

# Combining Relational and Attributional Similarity for Semantic Relation Classification

**Preslav Nakov**

Department of Computer Science  
National University of Singapore  
13 Computing Drive  
Singapore 117417  
nakov@comp.nus.edu.sg

**Zornitsa Kozareva**

University of Southern California  
Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, CA 90292-6695, USA  
kozareva@isi.edu

## Abstract

We combine relational and attributional similarity for the task of identifying instances of semantic relations, such as PRODUCT-PRODUCER and ORIGIN-ENTITY, between nominals in text. We use no pre-existing lexical resources, thus simulating a realistic real-world situation, where the coverage of any such resource is limited. Instead, we mine the Web to automatically extract patterns (verbs, prepositions and coordinating conjunctions) expressing the relationship between the relation arguments, as well as hypernyms and co-hyponyms of the arguments, which we use in instance-based classifiers. The evaluation on the dataset of SemEval-1 Task 4 shows an improvement over the state-of-the-art for the case where using manually annotated WordNet senses is not allowed.

## 1 Introduction

Recently, the natural language processing (NLP) community has shown renewed interest in the problem of deep language understanding, which was inspired by the notable progress in this important research direction in the last few years. Today, lexical semantics tasks such as word sense disambiguation, semantic role labeling, and textual entailment are already well-established and are gradually finding their way in real NLP applications, while a number of new semantic tasks are emerging. One such example is the task of extracting semantic relations between nominals from text, which has attracted a lot of research attention following the creation of two benchmark datasets as part of SemEval-1 Task 4 (Girju et al., 2007) and SemEval-2 Task 8 (Hendrickx et al., 2010).

The ability to recognize semantic relations in text could potentially help many NLP applications. For example, a question answering system facing the question *What causes tumors to shrink?* would need to identify the CAUSE-EFFECT relation between *shrinkage* and *radiation* in order to be able to extract the answer from the following sentence: *The period of tumor shrinkage after radiation therapy is often long and varied.* One can also imagine a relational search engine that can serve queries such as “*find all X such that X causes wrinkles*”, asking for all entities that are in a particular relation with a given entity (Cafarella et al., 2006). Finally, modeling semantic relations has been shown to help statistical machine translation (Nakov, 2008a).

The task of identifying semantic relations in text is complicated by their heterogeneous nature. Thus, it is often addressed using non-parametric instance-based classifiers like the  $k$  nearest neighbors (kNN), which effectively reduce it to *measuring the relational similarity* between a testing and each of the training examples. The latter is studied in detail by Turney (2006), who distinguishes between *attributional similarity* or correspondence between attributes, and *relational similarity* or correspondence between relations. Attributional similarity is interested in the similarity between two *words* (or nominals, noun phrases), A and B. In contrast, relational similarity focuses on the relationship between two *pairs* of words (or nominals, noun phrases), i.e., it asks how similar the relations A:B and C:D are. Measuring relational similarity directly is hard, and thus it is rarely done directly. Instead, relational similarity is typically modeled as a function of two instances of attributional similarity: (1) between A and C, and (2) between B and D.

Going back to semantic relations, there is a similar split between two general lines of research. The first one learns the relation directly, e.g., using suitable patterns that can connect the arguments (Hearst, 1992; Turney and Littman, 2005; Nakov and Hearst, 2006; Kim and Baldwin, 2006; Pantel and Pennacchiotti, 2006; Davidov and Rappoport, 2008; Nakov, 2008b; Nakov and Hearst, 2008; Katrenko et al., 2010). This is useful for context-dependent relations like CAUSE-EFFECT, which are dynamic and often episodic in nature, e.g., *My Friday's exam causes me anxiety*. The second line focuses on the arguments, e.g., by generalizing them over a lexical hierarchy (Rosario et al., 2002; Girju et al., 2005; Kim and Baldwin, 2007; Ó Séaghdha, 2009). This works well for relations like PART-WHOLE, which are more permanent and context-independent, e.g., *door-car*.

