UCD-FC: Deducing semantic relations using WordNet senses that occur frequently in a database of noun-noun compounds [*]

Fintan J. Costello,
School of Computer Science and Informatics, University College Dublin,
Belfield, Dublin 6, Ireland.
fintan.costello@ucd.ie

Abstract

This paper describes a system for classifying semantic relations among nominals, as in SemEval task 4. This system uses a corpus of 2,500 compounds annotated with WordNet senses and covering 139 different semantic relations. Given a set of nominal pairs for training, as provided in the SemEval task 4 training data, this system constructs for each training pair a set of features made up of relations and WordNet sense pairs which occurred with those nominals in the corpus. A Naive Bayes learning algorithm learns associations between these features and relation membership categories. The identification of relations among nominals in test items takes place on the basis of these associations.

1 Introduction

This paper describes a system for deducing the correct semantic relation between a pair of nominals in a sentence, as in SemEval task 4 (Girju, Hearst, Nakov, Nastase, Szpakowicz, Turney, & Yuret, 2007). This system is an adaptation of an existing system for deducing the correct semantic relation between the pair of words in a noun-noun compound. This compound disambiguation system (named PRO, for Proportional Relation Occurrence; see Costello, Veale, & Dunne, 2006) makes use of a corpus of 2,500 compounds annotated with WordNet senses and covering 139 different semantic relations, with each noun and each relation annotated with its correct WordNet sense.[1] Section 2 of the paper will describe the format and structure of this corpus, Section 3 will describe the original PRO compound disambiguation system, and Section 4 will explain how the PRO system was adapted to deduce the correct semantic relation between a pair of nominals, as in SemEval task 4. Four different versions of the adapted system were produced (versions A,B, C and D), either using or not using the WordNet labels and the Query labels provided with training and test items in SemEval task 4. Section 5 discusses the performance of these different versions of the system. Finally, Section 6 finishes the paper with some discussion and ideas for future work.

2 A Corpus of Annotated Compounds

Using WordNet (Miller, 1995), version 2.0, a corpus of noun-noun compounds was constructed such that each compound was annotated with the correct WordNet noun senses for constituent words, the correct semantic relation between those words, and the correct WordNet verb sense for that relation, as described below.

2.1 Corpus Procedure

The compounds used in this corpus were selected from the set of noun-noun compounds defined in WordNet. Compounds from WordNet were used because each compound had an associated gloss or

[*] This research was supported by the FP6 NEST Programme of the European Commission (ANALOGY: Humans the Analogy-Making Species: STREP Contr. No 029088)

[1] A file containing this corpus is available for download from http://inismor.ucd.ie/~fintanc/wordnet_compounds

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 370–373,
Prague, June 2007. ©2007 Association for Computational Linguistics

definition explaining the relation between the words in that compound (compounds from other sources would not have such associated definitions). Also, using compounds from WordNet guarantees that all constituent words of those compounds would also have entries in WordNet. An initial list of over 40,000 two-word noun-noun compounds was extracted from WordNet 2.0. From this list a random subset was selected. From that set all compounds using scientific latin (e.g. ocimum basilicum), idiomatic compounds (e.g. zero hour), compounds containing proper nouns (e.g. Yangtze river), non-english compounds (e.g. faux pas), and chemical terminology (e.g. carbon dioxide) were excluded.

The remaining compounds were placed in random order, and a research assistant annotated each with the WordNet noun senses of the constituent words, the semantic relation between those words, and the WordNet verb sense of that relation. A web page was created for this annotation task, showing the annotator the compound to be annotated and the WordNet gloss (meaning) for that compound. This page also showed the annotator the list of WordNet senses for the modifier noun and head noun in the compound, allowing the annotator to select the correct sense for each word. After word-sense selection another page was presented allowing the annotator to identify the correct semantic relation for that compound and to select the correct WordNet sense for the verb in that relation.

2.2 Corpus Results

Word sense, relation, and relation sense information was gathered for 2,500 compounds. Relation occurrence was well distributed across these compounds: there were 139 different relations used in the corpus. Note that in SemEval task 4, the number of relation categories available was much smaller than the set of relation categories available in our corpus (just 7 relation categories in the SemEval task).

