Benchmarking Noun Compound Interpretation

Su Nam Kim and Timothy Baldwin
Department of Computer Science and Software Engineering
and
NICTA Victoria Lab
University of Melbourne, VIC 3010 Australia
{snkim,tim}@csse.unimelb.edu.au

Abstract

In this paper we provide benchmark results for two classes of methods used in interpreting noun compounds (NCs): semantic similarity-based methods and their hybrids. We evaluate the methods using 7-way and binary class data from the nominal pair interpretation task of SEMEVAL-2007.[1] We summarize and analyse our results, with the intention of providing a framework for benchmarking future research in this area.

1 Introduction

This paper reviews a range of simple and hybrid approaches to noun compound (NC) interpretation. The interpretation of NCs such as *computer science* and *paper submission* involves predicting the **semantic relation** (SR) that underlies a given NC. For example, *student price* conventionally expresses the meaning that a *student* benefits from the *price* (SR = BENEFICIARY), while *student protest* conventionally means a *student* undertaking a *protest* (SR = AGENT).[2]

NCs are formed from simplex nouns with high productivity. The huge number of possible NCs and potentially large number of SRs makes NC interpretation a very difficult problem. In the past, much NC interpretation work has been carried out which targets particular NLP applications such as information extraction, question-answering and machine translation. Unfortunately, much of it has not gained traction in real-world applications as the accuracy of the methods has not been sufficiently high over open-domain data. Most prior work has been carried out under specific assumptions and with one-off datasets, which makes it hard to analyze performance and to build hybrid methods. Additionally, disagreement in the inventory of SRs and a lack of resource sharing has hampered comparative evaluation of different methods.

The first step in NC interpretation is to define a set of SRs. Levi (1979), for example, proposed a system of 9 SRs, while others have proposed classifications with 20-30 SRs (Finin, 1980; Barker and Szpakowicz, 1998; Moldovan et al., 2004). Smaller sets tend to have reduced coverage due to coarse granularity, whereas larger sets tend to be too fine grained and suffer from low inter-annotator agreement. Additionally pragmatic/contextual differentiation leads to difficulties in defining and interpreting SRs (Downing, 1977; SparckJones, 1983).

Recent attempts in the area of NC interpretation have taken two basic approaches: analogy-base interpretation (Rosario, 2001; Moldovan et al., 2004; Kim and Baldwin, 2005; Girju, 2007) and semantic disambiguation relative to an underlying predicate or semantically-unambiguous paraphrase (Vanderwende, 1994; Lapata, 2002; Kim and Baldwin, 2006; Nakov, 2006). Most methods employ rich ontologies and ignore the context of use, supporting the claim by Fan (2003) that axioms and ontological distinctions are more important than detailed knowledge of specific nouns for NC interpretation. Additionally, most approaches use supervised learning, raising questions about the generality of the test and

[1] The 4th International Workshop on Semantic Evaluation

[2] SRs used in the examples are taken from Barker and Szpakowicz (1998).

Kim, Su Nam and Timothy Baldwin (2008) Benchmarking Noun Compound Interpretation, In Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008), Hyderabad, India.

training data sets and the effectiveness of the algorithms in different domains (coverage of SRs over the NCs is another issue).

Our aim in this paper is to compare and analyze existing NC interpretation methods over a common, publicly available dataset. While recent research has made significant progress, bringing us one step closer to practical applicability in NLP applications, no direct comparison or analysis of the approaches has been attempted to date. As a result, it is hard to determine which approach is appropriate in a given domain or build hybrid methods based on prior approaches. We also investigate the impact on performance of relaxing assumptions made in the original research, to compare different approaches in an identical setting.

In the remainder of the paper, we review the research background and NC interpretation methods in Section 2, describe the methods and system architectures in Section 3, detail the datasets used in our experiments in Section 4, carry out a system evaluation in Section 5 and Section 6, and finally present a discussion and conclusions in Section 7 and Section 8, respectively.

