CMU-AT: Semantic Distance and Background Knowledge for Identifying Semantic Relations

Alicia Tribble
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
atribble@cs.cmu.edu

Scott E. Fahlman
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
sef@cs.cmu.edu

Abstract

This system uses a background knowledge base to identify semantic relations between base noun phrases in English text, as evaluated in SemEval 2007, Task 4. Training data for each relation is converted to statements in the Scone Knowledge Representation Language. At testing time a new Scone statement is created for the sentence under scrutiny, and presence or absence of a relation is calculated by comparing the total semantic distance between the new statement and all positive examples to the total distance between the new statement and all negative examples.

1 Introduction

This paper introduces a knowledge-based approach to the task of semantic relation classification, as evaluated in SemEval 2007, Task 4: "Classifying Relations Between Nominals". In Task 4, a full sentence is presented to the system, along with the WordNet sense keys for two noun phrases which appear there and the name of a semantic relation (e.g. "cause-effect"). The system should return "true" if a person reading the sentence would conclude that the relation holds between the two labeled noun phrases.

Our system represents a test sentence with a semantic graph, including the relation being tested and both of its proposed arguments. Semantic distance is calculated between this graph and a set of graphs representing the training examples relevant to the test sentence. A near-match between a test sentence and a positive training example is evidence that the same relation which holds in the example also holds in the test. We compute semantic distances to negative training examples as well, comparing the total positive and negative scores in order to decide whether a relation is true or false in the test sentence.

2 Motivation

Many systems which perform well on related tasks use syntactic features of the input sentence, coupled with classification by machine learning. This approach has been applied to problems like compound noun interpretation (Rosario and Hearst 2001) and semantic role labeling (Gildea and Jurafsky 2002).

In preparing our system for Task 4, we started by applying a similar syntax-based feature analysis to the trial data: 140 labeled examples of the relation "content-container". In 10-fold cross-validation with this data we achieved an average f-score of 70.6, based on features similar to the subset trees used for semantic role labeling in (Moschitti 2004). For classification we applied the updated tree-kernel package (Moschitti 2006), distributed with the svm-light tool (Joachims 1999) for learning Support Vector Machines (SVMs).

Training data for Task 4 is small, compared to other tasks where machine learning is commonly applied. We had difficulty finding a combination of features which gave good performance in cross-validation, but which did not result in a separate support vector being stored for every training sentence – a possible indicator of overfitting. As an example, the ratio of support vectors to training

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 121–124,
Prague, June 2007. ©2007 Association for Computational Linguistics

examples for the experiment described above was .97, nearly 1-to-1.

As a result of this analysis we started work on our knowledge-based system, with the goal of using the two approaches together. We were also motivated by an interest in using relation definitions and background knowledge from WordNet to greater advantage. The algorithm we used in our final submission is similar to recent systems which discover textual entailment relationships (Haghighi, Ng et al. 2005; Zanzotto and Moschitti 2006). It gives us a way to encode information from the relation definitions directly, in the form of statements in a knowledge representation language. The inference rules that are learned by this system from training examples are also easier to interpret than the models generated by an SVM. In small-data applications this can be an advantage.

3 System Description: A Walk-Through

The example sentence below is taken (in abbreviated form) from the training data for Task 4, Relation 7 "Content-Container" (Girju, Hearst et al. 2007):

The *kitchen* holds a *cooker*.

We convert this positive example into a semantic graph by creating a new instance of the relation Contains and linking that instance to the WordNet term for each labeled argument ("kitchen%1:06:00:", "cooker%1:06:00:"). The result is shown in Figure 1. WordNet sense keys (Fellbaum 1998) have been mapped to a term, a part of speech (pos), and a sense number.

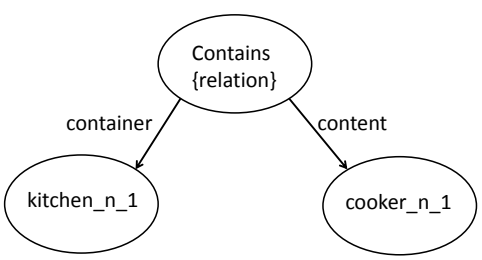

Figure 1. Semantic graph for the training example "The *kitchen* holds a *cooker*". Arguments are represented by a WordNet term, part of speech, and sense number.

