methods. Ensemble classifiers built upon decision trees can also learn non-linear objectives, and have the added bonus that they can be trained largely in parallel. Two out of 18 submissions used random forest classifiers for the entailment task with which we are concerned.

We followed the latter approach, and trained an ensemble random forest classifier, with an entropy-gain objective criteria. Random thresholds for decision boundaries in feature subsets were optionally employed. In addition, we built a one-versus-rest binary logistic regression model trained with L-BFGS to assess the impact of data non-linearity. Hyperparameters were determined with a grid search over pipeline elements, using ten-fold cross-validation. All classifiers used the standard implementation given in Python's scikit-learn module.

Feature selection was used as a pre-processing step, achieved through either univariate chi-squared tests or model-specific selection. In model-based approaches, unregularized logistic regression was used to prune features recursively. In practice, our optimal performing models selected the 5000 best features using model-specific selection.

4 Feature Descriptions

4.1 FrameNet

FrameNet [2] aims to establish a universal semantic framework which describes the images – or frames – in a sentence. For example, "the horse who ate the hay raced too slowly" might include frames like "racing" or "eating" (ate) or "moving" (raced). There are 1200 of these frames in the FrameNet database. Philosophically, FrameNet is based on the notion that language is best viewed as an attempt by the writer to put an image into the reader's head via a limited number of imprecise archetypal images shared by the writer and the reader (the frames), rather than through the exact logic of the language. These frames are arranged in a hierarchy, like WordNet's, with more general frames, like "moving", appearing above more specific frames, like "racing". A frame-semantic parser, then, aims to map a sentence to a set of frames that are meant to be evoked by the sentence. It also ensures that the frames include details about the subjects and objects in the frame – e.g. that it is a "horse" eating "hay" in the "eating" frame.

We used the Carnegie Mellon SEMAFOR frame semantic parser [11] out of the box for our parsing task.

Frame Overlap The frame overlap feature encodes the specific frame types that are shared between the two sentences. For each frame type (e.g. "eating") that is shared, we add a feature specific for that type. We also add a feature measuring the total number of frames shared between the two sentences.

Frame Entailment Semantic frames already induce a precise notion of entailment, through a predefined FrameNet hierarchy, corresponding closely to the entailment task generalized to sentences. Frame entailment captures frames like "running" –¿ "moving," creating a frame entailment feature for the more general frame each time such a relation is discovered.

Frame Alignment For each frame type that occurs in both sentences, we measure roughly the degree to which the frames refer to the same subjects and objects. These subjects and objects are formalized within the FrameNet system as semantic "labels" and are parsed explicitly by the SEMAFOR parser. The alignment score compares corresponding labels between frames of the same type, computing the percentage of labels that match (after lemmatization). For example, "the horse ate the hay" versus "the horse ate the oats" should have an "eating" frame alignment score of 1/2, since "horse" is the same between the "eating" frames but "hay" is different from "oats." For each frame we output the alignment score, and for each sentence pair we output the average, maximum, and minimum alignment scores across all frame matches.

4.2 GloVe Vectors

. GloVe distributional vectors seek to formally learn word weights such that for any two words, the dot product of two vectors approximates the log-probability of co-occurrence. Thus, GloVe vectors encode vector difference as a log-ratio of occurrence [10]. Since words of similar co-occurrence scenarios, but different frequencies are likely to suggest entailment, it follows that employing the difference of glove vectors is likely to suggest a direction of entailment.

Using this intuition, we developed two feature templates:

Sentence Difference. A sentence vector is built as the mean of all the words in the sentence, weighted by the depth of each word in the syntactic parse tree. The difference between the first and second sentences becomes a set of *ndim* features, with *ndim* being the size of the word vectors. Emphasizing the leaves of the tree is effective because complex noun and verb phrases are conventionally deeper within a sentence structure, whereas shallower words have less of an impact on the nuance of a topic. This feature was the most powerful compositional feature, leading to 0.68 F1 on the development set with only 50 features. Thus, GLoVe difference is nearly as powerful as word cross product, but with a few orders of magnitude fewer features.