2 Background and Methods

2.1 Research Background

In this study, we selected three semantic similarity based models which had been found to perform strongly in previous research, and which were easy to re-implement: SENSE COLLOCATION (Moldovan et al., 2004), CONSTITUENT SIMILARITY (Kim and Baldwin, 2005) and CO-TRAINING, e.g. using SENSE COLLOCATION or CONSTITUENT SIMILARITY (Kim and Baldwin, 2007). These approaches were evaluated over a 7-way classification using open-domain data from the nominal pair interpretation task of SEMEVAL-2007 (Girju et al., 2007). We test their performance in both 7-way and binary-class classification settings.

2.2 Sense Collocation Method

The SENSE COLLOCATION method of Moldovan et al. (2004) is based on the pair of word senses of NC constituents. The basic idea is that NCs which have the same or similar sense collocation tend to have the same SR. As an example, *car factory* and *automobile factory* share the conventional interpretation of MAKE, which is predicted by *car* and *automobile* having the same sense across the two NCs, and *factory* being used with the same sense in each instance. This intuition is formulated in Equations 1 and 2 below.

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

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

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

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

2.3 Constituent Similarity Method

The intuition behind the CONSTITUENT SIMILARITY method is similar to the SENSE COLLOCATION method, in that NCs made up of similar words tend to share the same SR. The principal difference is that it doesn't presuppose that we know the word sense of each constituent word (i.e. the similarity is calculated at the *word* rather than sense level). The method takes the form of a 1-nearest neighbour classifier, with the best-matching training instance for each test instance predicting its SR. For example, we may find that test instance *chocolate milk* most closely matches *apple juice* and hence predict that the SR is MATERIAL.

This idea is formulated in Equation 3 below. Formally, S_A is the similarity between NCs $(N_{i,1}, N_{i,2})$ and $(B_{j,1}, B_{j,2})$:

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

where $S1$ is the modifier similarity (i.e. $S(N_{i,1}, B_{j1})$) and $S2$ is the head noun similarity (i.e. $S(N_{i,2}, B_{j2})$); $\alpha \in [0, 1]$ is a weighting factor. The similarity scores are calculated across the bag of WordNet senses (without choosing between them) 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 the two words in question, and the overall lexical similarity is calculated as the average across the pairwise sense similarities.

2.4 Co-Training by Sense Collocation

Co-training by sense collocation (SColl co-training) is based on the sense collocation method and lexical substitution (Kim and Baldwin, 2007). It expands the set of training NCs from a relatively small number of manually-tagged seed instances. That is, it makes use of extra training instances fashioned through a bootstrap process. For example, assuming *automobile factory* with the SR MAKE were a seed instance, NCs generated from synonyms, hypernyms and sister words of its constituents would be added as extra training instances, with the same SR of MAKE. That is, we would add *car factory* (**SYNONYM**), *vehicle factory* (**HYPERNYM**) and *truck factory* (**SISTER WORD**), for example. Note that the substitution takes place only for one constituent at a time to avoid extreme variation.

2.5 Co-training by Constituent Similarity

Co-training by Constituent Similarity (CS co-training) is also a co-training method, but based on constituent similarity rather than sense collocation. The basic idea is that when NCs are interpreted using the constituent similarity method, the predictions are more reliable when the lexical similarity is higher. Hence, we progressively reduce the similarity threshold, and incorporate higher-similarity instances into our training data earlier in the bootstrap process. That is, we run the constituent similarity method and acquire NCs with similarity equal to or greater than a fixed threshold. Then in the next iteration, we add the acquired NCs into the training dataset for use in classifying more instances. As a result, in each step, the number of training instances increases monotonically. We "cascade" through a series of decreasing similarity thresholds until we reach a saturation point. As our threshold, we used a starting value of 0.90, which was decremented down to 0.65 in steps of 0.05.

