MELB-KB: Nominal Classification as Noun Compound Interpretation

Su Nam Kim and Timothy Baldwin
Computer Science and Software Engineering
University of Melbourne, Australia
{snkim,tim}@csse.unimelb.edu.au

Abstract

In this paper, we outline our approach to interpreting semantic relations in nominal pairs in SemEval-2007 task #4: Classification of Semantic Relations between Nominals. We build on two baseline approaches to interpreting noun compounds: sense collocation, and constituent similarity. These are consolidated into an overall system in combination with co-training, to expand the training data. Our two systems attained an average F-score over the test data of 58.7% and 57.8%, respectively.

1 Introduction

This paper describes two systems entered in SemEval-2007 task #4: Classification of Semantic Relations between Nominals. A key contribution of this research is that we examine the compatibility of noun compound (NC) interpretation methods over the extended task of nominal classification, to gain empirical insight into the relative complexity of the two tasks.

The goal of the nominal classification task is to identify the compatibility of a given semantic relation with each of a set of test nominal pairs, e.g. between *climate* and *forest* in the fragment *the <u>climate</u> in the <u>forest</u>* with respect to the CONTENT-CONTAINER relation. Semantic relations (or SRs) in nominals represent the underlying interpretation of the nominal, in the form of the directed relation between the two nominals.

The proposed task is a generalisation of the more conventional task of interpreting noun compounds (NCs), in which we take a NC such as *cookie jar* and interpret it according to a pre-defined inventory of semantic relations (Levi, 1979; Vanderwende, 1994; Barker and Szpakowicz, 1998). Examples of semantic relations are MAKE,[1], as exemplified in *apple pie* where the *pie* is made from *apple(s)*, and POSSESSOR, as exemplified in *family car* where the *car* is possessed by a *family*.

In the SemEval-2007 task, SR interpretation takes the form of a binary decision for a given nominal pair in context and a given SR, in judging whether that nominal pair conforms to the SR. Seven relations were used in the task: CAUSE-EFFECT, INSTRUMENT-AGENCY, PRODUCT-PRODUCER, ORIGIN-ENTITY, THEME-TOOL, PART-WHOLE and CONTENT-CONTAINER.

Our approach to the task was to: (1) naively treat all nominal pairs as NCs (e.g. *the climate in the forest* is treated as an instance of *climate forest*); and (2) translate the individual binary classification tasks into a single multiclass classification task, in the interests of benchmarking existing SR interpretation methods over a common dataset. That is, we take all positive training instances for each SR and pool them together into a single training dataset. For each test instance, we make a prediction according to one of the seven relations in the task, which we then map onto a binary classification for final evaluation purposes. This mapping is achieved by determining which binary SR classification the test instance was sourced from, and returning a positive classification if the predicted SR coincides with the target SR, and a negative classification if not.

We make three (deliberately naive) assumptions in our approach to the nominal interpretation task. First, we assume that all the positive training in-

[1] For direct comparability with our earlier research, semantic relations used in our examples are taken from (Barker and Szpakowicz, 1998), and differ slightly from those used in the SemEval-2007 task.

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 231–236,
Prague, June 2007. ©2007 Association for Computational Linguistics

stances correspond uniquely to the SR in question, despite the task organisers making it plain that there is semantic overlap between the SRs. As a machine learning task, this makes the task considerably more difficult, as the performance for the standard baselines drops considerably from that for the binary tasks. Second, we assume that each nominal pair maps onto a NC. This is clearly a misconstrual of the task, and intended to empirically validate whether such an approach is viable. In line with this assumption, we will refer to nominal pairs as NCs for the remainder of the paper. Third and finally, we assume that the SR annotation of each training and test instance is insensitive to the original context, and use only the constituent words in the NC to make our prediction. This is for direct comparability with earlier research, and we acknowledge that the context (and word sense) is a strong determinant of the SR in practice.

Our aim in this paper is to demonstrate the effectiveness of general-purpose SR interpretation over the nominal classification task, and establish a new baseline for the task.

The remainder of this paper is structured as follows. We present our methods in Section 2 and depict the system architectures in Section 4. We then describe and discuss the performance of our methods in Section 5 and conclude the paper in Section 6.

2 Approach

We used two basic NC interpretation methods. The first method uses sense collocations as proposed by Moldovan et al. (2004), and the second method uses the lexical similarity of the component words in the NC as proposed by Kim and Baldwin (2005). Note that neither method uses the context of usage of the NC, i.e. the only features are the words contained in the NC.