Cosine distance. This template computes the

word vector cosine over the Cartesian product of words in each sentence. Feature buckets delineate the cosine difference of words by the depth at which each word occurs. Cosine distance emitted 209 features, and overall had intermediate performance – by itself, it achieved 0.63 F1 on the development set.

Interestingly enough, larger GLoVe vectors trained over a given corpora did not improve the performance of either GLoVe-based feature template.

4.3 Language Model Features

We compute overlap of n-grams in each sentence, returning a feature for each word common to both sentences. We also compute the Cartesian product of all words in the sentence, generating a feature of each pairwise word. Our model uses unigrams, bigrams, and trigrams for direct overlap, and unigrams and bigrams for cross product. None of these features by themselves is meant to be particularly accurate, but together they cast a wide net that could potentially be useful. We rely on feature selection criteria to weed out gram pairings that have no statistical significance. Cross-product features account for over 90% of features generated.

For unigram word overlap only, we also emit the number of overlaps that occur, as well as the disjoint lengths of each sentence. These extra statistics would have been immediately useful for the semantic relatedness subtask of SemEval14, but also was useful for entailment. When trained with a logistic regression model, these three features slightly outperformed the other $\sim$1400 word overlap features, but combined with word overlap they performed significantly better.

Feature selection was robust enough that all WordNet features that incorporated word level composition were a subset of the effective language model features. Thus, while we saw reasonable results with discretized (binary) features based off of WordNet relations, they were not better than the suite of language model features, outside of a very narrow set of features selected (< 100). N-gram overlap and cross-product provide a large set of sparse features that adequately cover sets of linguistic word features. For this reason, we only successfully incorporated a broad hypernym feature.

Negation. Our negation feature tests for the inclusion of key negation words occurring in only one of the sentences in a pair. Specifically, a feature is given for each of the words 'no', 'not', 'nobody', and 'none' if they appear in one sentence but not the other. Our sentence parse treated the contraction "n't" as a separate word, so we also tested for "n't".

Hypernym. WordNet is used to extract a set of synsets across the entire sentence that have the same part-of-speech tag as the word in the sentence. The part of speech tag is determined over the entire sentence by the NLTK tagger. For example, if "run" is tagged as a noun, then only the noun synsets of "run" are extracted. We count the size of the set intersection of hypernyms of synsets of the second sentence and synsets of the first sentence. This count is binarized, and placed into a bin with the number of synsets present in each sentence. In effect, the binarization serves to distinguish the ratio of hypernym overlaps by considering the range of word senses. Nominally, this prevents an overreaction to a single hypernym occurrence when there are many possible word senses.

5 Results

The SemEval challenge permitted entrants up to five runs on the test portion of the dataset. During our five permitted runs, we trained and performed ten fold cross-validation on a combined dataset consisting of the train and dev splits. Our final model system achieved a best F1 of 80.9% on the test split across the five runs. Had we participated in the actual competition, this would have ranked our system sixth out of eighteen entrants. We include the results of the top eight entrants on the competition below for reference with our system bolded and ranked appropriately:

Team ID	F1 Accuracy
Illinois-LH_run1	84.575
ECNU_run1	83.641
UNAL-NLP_run1	83.053
SemantiKLUE_run1	82.322
The_Meaning_Factory_run1	81.591
Cardinal_WIEN_run5	**80.948**
CECL_ALL_run1	79.988
BUAP_run1	79.663

To test the power of each of our individual feature classes, we performed an ablation study as follows: we did a single run for each feature class, building a system with only that feature class and word overlap enabled, performing ten-fold cross-validation on the train set and then testing on the dev set. Due to the large discrepancy in emitted features between the cross product features and everything else, runs including a cross product features were limited to 5000 features, to give an accurate sense of their potential contribution. Results of the study are given in Table 1.

We also did one run with all of the features enabled, cross-validating on train and dev combined, and testing

on the test set.

The results for this ablation study are included below along with the result of a most-common label baseline. We report F1 on three classification labels as well as the combined F1 (denoted E.F1, C.F1, N.F1, and T.F1):

As we expected with the test of each single feature, GloVe sentence difference performed very well, even with word overlap. GloVe cosine difference, on the other hand, provided mediocre increases in F1. One important difference between the two GloVe features is that GloVe cosine weighted word distance by absolute compositional depth, whereas sentence difference did not have a notion of whether a sentence parse was "unusually" deep. The former feature became considerably more sparse for word differences in either sentence at depths 10 and larger. The latter, in computing a weighted mean word vector, considered relative depth - if many words in the sentence are deep, then any single word at a given depth has less influence on the sentence composition vector.