Method	Description
SColl	sense collocation
SColl$_{\text{CT}}$	sense collocation + SColl co-training
CSim	constituent similarity
CSim +SColl$_{\text{CT}}$	constituent similarity + SColl co-training
hybrid	SColl + CSim + SColl$_{\text{CT}}$
CSim$_{\text{CT}}$	constituent similarity + CSim co-training

Table 1: Systems used in our experiments

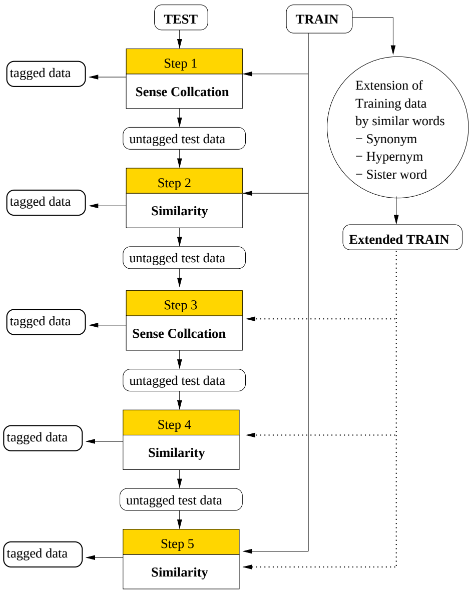

Figure 1: Architecture of the hybrid method

3 Systems and Architectures

We tested the original methods of Moldovan et al. (2004) and Kim and Baldwin (2005), and combined them with the co-training methods of Kim and Baldwin (2007) to come up with six different hybrid systems for evaluation, as detailed in Table 1. To build the classifiers, we used the TiMBL5.0 memory-based learner (Daelemans et al., 2004).

The hybrid method consists of five interpretation steps. The first step is to use the sense collocation method over the original training data. When the sense collocation of the test and training instances is the same, we judge the predicted SR to be correct. The second step is to apply the constituent similarity method over the original training data. In order to confirm that the predicted SR is correct, we use a threshold of 0.8 to interpret the test instances. The third step is to apply sense collocation over the expanded train-

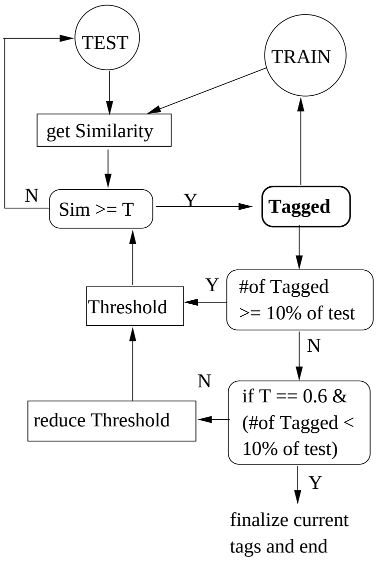
TEST
TRAIN
get Similarity
N
Sim >= T
Y
Tagged
Y
Threshold
#of Tagged >= 10% of test
N
N
reduce Threshold
if T == 0.6 & (#of Tagged < 10% of test)
Y
finalize current tags and end

Figure 2: Architecture of the CSIM$_{CT}$ system

ing data through the advent of hypernyms and sister words, using the SCOLL CO-TRAINING method. This step benefits from a larger amount of training data (17,613 vs. 937). The fourth step is to apply the CONSTITUENT SIMILARITY method (EXTCS) over the consolidated training data, with the threshold unchanged at 0.8. The final step is to apply the CONSTITUENT SIMILARITY (CSTT) method over the combined training data without any restriction on the threshold (to guarantee a SR prediction for every test instance). We select SRs from the training instances whose similarity is higher than the original training data and expanded training data. However, since the generated training instances are more likely to contain errors, we apply a linear weight of 0.8 to the similarity values for the expanded training instances. This gives preferential treatment to predictions based on the original training instances. Note that this weight was based on analysis of the error rate in the expanded training instances. In previous work (Kim and Baldwin, 2007), we found the overall classification accuracy rate after the first iteration to be 70-80%. Hence, we settled on a weight of 0.8.