2.1 Sense Collocation Method

Moldovan et al. (2004) proposed a method called semantic scattering for interpreting NCs. The intuition behind this method is that when the sense collocation of NCs is the same, their SR is most likely the same. For example, the sense collocation of *automobile factory* is the same as that of *car factory*, because the senses of *automobile* and *car*, and *factory* in the two instances, are identical. As a result, the two NCs have the semantic relation MAKE.

The semantic scattering model is outlined below.

The probability $P(r|f_i f_j)$ (simplified to $P(r|f_{ij})$) of a semantic relation r for word senses f_i and f_j is calculated based on simple maximum likelihood estimation:

$$P(r|f_{ij}) = \frac{n(r, f_{ij})}{n(f_{ij})} \qquad (1)$$

and the preferred SR r^* for the given word sense combination is that which maximises the probability:

$$\begin{aligned} r^* &= \text{argmax}_{r \in R} P(r|f_{ij}) \\ &= \text{argmax}_{r \in R} P(f_{ij}|r) P(r) \end{aligned} \qquad (2)$$

Note that in limited cases, the same sense collocation can lead to multiple SRs. However, since we do not take context into account in our method, we make the simplifying assumption that a given sense collocation leads to a unique SR.

2.2 Constituent Similarity Method

In earlier work (Kim and Baldwin, 2005), we proposed a simplistic general-purpose method based on the lexical similarity of unseen NCs with training instances. That is, the semantic relation of a test instance is derived from the train instance which has the highest similarity with the test instance, in the form of a 1-nearest neighbour classifier. For example, assuming the test instance *chocolate milk* and training instances *apple juice* and *morning milk*, we would calculate the similarity between modifier *chocolate* and each of *apple* and *morning*, and head noun *milk* and each of *juice* and *milk*, and find, e.g., the similarities .71 and .27, and .83 and 1.00 respectively. We would then add these up to derive the overall similarity for a given NC and find that *apple juice* is a better match. From this, we would assign the SR of MAKE from *apple juice* to *chocolate milk*.

Formally, S_A is the similarity between NCs $(N_{i,1}, N_{i,2})$ and $(B_{j,1}, B_{j,2})$:

$$S_A((N_{i,1}, N_{i,2}), (B_{j,1}, B_{j,2})) = \frac{((\alpha S1 + S1) \times ((1 - \alpha) S2 + S2))}{2} \qquad (3)$$

where $S1$ is the modifier similarity (i.e. $S(N_{i,1}, B_{j1})$) and $S2$ is head noun similarity (i.e. $S(N_{i,2}, B_{j2})$); $\alpha \in [0, 1]$ is a weighting factor. The similarity scores are calculated using the method of Wu and Palmer (1994) as implemented in `WordNet::Similarity` (Patwardhan et al., 2003). This is done for each pairing of WordNet senses of each of the two words in question, and the overall lexical similarity is calculated as the average across the pairwise sense similarities.

The final classification is derived from the training instance which has the highest lexical similarity with the test instance in question.

3 Co-Training

As with many semantic annotation tasks, SR tagging is a time-consuming and expensive process. At the same time, due to the inherent complexity of the SR interpretation task, we require large amounts of training data in order for our methods to perform well. In order to generate additional training data to train our methods over, we experiment with different co-training methodologies for each of our two basic methods.

3.1 Co-Training for the Sense Collocation Method

For the sense collocation method, we experiment with a substitution method whereby we replace one constituent in a training NC instance by a similar word, and annotate the new instance with the same SR as the original NC. For example, *car* in *car factory* (SR = MAKE) has similar words *automobile, vehicle, truck* from the synonym, hypernym and sister word taxonomic relations, respectively. When *car* is replaced by a similar word, the new noun compound(s) (i.e. *automobile/vehicle/truck factory*) share the same SR as the original *car factory*. Note that each constituent in our original example is tagged for word sense, which we use both in accessing sense-specific substitution candidates (via WordNet), and sense-annotating the newly generated NCs.

Substitution is restricted to one constituent at a time in order to avoid extreme semantic variation. This procedure can be repeated to generate more training data. However, as the procedure goes further, we introduce increasingly more noise.