The higher n-gram overlaps and cross products performed worse than their lower n-gram counterparts. We suspect that this is due to increased sparsity of feature emission, and a lack of reweighting. Furthermore, for the entailment task, higher order n-grams may simply not present any new information that couldn't be learned through feature weights on lower n-grams. Although 4-gram overlap occurs in the train set rather often, it did not contribute positively to our results during development when unigram, bigram, and trigram overlap features were already included.

The FrameNet features yields a large boost in F1, of nearly 3 points each. Strangely enough, the Frame Entailment feature was the best marker of the three in determining contradiction. Hypernym features were broad, but only changed the results marginally.

Despite the brittleness of our context-less negation feature, it performed admirably at distinguishing cases of contradiction and neutral, achieving 80 F1 on both. We think that its performance is more indicative of fundamental flaws in the SICK data set, discussed further in section 7.

6 Discussion

The most interesting outcome of the project was the discovery of two helpful features: the GloVe vector sentence difference and the overlap of parsed semantic frames from FrameNet. Our work demonstrates that they can be effective in recognizing textual entailment.

FrameNet's success as a group of features in our model points to the general effectiveness of frame semantic analysis. Universal frames are a precise way to draw semantic parallels between sentences without relying on any sort of imprecise similarity metrics. Frame semantic parsers can find common ground between sentences that other parsers might struggle with due to word ambiguities, differing specificity in words (e.g. "sculpting" as a type of "making"), multi-word phrases (e.g. capturing the relatedness between "visiting the mall" and "going shopping"), or imprecise grammar.

The success of GloVe sentence difference features shows that composing word-based GLoVe vectors learns sentence entailment on a deeper level than just the words. Surprisingly, sentence difference is the best singular feature template at distinguishing neutral sentence pairs. One possible interpretation is that constituency-weighted sentence-level composition creates a quasi-topic for the sentence, where dissimilar nouns and verbs tend to separate the topic. Its performance relative to the GloVe cosine feature set implies that absolute constituency tree depth does not provide a useful guide to sentence-level entailment. Overall, GLoVe's success with standard machine-learning classifiers suggests that the information included in distributional semantic vectors is useful for determining entailment in a general sense.

We found that another significant feature was the presence of negation keywords. This is a weak and brittle feature, that likely was only effective because of the idiosyncrasies and monotonicity of the SICK dataset. The top performing submission from the University of Illinois, Urbana-Champaign [6] reported a similar outcome from their negation feature. This also calls into question how generalizable results on the SICK dataset are to other datasets and the general problem of recognizing textual entailment. In other words, our good performance on this dataset may not necessarily entail that we really have a system that has "solved" natural inference. The power of the unigram cross product feature, which single-handedly achieved the maximum gain in total F1 for the ablation study, speaks to the relative homogeneity of the SICK dataset. This becomes quite clear if we actually look at sentence pairs in the train data. Very often the two sentences are literally identical in structure except for a single word synonym or a single word lacking. Consider, for example the following sentences taken from the train dataset: *A woman is wearing an Egyptian headdress* vs. *A woman is wearing an Indian headdress* or *Two people are kickboxing and spectators are not watching* vs. *Two people are kickboxing and spectators are watching.*

Table 1: Ablation Study

Feature Class	C.F1	E.F1	N.F1	T.F1
Mod. Unigram Overlap	0.67626	0.57244	0.81315	0.72356
+ Frame Overlap	0.69173	0.63636	0.83993	0.75937
+ Frame Similarity	0.67176	0.62976	0.83793	0.75338
+ Frame Entailment	0.70677	0.62284	0.83045	0.75235
+ Negation Keywords	0.80303	0.63481	0.80000	0.75287
+ Bigram Overlap	0.67669	0.61268	0.83705	0.74870
+ Trigram Overlap	0.67143	0.59498	0.82960	0.73862
+ Unigram Cross Product	0.80000	0.68667	0.82456	0.78121
+ Bigram Cross Product.	0.75200	0.58571	0.81008	0.73687
+ Glove Vec Sentence Diff.	0.70769	0.62731	0.84474	0.76184
+ Glove Vec Cosine	0.69767	0.59498	0.82432	0.73953
+ Hypernym	0.68182	0.57565	0.82412	0.73150
All Combined (Test)	0.82881	0.72681	0.84635	0.80948
Most Common Base.	0.000	0.000	0.76000	0.4100