The CSIM$_{CT}$ system is based solely on the CONSTITUENT SIMILARITY method with cascading. We perform iterative CS co-training as described in Section 2.5, with the slight variation that we hold off on reducing the threshold if less 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.

	Binary			7-way		
SR	Test	Train	Train*	Test	Train	Train*
CE	80	136	2,588	36	71	1,854
IA	78	135	1,400	36	68	1,001
PP	93	126	2,591	55	78	2,089
OE	81	136	3,085	35	52	1,560
TT	71	129	2,994	27	50	1,718
PW	72	138	2,577	28	64	1,510
CC	74	137	2,378	37	63	1,934
Total	549	937	17,613	254	446	11,664

Table 3: Number of instances associated with each SR (Train* is the number of expanded train instances)

4 Data

We used the dataset from the SEMEVAL-2007 nominal pair interpretation task, which is based on 7 SRs: CAUSE-EFFECT (CE), INSTRUMENT-AGENCY (IA), PRODUCT-PRODUCER (PP), ORIGIN-ENTITY (OE), THEME-TOOL (TT), PART-WHOLE (PW), CONTENT-CONTAINER (CC). The task in SEMEVAL-2007 was to identify the compatibility of a given SR for each test instances using word senses retrieved from WORDNET 3.0 (Fellbaum, 1998) and queries. Table 2 shows the definition of the SRs.

In our research, we interpret the dataset in two ways: (1) as a **binary classification** task for each SR based on the original data; and (2) as a **7-way classification** task, combining together all positive test and training instances for each of the 7 SR datasets into a single dataset. Hence, the size of the dataset for 7-way classification is much smaller than that of the original dataset. We also expand the training instances using SCOLL CO-TRAINING. Table 3 describes the number of test and train instances for NC interpretation for the binary and 7-way classification tasks.

Our analysis shows that only 5 NCs are repeated

Semantic relation	Definition	Examples
Cause-Effect (**CE**)	N_1 is the cause of N_2	*virus flu, hormone growth*
Instrument-Agency (**IA**)	N_1 is the instrument of N_2; N_2 uses N_1	*laser printer, axe murderer*
Product-Producer (**PP**)	N_1 is a product of N_2; N_2 produces N_1	*honey bee, music clock*
Origin-Entity (**OE**)	N_1 is the origin of N_2	*bacon grease, desert storm*
Theme-Tool (**TT**)	N_2 is intended for N_1	*reorganization process, copyright law*
Part-Whole (**PW**)	N_1 is part of N_2	*table leg, daisy flower*
Content-Container (**CC**)	N_1 is stored or carried inside N_2	*apple basket, wine bottle*

Table 2: The set of 7 semantic relations, where N_1 is the modifier and N_2 is the head noun

across multiple SR datasets (i.e. occur as an instance in more than one of the 7 datasets), none of which occur as positive instances for multiple SRs. As such, no NC instances in the 7-way classification task end up with a multiclass classification. Also note that some of NCs are contained within ternary or higher-order NCs: 40 test NCs and 81 training NCs for the binary classification task, and 24 test NCs and 42 training NCs for the 7-way classification task. For these NCs, we extracted a "base" binary NC based on the provided bracketing. The following are examples of extraction of binary NCs from ternary or higher-order NCs.

((billiard table) room) → table room
(body (bath towel)) → body towel

In order to extract a binary NC, we take the head noun of each embedded NC and combine this with the corresponding head noun or modifier. E.g., *table* is the head noun of *billiard table*, which combines with the head noun of the complex NC *room* to form *table room*.

5 Experiment 1: 7-way classification

Our first experiment was carried out over the 7-way classification task—i.e. all 7 SRs in a single classification task—using the 6 systems from Section 3. In our results in Table 4, we use the system categories from SEMEVAL-2007 of **A4** and **B4**, where A4 systems use none of the provided word senses, and B4 systems use the word senses.[3] We categorized our systems into these two groups in order to evaluate them separately within the bounds of the original SEMEVAL-2007 task. In each case, the baseline is a majority class classifier.