In our experiments, we use this co-training method with the sense collocation method to expand the size and variation of training data, using synonym, hypernym and sister word relations. For our experiment, we ran the expansion procedure for only one iteration in order to avoid generating excessive amounts of incorrectly-tagged NCs.

3.2 Co-Training for the Constituent Similarity Method

Our experiments with the constituent similarity method over the trial data showed, encouragingly, that there is a strong correlation between the strength of overall similarity with the best-matching training NC, and the accuracy of the prediction. From this, we experimented with implementing the constituent similarity method in a cascading architecture. That is, we batch evaluate all test instances on each iteration, and tag those test instances for which the best match with a training instance is above a preset threshold, which we decrease on each iteration. In subsequent iterations, all tagged test instances are included in the training data. Hence, on each iteration, the number of training instances is increasing. As our threshold, we used a starting value of 0.85, which was decreased down to 0.65 in increments of 0.05.

4 Architectures

In Section 4.1 and Section 4.2, we describe the architecture of our two systems.

4.1 Architecture (I)

Figure 1 presents the architecture of our first system, which interleaves sense collocation and constituent similarity, and includes co-training for each. There are five steps in this system.

First, we apply the basic sense collocation method relative to the original training data. If the sense collocation between the test and training instances is the same, we judge the predicted SR to be correct.

Second, we apply the similarity method described in Section 2.2 over the original training data. However, we only classify test instances where the final similarity is above a threshold of 0.8.

Third, we apply the sense collocation co-training method and re-run the sense collocation method over the expanded training data from the first two steps. Since the sense collocations in the expanded

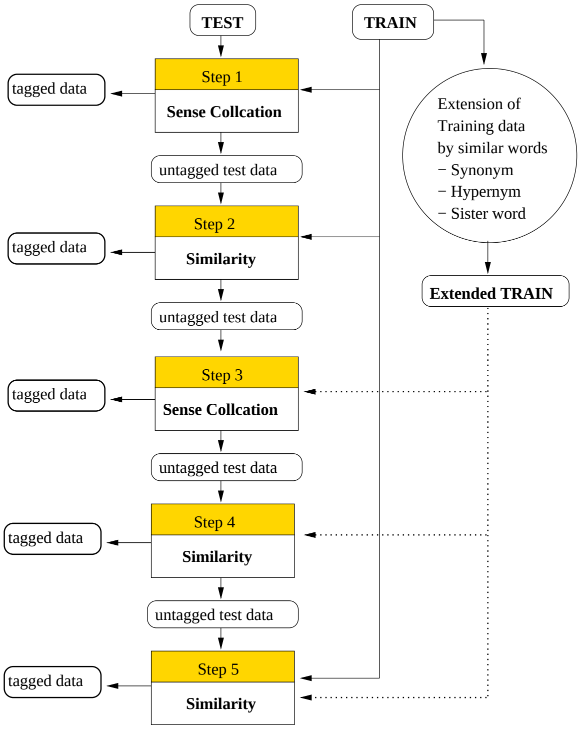

Figure 1: System Architecture (I)

training data have been varied through the advent of hypernyms and sister words, the number of sense collocations in the expanded training data is much greater than that of the original training data (937 vs. 16,676).

Fourth, we apply the constituent similarity co-training method over the consolidated training data (from both sense collocation and constituent similarity co-training) with the threshold unchanged at 0.8.

Finally, we apply the constituent similarity method over the combined training data, without any threshold (to guarantee a SR prediction for every test instance). However, since the generated training instances are more likely to contain errors, we decrement the similarity values for generated training instances by 0.2, to prefer predictions based on the original training instances.

4.2 Architecture (II)

Figure 2 depicts our second system, which is based solely on the constituent similarity method, with co-training.

We perform iterative co-training as described in Section 3.2, with the slight variation that we hold off reducing the threshold if more than 10% of the test instances are tagged on a given iteration, giving other test instances a chance to be tagged at a higher threshold level relative to newly generated training instances. The residue of test instances on completion of the final iteration (threshold = 0.6) are tagged according to the best-matching training instance, irrespective of the magnitude of the similarity.