The relatively high frequency of such sentence pairs shows the lack of lexical and semantic diversity of the SICK dataset, making it so that a few carefully chosen and admittedly simplistic features can learn substantive regularities of the data.

7 Further Improvements and Future Work

We could generally improve the performance of most our features by first processing the text with rich off-the-shelf NLP pipeline tools, such as NER tagging or sentiment analysis. There are many natural language processing tools available that could have aided in developing phrase-based composition. We built a greedy word alignment baseline, aided by part-of-speech tagging and WordNet synset similarity measures. The guiding intuition behind building a greedy alignment model is that the inherent sentence topic similarity in the corpus made it fundamentally easier than general phrase alignment. A more robust semantic alignment algorithm, such as the MANLI algorithm [7] or a general alignment model may overcome these limitations. Such a semantic alignment algorithm would also open the door for nuanced natural logic features, such as those used in NatLog.

Many of our features could be creatively improved upon using word-order heuristics – for example, improving the negation feature by comparing the similarity of the words being negated. A similar improvement could come from comparing the similarity of the sentences being negated. This conjunction of the negation feature and word cross-product features would help the classifier make more sensitive negation choices. Further, a re-examination of WordNet features that compose words into phrases in novel manners, rather than firing indicator features, may be fruitful.

Future work should extend both of the significant feature areas we dipped into; that is, measures based on distributional word vectors and measures based on frame-semantic analysis. In particular, it may be interesting to use phrase alignment tools such as the Berkeley aligner to align phrases across sentences and construct compositional vectors using these alignments. Word vectors and FrameNet show a lot of promise for doing more detailed semantic analysis and parsing of utterances, ones that can bring us closer to building systems with complete natural language understanding. In addition, we hope to test on more linguistically rich inference datasets. As of this writing, we are in the process of running our system on the newly created Stanford Natural Language Inference dataset [3].

8 Conclusion

In this project, we tackled the problem of having a machine recognize textual entailment, building a statistical classifier to learn three classes of inference relationships. We developed a diverse collection of linguistic and semantic features, some of which, to the best of our knowledge, have never been tried before in existing natural language inference systems. We achieved competitive performance on the SICK 2014 dataset, one of the canonical entailment datasets in the research community.

References

[1] Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. volume 38, pages 135–187, USA. AI Access Foundation.

[2] Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

[3] Bowman, S., Angeli, G., Potts, C., and Manning, C. (2015). Learning natural language inference from a large annotated corpus. In Review for EMNLP 2015.

[4] Bowman, S. R., Potts, C., and Manning, C. D. (2014). Recursive neural networks for learning logical semantics. *CoRR*, abs/1406.1827.

[5] Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In Quionero-Candela, J., Dagan, I., Magnini, B., and dAlch Buc, F., editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer Berlin Heidelberg.

[6] Lai, A. and Hockenmaier, J. (2014). Illinois-lh: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

[7] MacCartney, B., Galley, M., and Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 802–811, Stroudsburg, PA, USA. Association for Computational Linguistics.

[8] MacCartney, B. and Manning, C. D. (2009). An extended model of natural logic. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 140–156, Stroudsburg, PA, USA. Association for Computational Linguistics.

[9] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8. Association for Computational Linguistics.

[10] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

[11] Tsatsaronis, G., Varlamis, I., and Nørvåg, K. (2012). Semafor: Semantic document indexing using semantic forests. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1692–1696, New York, NY, USA. ACM.

[12] Watanabe, Y., Mizuno, J., Nichols, E., Okazaki, N., and Inui, K. (2012). A latent discriminative model for compositional entailment relation recognition using natural logic. In *Proceedings of COLING 2012*, pages 2805–2820. The COLING 2012 Organizing Committee.