3 Compound Disambiguation Algorithm

This section presents the ‘Proportional Relation Occurrence’ (PRO) algorithm which makes use of the corpus results described above to deduce semantic relations for noun-noun compounds. In Section 4 this algorithm is adapted to deduce relations between nominals in SemEval task 4.

```
Preconditions:
The entry for each compound $C$ in corpus $D$ contains:
$C_{modList}$ = sense + hypernym senses for modifier of $C$;
$C_{headList}$ = sense + hypernym senses for head of $C$;
$C_{rel}$ = semantic relation of $C$;

Input:
$X$ = compound for which a relation is required;
$modList$ = sense + hypernym senses for modifier of $X$;
$headList$ = sense + hypernym senses for head of $X$;
$finalRelationList$ = ();
$finalPairList$ = ();

Begin:
for each modifier sense $M \in modList$
    for each head sense $H \in headList$
        $relCount$ = ();
        $matchCount$ = 0;
        $P = (M, H)$;
        for each compound $C \in$ corpus $D$
            if (($M \in C_{modList}$) and ($H \in C_{headList}$))
                $relCount[C_{rel}]$ = $relCount[C_{rel}]$ + 1;
                $matchCount = matchCount$ + 1;
        for each relation $R \in relCount$
            $score$ = $relCount[R]/matchCount$;
            $prevScore = finalRelationList[R]$;
            if ($score > prevScore$)
                $finalRelationList[R] = score$;
            if ($score > pairScore$)
                $finalPairList[P] = score$;
sort $finalRelationList$ by score ;
sort $finalPairList$ by score ;
return ($finalRelationList, finalPairList$);
End.
```

Figure 1: PRO disambiguation algorithm.

The approach to compound disambiguation taken here is similar to that taken by for example Kim & Baldwin (2005) and Girju, Moldovan, Tatu, & Antohe (2005), and works by finding other compounds containing words from the same semantic categories as the words in the compound to be disambiguated: if a particular relation occurs frequently in those other compounds, that relation is probably also the correct relation for the compound in question. We take WordNet senses to represent semantic categories. Once the correct WordNet sense for a word has been identified, that word can placed in a set of nested semantic categories: the category represented by that sense, by the parent sense (or hypernym) of that sense, the parent of that parent, and so on up to the (notional) root sense of WordNet.

Figure 1 shows the algorithm in pseudocode. The algorithm uses the corpus of annotated noun-noun compounds and, to disambiguate a compound, takes as input the correct WordNet sense for the modifier and head words of that compound (if known) plus all hypernyms of those senses. If modifier and head word senses are not known, the most frequent senses for those words are used, plus all hypernyms of those senses. The algorithm pairs each modifier sense with every head sense. For each sense-pair, the algorithm goes through the corpus of compounds and extracts every compound whose modifier sense (or a hypernym of that sense) is equal to the modifier sense in the current sense-pair, and whose head sense (or a hypernym of that sense) is equal to the head sense in that pair. The algorithm counts the number of times each relation occurs in that set of compounds, and assigns each relation a Proportional Relation Occurrence (PRO) score for that pair, equal to the conditional probability of relation R given sense-pair S.

If the PRO score for relation R in the current sense-pair is greater than the score obtained for R with some other pair, the current score is recorded for R. If the score for R for the current pair P is greater than any previous score obtained for P, that score is recorded for P. In this way the algorithm finds the maximum score for each relation R across all sense-pairs, and the maximum score for each pair P across all relations. The algorithm returns a list of relations and of sense-pairs for the compound, both sorted by score. The relations and sense-pairs with the highest scores are those most likely to be correct for that compound and to be most important for its relational meaning.

In Costello, Veale and Dunne (2006), this algorithm was tested by applying it to the annotated corpus using a leave-one-out approach. These tests showed a reliable relationship between PRO score and accuracy of response. At a PRO level of 1, the algorithm return a response (selects a relation) for just over 900 compounds, and approximately 850 of those responses are correct (the algorithm's precision at this level is 0.92).

4 Adapting to the SemEval 4 task

To apply the PRO algorithm to the training and test sentences in SemEval task 4 first required a mapping from the labels used to tag nominals in that task (labels *e1* and *e2*) to the modifier and head categories used by the PRO algorithm. To carry out this mapping the nominal whose label appeared in the first position in a relation tag was taken to be the modifier for that relation, and that in the second position was taken to be the head; for example, with the relation tag *CONTAINER-CONTENT(E1,E2)* the nominal *e1* would be taken to be the modifer and *e2* to be the head. Given this mapping the PRO algorithm could be applied to sentences from SemEval task 4, taking modifier and head nominals as input and producing as output lists of candidate relations and relevant sense pairs (sorted by PRO score).

The relations produced by the PRO algorithm do not correspond to the 7 relations in SemEval task 4. To make predictions about the 7 SemEval relations, the scored relation lists and sense-pair lists returned by the PRO algorithm were used as features for a straightforward Naive Bayes learning algorithm, as implemented in the Perl module *Algorithm::NaiveBayes*. For each sentence in a training set in SemEval task 4, the PRO algorithm was applied to produce a list of relations and sense pairs describing that sentence. Each relation and each sense pair in this list has an associated PRO score, and Naive Bayes was trained on these features of all members of the training set, and then applied to test set sentences to produce predictions about each sentence's membership or non-membership in the relation in question.