Class	Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{A}$
–	Majority				.217
A4	CSIM	.518	.522	**.449**	**.528**
	CSIM$_{\text{CT}}$	.517	.511	.426	.522
B4	SCOLL	.705	.444	.477	.496
	SCOLL$_{\text{CT}}$	.646	.466	**.498**	.508
	CSIM +SCOLL$_{\text{CT}}$	.523	.520	.454	**.528**
	HYBRID	.500	.505	.416	.516

Table 4: Experiment 1: Results ($\mathcal{P}$=precision, $\mathcal{R}$=recall, $\mathcal{F}_1$=F-score, $\mathcal{A}$=accuracy)

Step	Method	Tagged	$\mathcal{A}_i$	Untagged
1	SCOLL	12	1.000	242
2	CSIM	57	.719	185
3	extSCOLL	0	.000	185
4	extCSIM	78	.462	107
5	CSIM$_{\text{REST}}$	107	.393	0

Table 5: Experiment 1: Classifications for each step of the HYBRID method (CS$_{\text{REST}}$=the final application of CS over the remaining test instances, $\mathcal{A}_i$=accuracy for classifications made at step i)

Tables 5 and 6 show the results at each step for the HYBRID and CSIM$_{\text{CT}}$ methods, respectively. As each method proceeds, the amount of tagged data increases but the classification accuracy of the system decreases, due to the inclusion of increasingly noisy training instances in the previous step. The performance of each individual relation is shown in Figure 3, which largely mirrors the findings of the systems in the original SEMEVAL-2007 task in terms of the relative difficulty to predict each of the 7 SRs.

6 Experiment 2: binary classification

In the second experiment, we performed a separate binary classification task for each of the 7 SRs, in the manner of the original SEMEVAL-2007 task. Table 7 shows the three baselines provided by the SEMEVAL-2007 organisers and performance of our

[3] In the original SEMEVAL-2007 task, there were two further categories, which incorporated the "query" with or without the sense information.

Iteration	θ	Tagged	$\mathcal{A}_i$	Untagged
1	.90	29	.897	225
2	.85	12	.750	213
3	.80	31	.613	182
4	.75	43	.535	139
5	.70	63	.540	76
6	.65	26	.346	50
7	<.65	49	.250	1

Table 6: Experiment 1: Classifications at each step of the CSim$_{\text{CT}}$ method (θ=threshold, $\mathcal{A}_i$=accuracy for classifications made at iteration i)

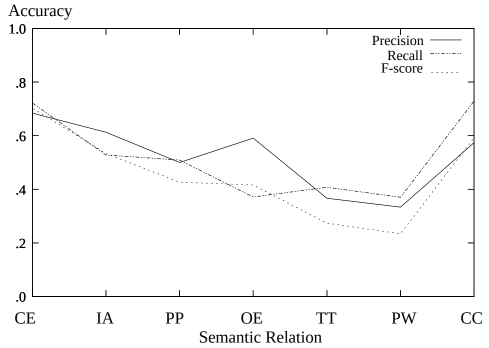

Figure 3: Experiment 1: Performance over each SR (CSim +SColl$_{\text{CT}}$ method)

6 systems. We also present the best-performing system within each group from the SemEval-2007 task. The methods for computing the baselines are described in Girju et al. (2007).

As with the first experiment, we analyzed the number of tagged instances and accuracy for the hybrid and CSim$_{\text{CT}}$ methods, as shown in Tables 8 and 9, respectively. The overall results are similar to those for the 7-way classification task.

Figures 4 and 5 show the performance for positive and negative classifications for each individual SR. The performance when the classifier outputs are mapped onto the 7-way classification task are similar to those in Figure 3.

7 Discussion and Conclusion

We compared the performance of the 6 systems in Tables 4 and 7 over the 7-way and binary classification tasks, respectively. The performance of all methods exceeded the baseline. The constituent similarity (CSim) system performed the best in group A4 and constituent similarity + SColl$_{\text{CT}}$ (CSim +SColl$_{\text{CT}}$) system performed the best in group B4 for both classification tasks. In general, the performance of constituent similarity is marginally better than that of sense collocation. Also, the utility of co-training is confirmed by it outperforming both constituent similarity and sense collocation.