This graph is instantiated as a statement using the Scone Knowledge Representation System, or

```
(new-statement {kitchen_n_1} {contains} {cooker_n_1})
(new-statement {artifact_n_1} {contains} {artifact_n_1})
(new-statement {whole_n_1} {contains} {whole_n_1})
```

Figure 2. Statements in Scone KR syntax, based on generalizing the training example "The *kitchen* holds a *cooker*".

"Scone" (Fahlman 2005). Scone gives us a way to store, search, and perform inference on graphs like the one shown above. After instantiating the graph we generalize it using hypernym information from WordNet. This generates additional Scone statements which are stored in a knowledge base (KB), shown in Figure 2. The first statement in the figure was generated verbatim from our training sentence. The remaining statements contain hypernyms of the original arguments.

For each argument seen in training, we also extract hypernyms and siblings from WordNet. For the argument kitchen, we extract 101 ancestors (artifact, whole, object, etc.) and siblings (structure, excavation, facility, etc.). A similar set of WordNet entities is extracted for the argument cooker. These entities, with repetitions removed, are encoded in a second Scone knowledge base, preserving the hierarchical (IS-A) links that come from WordNet. The hierarchy is manually linked at the top level into an existing background Scone KB where entities like animate, inanimate, person, location, and quantity are already defined.

After using the training data to create these two KBs, the system is ready for a test sentence. The following example is also adapted from SemEval Task 4 training data:

Equipment was carried in a *box*.

First we convert the sentence to a semantic graph, using the same technique as the one described above. The graph is implemented as a new Scone statement which includes the WordNet pos and sense number for each of the arguments: "box_n_1 contains equipment_n_1".

Next, using inference operations in Scone, the system verifies that the statement conforms to high-level constraints imposed by the relation definition. If it does, we calculate semantic distances between the argument nodes of our test statement and the analogous nodes in relevant training statements. A training statement is relevant if both of its arguments are ancestors of the appropriate ar-

guments of the test sentence. In our example, only two of the three KB statements from Figure 2 are relevant to the test statement "box contains equipment": "whole contains whole" and "artifact contains artifact". The first statement, "kitchen contains cooker" fails to apply because kitchen is not an ancestor of box, and also because cooker is not an ancestor of equipment.

Figure 3 illustrates the distance from "*box* contains *equipment*" to "*whole* contains *whole*", calculated as the sum of the distances between *box*-*whole* and *equipment*-*whole*.

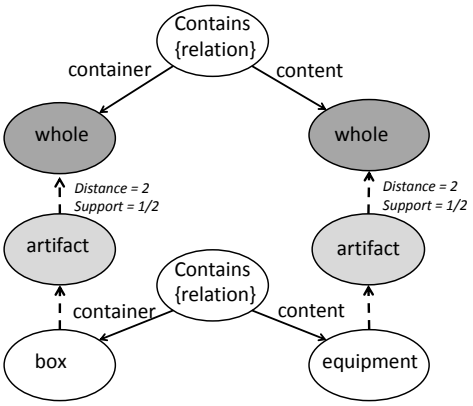

Figure 3. Calculating the distance through the knowledge base between "*equipment* contains *box*" and "*whole* contains *whole*". Dashed lines indicate IS-A links in the knowledge base.

The total number of these relevant, positive training statements is an indicator of "support" for the test sentence throughout the training data. The distance between one such statement and the test sentence is a measure of the strength of support. To reach a verdict, we sum over the inverse distances to all arguments from positive relevant examples: in Figure 3, the test statement "box contains equipment" receives a support score of (½ + ½ + 1 + 1), or 3.

Counter-evidence for a test sentence can be calculated in the same way, using relevant negative statements. In our example there are no negative training statements, so the total positive support score (3) is greater than the counter-evidence score (0), and the system verdict is "true".

4 System Components in Detail

As the detailed example above shows, this system is designed around its knowledge bases. The KBs provide a consistent framework for representing knowledge from a variety of sources as well as for calculating semantic distance.

4.1 Background knowledge

WordNet-extracted knowledge bases of the type described in Section 3 are generated separately for each relation. Average depth of these hierarchies is 4; we store only hypernyms of WordNet depth 7 and above, based on experiments in the literature by Nastase, et al. (2003; 2006).

Relation-specific and task-specific knowledge is encoded by hand. For each relation, we examine the relation definition and create a set of constraints in Scone formalism. For example, the definition of "container-contains" includes the following restriction (taken from training data for Task 4): *There is strong preference against treating legal entities (people and institutions) as content.*

In Scone, we encode this preference as a type restriction on the container role of any Contains relation: (new-is-not-a {container} {potential agent})

During testing, before calculating semantic distances, the system checks whether the test statement conforms to all such constraints.