Version A of the system used neither the WordNet sense tags nor the Query labels provided with the 7 relation categories used. Instead of using WordNet senses for the input words the system simply used the first (most frequent) noun senses for those words, and proceeded as described above. Version B used WordNet sense tags. Versions C and D of the system used either the first WordNet sense or the provided sense tags, coupled with the query terms used in the SemEval task. An additional module in the system was intended to make use of these query terms in relation classification by comparing the query term of the sentence to be classified with query terms in positive or negative training examples of that relation, and making a decision based on that comparison. Unfortunately, due to an error this query term module was not activated in the submitted runs, so the results from versions C and D are the same as from A and B.

Table 1: F-Score results by relation and run.

relation	A4	B4	C4	D4
Cause-Effect	72.1	65.1	72.1	65.1
Instrument-Agency	69.8	58.1	69.8	58.1
Product-Producer	73.1	73	73.1	73
Origin-Entity	43.1	42.3	43.1	42.3
Theme-Tool	50	49.2	50	49.2
Part-Whole	71.7	75	71.7	75
Content-Container	73.8	59.4	73.8	59.4
Avg	64.8	60.3	64.8	60.3

5 SemEval 4 task results

Table 1 shows the results returned for the PRO system for training run 4 (using all 140 training items in each relation) for the four possible runs A, B, C and D. Due to the error in activating the query term module, columns C4 and D4 are identical to columns A4 and B4. There are two notable aspects of the results in Table 1. First, the system's performance was better for run A4 (that did not use WordNet senses) than for B4 (using WordNet senses). Indeed, the system came first out of 6 systems which took part in the A4 run. This was surprising: it had been expected that using the correct WordNet senses for nominals would improve the system's performance. Analysis revealed that A4 runs using most frequent WordNet senses provided more matches with entries in the compound corpus the B4 run using the correct WordNet senses. This may explain why the system gave a better performance for A4 than B4.

The second interesting aspect of Table 1 is the variation of the system's responses across the different relation categories. For the two relations 'Origin-Entity' and 'Theme-Tool' the system has an F-score of 50 or less, while for the other five relations the system's F-score is around 70. It is not as yet clear why the system performed so poorly for these relations: further investigation is needed to explain this curious pattern.

6 Conclusions

This paper has described a system for automatically seslecting relations between nominals which uses the PRO algorithm and compound corpus to form features for pairs of nominals (consisting of candidate relations and sense-pairs co-occurring with those relations), and uses a Naive Bayes algorithm to learn to identify relations between nominals from those features. The system performs best using the most frequent WordNet senses for those nominals, suggesting that the system may work usefully in deducing semantic relations between nominals without the need to deduce word senses. However, the system's performance does not seem particularly impressive or suitable for application to real-world tasks as yet. The system's best performance represents an accuracy of 66% across relations: in other words, the system gets 1 in three relations wrong in the SemEval task.

There is one very obvious area for improvement in the system described here. Currently the system uses a simple Naive Bayes algorithm for learning associations between features and relation categories. A more sophisticated approach (using Support Vector Machines, for example) would be likely to improve the systsem's performance noticably. The conversion of the system to use some form of SVM should not be difficult. A more difficult problem, however, is to address the system's poor performance on some relations. This is currently difficult to understand, and represents a serious flaw in the system. Resolving this problem may reveal some useful aspects of the structure of different sorts of semantic relations between nominals.

References

F. J. Costello, T. Veale and S. Dunne. 2006. Using WordNet to Automatically Deduce Relations between Words in Noun-Noun Compounds. In Proceedings of the COLING/ACL 2006 Main Conference, pp 160–167.

R. Girju, M. Hearst, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney and D. Yuret. 2007. Classification of Semantic Relations between Nominals: Dataset for Task 4 in SemEval 2007. 4th International Workshop on Semantic Evaluations, June 23-24, Prague, Czech Republic.

R. Girju, D. Moldovan, M. Tatu, and D. Antohe. 2005. On the semantics of noun compounds. Computer Speech and Language 19, 4, 479–496.

S. N. Kim and T. Baldwin. Automatic Interpretation of Noun Compounds using WordNet::Similarity. 2nd Internationl Joint Conference on Natual Language Processing, Korea, 2005.

G. Miller. 1995. WordNet: A lexical database. Communication of the ACM, 38(11), 39–41.