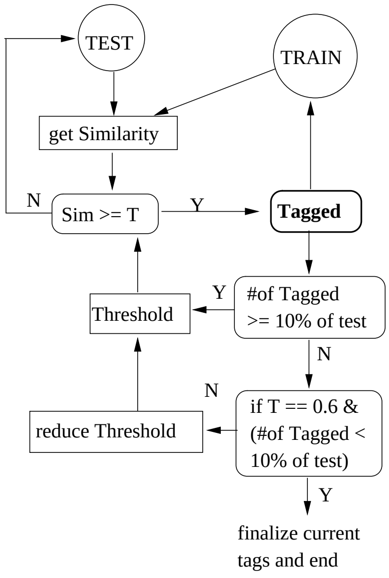

Figure 2: System Architecture (II)

5 Evaluation

We group our evaluation into two categories: (A) doesn't use WordNet 2.1 or the query context; and (B) uses WordNet 2.1 only (again without the query context). Of our two basic methods the sense collocation method and co-training method are based on WordNet 2.1 only, while the constituent similarity method is based indirectly on WordNet 2.1, but doesn't preserve WordNet 2.1 sense information. Hence, our first system is category B while our second system is (arguably) category A.

Table 1 presents the three baselines for the task, and the results for our two systems (System I and System II). The performance for both systems exceeded all three baselines in terms of accuracy, and all but the All True baseline (i.e. every instance is judged to be compatible with the given SR) in terms

Method	P	R	F	A
All True	48.5	100.0	64.8	48.5
Probability	48.5	48.5	48.5	51.7
Majority	81.3	42.9	30.8	57.0
System I	61.7	56.8	58.7	62.5
System II	61.5	55.7	57.8	62.7

Table 1: System results (*P* = precision, *R* = recall, *F* = F-score, and *A* = accuracy)

Team	P	R	F	A
759	66.1	66.7	64.8	66.0
281	60.5	69.5	63.8	63.5
633	62.7	63.0	62.7	65.4
220	**61.5**	**55.7**	**57.8**	**62.7**
161	56.1	57.1	55.9	58.8
538	48.2	40.3	43.1	49.9

Table 2: Results of category A systems

of F-score and recall.

Tables 2 and 3 show the performance of the teams which performed in the task, in categories A and B. Team 220 in Table 2 is our second system, and team 220 in Table 3 is our first system.

In Figures 3 and 4, we present a breakdown of the performance our first and second system, respectively, over the individual semantic relations. Our approaches performed best for the PRODUCT-PRODUCER SR, and worst for the PART-WHOLE SR. In general, our systems achieved similar performance on most SRs, with only PART-WHOLE being notably worse. The lower performance of PART-WHOLE pulls down our overall performance considerably.

Tables 4 and 5 show the number of tagged and untagged instances for each step of System I and System II, respectively. The first system tagged more than half of the data in the fifth (and final) step, where it weighs up predictions from the original and expanded training data. Hence, the performance of this approach relies heavily on the similarity method and expanded training data. Additionally, the difference in quality between the original and expanded training data will influence the performance of the approach appreciably. On the other hand, the number of instances tagged by the second system is well distributed across each iteration. However, since we accumulate generated training instances on each step, the relative noise level in the training data will increase across iterations, impacting on the final performance of the system.

Team	P	R	F	A
901	79.7	69.8	72.4	76.3
777	70.9	73.4	71.8	72.9
281	72.8	70.6	71.5	73.2
129	69.9	64.6	66.8	71.4
333	62.0	71.7	65.4	67.0
538	66.7	62.8	64.3	67.2
571	55.7	66.7	60.4	59.1
759	66.4	58.1	60.3	63.6
220	**61.7**	**56.8**	**58.7**	**62.5**
371	56.8	56.3	56.1	57.7
495	55.9	57.8	51.4	53.7

Table 3: Results of category B systems

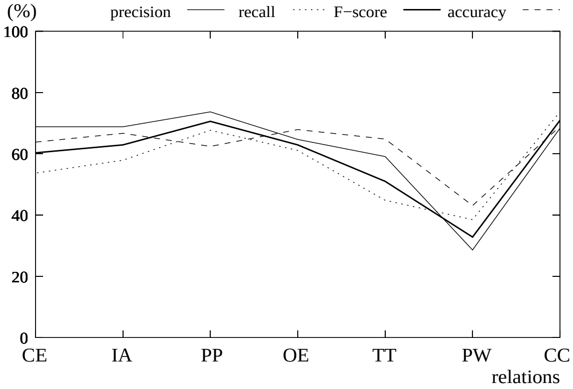

Figure 3: System I performance for each relation (CC=CAUSE-EFFECT, IA=INSTRUMENT-AGENCY, PP=PRODUCT-PRODUCER, OE=ORIGIN-ENTITY, TT=THEME-TOOL, PW=PART-WHOLE, CC=CONTENT-CONTAINER)

Over the trial data, we noticed that the system predictions are appreciably worse when the similarity value is low. In future work, we intend to analyse what is happening in terms of the overall system performance at each step. This analysis is key to improving the performance of our systems.