4.2 Calculating semantic distance

Semantic distances are calculated between concepts in the knowledge base, rather than through WordNet directly. Distance between two KB entites is calculated by counting the edges along the shortest path between them, as illustrated in Figure 3. In the current implementation, only ancestors in the IS-A hierarchy are considered relevant, so this calculation amounts to counting the number of ancestors between an argument from the test sentence and an argument from a training example. Quick type-checking features which are built into Scone allow us to skip the distance calculation for non-relevant training examples.

5 Results & Conclusions

This system performed reasonably well for relation 3, Product-Producer, outperforming the baseline (baseline guesses "true" for every test sentence). Performance for this relation was also higher than the average F-score for all comparable groups in Task 4 (all groups in class "B4"). Average recall for this system over all relations was mid-range,

compared to other participating groups. Average precision and average f-score fell below the baseline and below the average for all comparable groups. These scores are given in Table 1.

Relation	R	P	F
1. Cause-Effect	73.2	54.5	62.5
2. Instrument-Agency	76.3	50.9	61.1
3. Product-Producer	79.0	71.0	74.8
4. Origin-Entity	63.9	54.8	59.0
5. Theme-Tool	48.3	53.8	50.9
6. Part-Whole	57.7	45.5	50.8
7. Content-Container	68.4	59.1	63.4
Whole test set, not divided by relation	57.1	68.9	62.4
Average for CMU-AT	66.7	55.7	60.4
Average for all B4 systems	64.4	65.3	63.6
Baseline: “alltrue”	100.0	48.5	64.8

Table 1. Recall, Precision, and F-scores, separated by relation type. Baseline score is calculated by guessing "true" for all test setences.

Analysis of the training data reveals that relation 3 is the class where target nouns occur most often together in nominal compounds and base NPs, with little additional syntax to connect them. While other relations included sentences where the targets were covered by a single VP, Product-Producer did not. It seems that background knowledge plays a larger role in identifying the Producer-Produces relationship than it does for other relations. However this conclusion is softened by the fact that we also spent more time in development and cross-evaluation for relations 3 and 7, our two best performing relations.

This system demonstrates a knowledge-based framework that performs very well for certain relations. Importantly, the system we submitted for evaluation did not make use of syntactic features, which are almost certainly relevant to this task. We are already exploring methods for combining the knowledge-based decision process with one that uses syntactic evidence as well as corpus statistics, described in Section 2.

Acknowledgement

This work was supported by a generous research grant from Cisco Systems, and by the Defense Advanced Research Projects Agency (DARPA) under contract number NBCHD030010.

References

Fahlman, S. E. (2005). Scone User's Manual.

Fellbaum, C. (1998). WordNet An Electronic Lexical Database, Bradford Books.

Gildea, D. and D. Jurafsky (2002). "Automatic labeling of semantic roles." Computational Linguistics 28(3): 245-288.

Girju, R., M. Hearst, et al. (2007). Classification of Semantic Relations between Nominals: Dataset for Task 4. SemEval 2007, 4th International Workshop on Semantic Evaluations, Prague, Czech Republic.

Haghighi, A., A. Ng, et al. (2005). Robust Textual Inference via Graph Matching. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada.

Joachims, T. (1999). Making large-scale SVM learning practical. Advances in Kernel Methods - Support Vector Learning. B. Schölkopf, C. Burges and A. Smola.

Moschitti, A. (2004). A study on Convolution Kernel for Shallow Semantic Parsing. proceedings of the 42nd Conference of the Association for Computational Linguistics (ACL-2004). Barcelona, Spain.

Moschitti, A. (2006). Making tree kernels practical for natural language learning. Eleventh International Conference on European Association for Computational Linguistics, Trento, Italy.

Nastase, V., J. S. Shirabad, et al. (2006). Learning noun-modifier semantic relations with corpus-based and Wordnet-based features. 21st National Conference on Artificial Intelligence (AAAI-06), Boston, Massachusetts.

Nastase, V. and S. Szpakowicz (2003). Exploring noun-modifier semantic relations. IWCS 2003.

Rosario, B. and M. Hearst (2001). Classifying the semantic relations in Noun Compounds. 2001 Conference on Empirical Methods in Natural Language Processing.

Zanzotto, F. M. and A. Moschitti (2006). Automatic Learning of Textual Entailments with Cross-Pair Similarities. the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL), Sydney, Austrailia.