Class	Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{A}$
–	All True	.485	1.000	.648	.485
–	Probability	.485	.485	.485	.517
–	Majority	.813	.429	.308	.570
A4	Best	.661	.667	.648	.660
	CSim	.632	.628	**.627**	**.650**
	CSim$_{\text{CT}}$	.615	.557	.578	.627
B4	Best	.797	.698	.724	.763
	SColl	.672	.584	.545	.634
	SColl$_{\text{CT}}$	.602	.571	.554	.619
	CSim +SColl$_{\text{CT}}$	.660	.657	**.654**	**.669**
	hybrid	.617	.568	.587	.625

Table 7: Experiment 2: Binary classification results ($\mathcal{P}$=precision, $\mathcal{R}$=recall, $\mathcal{F}_1$=F-score, $\mathcal{A}$=accuracy)

Step	Method	Tagged	$\mathcal{A}_i$	Untagged
1	SColl	21	.810	526
2	CSim	106	.689	420
3	extSColl	0	.000	420
4	extCSim	61	.607	359
5	CSim$_{\text{REST}}$	359	.619	0

Table 8: Experiment 2: Classifications for each step of the hybrid method (CS$_{\text{REST}}$=the final application of CS over the remaining test instances, $\mathcal{A}_i$=accuracy for classifications made at step i)

In order to compare the original methods with the hybrid methods, we observed that the original methods, SColl and K, and their co-training variants performed consistently better than the hybrid methods, hybrid and CSim$_{\text{CT}}$. We found that the combination of the methods lowers overall performance. We also found that the number of training instances contributes to improved performance, predictably in the sense that the methods are supervised, but encouraging in the sense that the extra training data is generated automatically. As expected, the step-wise performance of hybrid and CSim$_{\text{CT}}$ degrades with each iteration, although there were instances where the performance didn't drop from one iteration to the next (e.g. iteration 3 = 59.46% vs. iteration 4 = 72.23% in Experiment 2). This confirms

Iteration	θ	Tagged	$\mathcal{A}_i$	Untagged
1	.90	73	.726	474
2	.85	73	.726	474
3	.80	56	.714	418
4	.75	74	.595	344
5	.70	101	.722	243
6	.65	222	.572	21
7	$<.65$	21	.996	0

Table 9: Experiment 2: Classifications at each step of the CSim$_{\text{CT}}$ method (θ=threshold, $\mathcal{A}_i$=accuracy for classifications made at iteration i)

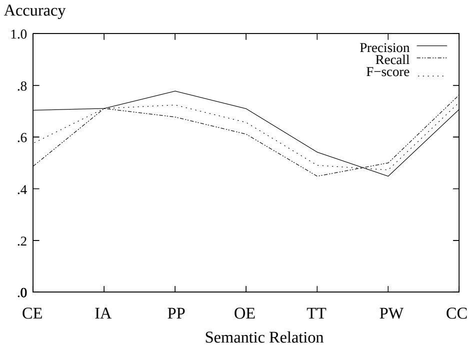

Figure 4: TPR for each SR for the binary task (positive instances, CSim +SColl$_{\text{CT}}$ method)

our expectation that: (a) the similarity threshold is strongly correlated with the quality of the resultant data; and (b) the method is susceptible to noisy training data.

Our performance comparison over the binary classification task from the SemEval-2007 task shows that our 6 systems performed below the best performing system in the competition, to varying degrees. This is partly because the methods were originally designed for multi-way (positive) classification and require adjustment for the binary task reformulation, although their performance is competitive.

Finally, comparing the SColl and CSim methods, we found that the methods interpret SRs with 100% accuracy when the sense collocations are found in both the test and training data. However, the CSim method is more sensitive than the SColl method to variation in the sense collocations, which leads to better performance. Also, the CSim method interprets NCs with high accuracy when the computed similarity is sufficiently high (e.g. with similarity ≥ 0.9 the accuracy is 89.7%). Another benefit of this method is that it interprets NCs without word sense information. As a result, we conclude that the CSim method is more flexible and robust. One possible weakness of CSim is its reliance on the similarity measure.