An important advantage of argument modeling approaches is that they can benefit from many pre-existing lexical resources. For example, systems using WordNet (Fellbaum, 1998) had sizable performance gains for SemEval-1 Task 4. However, this advantage was mainly due to manually annotated WordNet senses for the relation arguments being provided for this task. There was a restricted track where using them was not allowed: this track was dominated by relation modeling approaches.

Relation and argument modeling have their strengths and weaknesses, but there have been little attempts to combine them, which is our main objective. We use no lexical resources, thus simulating a realistic real-world situation, where the coverage of any such resource is limited. Instead, we mine the Web to extract linguistic patterns expressing the relation (verbs, prepositions, and coordinating conjunctions), as well as hypernyms and co-hyponyms of its arguments. We combine (a) relational and (b) attributional similarity between (i) the first and (ii) the second argument,<sup>1</sup> using weights that are tuned separately for each individual relation.

While semantic relations can hold between different parts of speech, e.g., between a verb and a noun, we focus on relations between nominals.<sup>2</sup>

<sup>1</sup>We will call the first relation argument a *modifier* and the second one a *head*, e.g., for PART-WHOLE, the modifier will be the PART and the head will be the WHOLE.

<sup>2</sup>A nominal is a noun or a base noun phrase (NP), excluding named entities. A base NP is a noun and its premodifiers, e.g., nouns, adjectives, determiners. For example, *coffee* and *guy* are nouns, *coffee boy* is a base NP, but *the coffee guy from our office* is a complex NP and thus not a nominal.

The most relevant related publication is that of Ó Séaghdha and Copestake (2009), who combine attributional and relational features using kernels. However, they are interested in a special kind of relations: between the nouns in a noun-noun compounds like *steel knife*. Moreover, they use the British National Corpus instead of the Web, which is known to cause data sparseness issues (Lapata and Keller, 2004), they do not focus on linguistically motivated relational features such as verbs and prepositions explicitly, they use co-hyponyms but not hypernyms to generalize the relation arguments, and they give equal weights to the similarities between heads and between modifiers.

The remainder of the paper is organized as follows: Section 2 introduces our Web mining methods for argument and relation modeling, Section 3 presents our experimental setup, Section 4 discusses the results, and Section 5 concludes and points to some directions for future work.

## 2 Method

### 2.1 Overview

As we said above, we combine argument modeling and relation modeling for the task of extracting semantic relations between nominals from text.

Given the heterogeneous nature of semantic relations, we use a non-parametric instance-based classifier: kNN. This effectively reduces the task to measuring the relational similarity between a given testing example and each of the training examples: we first need to find the training example that is most similar to the target testing example; then we assume they should have the same label.

For argument modeling, we generalize the arguments of each training/testing example using a set of possible hypernyms and co-hyponyms. For example, given *the guy who makes coffee*, which is an instance of the PRODUCT-PRODUCER relation, we generate a list of potential hypernyms such as *drink* and *beverage* for *coffee*, and *person* and *human* for *guy*. We further generate co-hyponyms for the arguments, e.g., *tea* and *milk* for *coffee*, and *girl* and *boy* for *guy*. These hypernyms and co-hyponyms are extracted from the Web and there is a frequency of extraction associated with each of them, which we use to build a hypernym/co-hyponym frequency vector for each argument and for each example. We then use these *argument vectors* to measure *attributional similarity* between training and testing examples.

For relation modeling, we mine the Web to find verbs, prepositions and coordinating conjunctions that can express the typical relationship between the arguments of the target example, e.g., we generate verbs like *make* and *brew*, prepositions like *with*, and coordinating conjunctions like *and* for the arguments *guy* and *coffee* of *the guy who makes coffee*. Again, the paraphrasing verbs and prepositions and the coordinating conjunctions are extracted from the Web, and there is a frequency of extraction associated with each of them, which we incorporate into a *relational vector* and use to measure *relational similarity* between training and testing examples.