Recall that we are generalising from the set of binary classification tasks in the original task, to a multiclass classification task. As such, a direct comparison with the binary classification baselines is perhaps unfair (particularly All True, which has no correlate in a multiclass setting), and it is if anything remarkable that our system compares favourably compared to the baselines. Similarly, while we clearly lag behind other systems participating in the

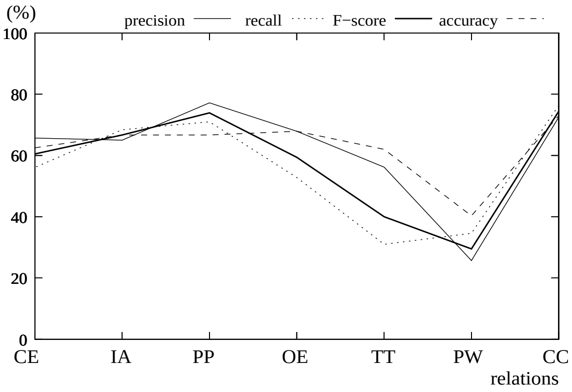

Figure 4: System II performance for each relation (CC=CAUSE-EFFECT, IA=INSTRUMENT-AGENCY, PP=PRODUCT-PRODUCER, OE=ORIGIN-ENTITY, TT=THEME-TOOL, PW=PART-WHOLE, CC=CONTENT-CONTAINER)

step	method	tagged	accumulated	untagged
s1	SC	21	3.8%	528
s2	Sim	106	23.1%	422
s3	extSC	0	23.1%	422
s4	extSim	61	34.2%	361
s5	SvsExtS	359	99.6%	2

Table 4: System I: Tagged data from each step (*SC*= sense collocation; *Sim* = the similarity method; *extSC* = SC over the expanded training data; *extSim* = similarity over the expanded training data; *SvsExtS* = the final step over both the original and expanded training data)

task, we believe we have demonstrated that NC interpretation methods can be successfully deployed over the more general task of nominal pair classification.

6 Conclusion

In this paper, we presented two systems entered in the SemEval-2007 Classification of Semantic Relations between Nominals task. Both systems are based on baseline NC interpretation methods, and the naive assumption that the nominal classification task is analogous to a conventional multiclass NC interpretation task. Our results compare favourably with the established baselines, and demonstrate that NC interpretation methods are compatible with the more general task of nominal classification.

I	T	tagged	accumulated	untagged
i1	.85	73	13.3%	476
i2	.80	56	23.5%	420
i3	.75	74	37.0%	346
i4	.70	101	55.4%	245
i5	.65	222	95.8%	23
–	<.65	21	99.6%	2

Table 5: System II: data tagged on each iteration (*T* = the threshold; *iX* = the iteration number)

Acknowledgments

This research was carried out with support from Australian Research Council grant no. DP0663879.

References

Ken Barker and Stan Szpakowicz. 1998. Semi-automatic recognition of noun modifier relationships. In *Proc. of the 17th International Conference on Computational Linguistics*, pages 96–102, Montreal, Canada.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Timothy W. Finin. 1980. *The Semantic Interpretation of Compound Nominals*. Ph.D. thesis, University of Illinois.

Su Nam Kim and Timothy Baldwin. 2005. Automatic interpretation of Noun Compounds using WordNet similarity. In *Proc. of the 2nd International Joint Conference On Natural Language Processing*, pages 945–956, JeJu, Korea.

Judith Levi. 1979. The syntax and semantics of complex nominals. In *The Syntax and Semantics of Complex Nominals*. New York:Academic Press.

Dan Moldovan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju. 2004. Models for the semantic classification of noun phrases. In *Proc. of the HLT-NAACL 2004 Workshop on Computational Lexical Semantics*, pages 60–67, Boston, USA.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proc. of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–57, Mexico City, Mexico.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *Proc. of the 15th conference on Computational linguistics*, pages 782–788, Kyoto, Japan.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, USA.