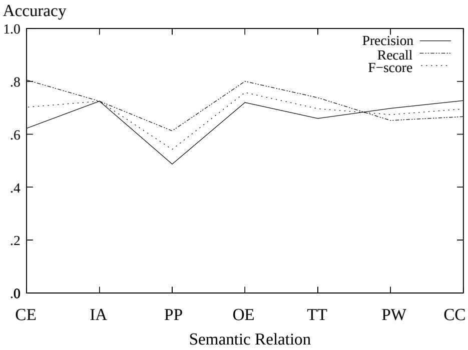

Figure 5: TNR for each SR for the binary task (negative instances, CSim +SColl$_{\text{CT}}$ method)

8 Conclusions and Future Work

In this paper, we have benchmarked and hybridised existing NC interpretation methods over data from the SemEval-2007 nominal pair interpretation task. In this, we have established guidelines for the use of the different methods, and also for the reinterpretation of the SemEval-2007 data as a more conventional multi-way classification task. We confirmed that Constituent Similarity is the best method due to its insensitivity to varied sense collocations. We also confirmed that co-training improves the performance of the methods by expanding the number of training instances.

Looking to the future, there is room for improvement for all the methods through such factors as threshold tweaking and expanding the training instances further.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pp. 805–810, Acapulco, Mexico.

Ken Barker and Stan Szpakowicz. 1998. Semiautomatic recognition of noun modifier relationships. In *Proceedings of the 17th International Conference* *on Computational Linguistics*, pp. 96–102, Montreal, Canada.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. *TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide*. ILK Technical Report 04-02.

Pamela Downing. 1977. On the Creation and Use of English Compound Nouns. *Language*, 53(4):810–842.

James Fan and Ken Barker and Bruce W. Porter. 2003. The knowledge required to interpret noun compounds. In *In Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 1483–1485.

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.

Roxana Girju. 2007. Improving the Interpretation of Noun Phrases with Cross-linguistic Information. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 568–575, Prague, Czech Republic.

Roxana Girju and Preslav Nakov and Vivi Nastase and Stan Szpakowicz and Peter Turney and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of Semantic Relations between Nominals. In *Proceedings of the 4th Semantic Evaluation Workshop (SemEval-2007)*, Prague, Czech Republic, pp.13–18.

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

Su Nam Kim and Timothy Baldwin. 2006. Interpreting Semantic Relations in Noun Compounds via Verb Semantics. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING/ACL-2006)*. pp. 491–498, Sydney, Australia.

Su Nam Kim and Timothy Baldwin. 2007. Interpreting Noun Compound Using Bootstrapping and Sense Collocation. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING)*, pp. 129–136, Melbourne, Australia.

Maria Lapata. 2002. The disambiguation of nominalizations. *Computational Linguistics*, 28(3):357–388.

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 *Proceedings of the HLT-NAACL 2004 Workshop on Computational Lexical Semantics*, pp. 60–67, Boston, USA.

Preslav Nakov and Marti Hearst. 2006. Using Verbs to Characterize Noun-Noun Relations. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)*, Bularia.

Diarmuid Ó Séaghdha and Ann Copestake. 2007. Co-occurrence Contexts for Noun Compound Interpretation. In *Proc. of the ACL-2007 Workshop on A Broader Perspective on Multiword Expressions*, Prague, Czech Republic.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.

Barbara Rosario and Hearst Marti. 2001. Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy. In *In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP-2001)*, 82–90.

Karen Sparck Jones. 1983. Compound noun interpretation problems. Computer Speech Processing, Frank Fallside and William A. Woods, Prentice-Hall, Englewood Cliffs, NJ.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *Proceedings of the 15th Conference on Computational linguistics*, pp. 782–788.

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