## 2.2 Argument Modeling

We model the arguments using a distribution over Web-derived hypernyms and co-hyponyms.

Multiple knowledge harvesting procedures have been proposed in the literature for the automatic acquisition of hyponyms (Hearst, 1992; Paşca, 2007; Kozareva et al., 2008) and hypernyms (Ritter et al., 2009; Hovy et al., 2009).

While we could have used any of them for our experiments, we chose the method of Kozareva et al. (2008), which (i) can extract hypernyms and hyponyms simultaneously, (ii) has been shown to achieve higher accuracy than the methods described in (Paşca, 2007; Ritter et al., 2009), and also (iii) is easy to implement. It uses a doubly-anchored pattern (DAP) of the following general form:

“*sem-class* such as *term<sub>1</sub>* and *term<sub>2</sub>*”

where *sem-class* stands for a semantic class, and *term<sub>1</sub>* and *term<sub>2</sub>* are members of this class.

In our experiments, we use the following two-placeholder form of DAP, which takes only one noun as a parameter and simultaneously extracts pairs of its hypernyms and co-hyponyms:

“\* such as *noun* and \*”

We execute the pattern against *Google*, trying both a plural and a singular form of *noun*, and we collect the returned snippets. Then, we extract the terms from the \* positions, and we build a frequency vector of hypernyms and co-hyponyms. Table 1 shows an example for *coffee guy*.

## 2.3 Relation Modeling

We model the relation itself as a distribution over Web-derived verbs, prepositions, and coordinating conjunctions that can connect the target nouns.

Frequency	Hyp./co-hyp. for arg. 1/2
311	cohyp_arg1:tea
175	hyper_arg1:beverage
102	hyper_arg1:drink
80	hyper_arg1:item
59	hyper_arg1:product
51	cohyp_arg1:chocolate
32	cohyp_arg1:cocoa
27	cohyp_arg1:soda
24	hyper_arg1:crop
22	hyper_arg1:food
21	cohyp_arg1:sugar
19	cohyp_arg1:fruit
19	hyper_arg1:stimulant
...	...
119	hyper_arg2:people
21	hyper_arg2:friend
...	...

Table 1: **Vector of hypernyms and co-hyponyms for the two arguments of *coffee guy*.**

Following Nakov and Hearst (2008), we use generalized patterns of the form:

“noun1 THAT? \* noun2”

“noun2 THAT? \* noun1”

where *noun1* and *noun2* are inflected variants of the head nouns in the relation arguments, *THAT?* stands for *that*, *which*, *who* or the empty string, and \* stands for up to eight instances<sup>3</sup> of the search engine’s star operator.

We instantiate these generalized patterns and we submit them to *Google* as exact phrase queries. We then collect the snippets for all returned results (up to 1,000). We split the extracted snippets into sentences, and we filter out all incomplete ones and those that do not contain the target nouns. We POS tag the sentences using the *Stanford POS tagger* (Toutanova et al., 2003) and we make sure that the word sequence following the second mentioned target noun is non-empty and contains at least one non-noun, i.e., that the snippet includes the entire noun phrase of the second noun in the pattern instantiation. This is because we want the second noun in the pattern instantiation to be the head of an NP: if the NP is incomplete, the second noun could be a modifier in that partial NP.

<sup>3</sup>Using multiple instances of the star operator increases the number of possible instantiations of the generalized pattern and allows extracting additional snippets.

Frequency	Paraphrase
58	V:have
54	V:make
34	V:get
32	V:sell
31	V:serve
30	V:sip
17	V:buy
16	V:want
16	V:pour
13	RV:be made by
12	V:bring
11	P:with
9	RP:from
4	C:and
...	...

Table 2: Vector of paraphrases for *coffee guy*.

We then run the *OpenNLP tools*<sup>4</sup> to shallow parse the sentences and to extract the verbs, prepositions and coordinating conjunctions connecting the two nouns. Finally, we lemmatize all extracted verbs.

As a result, we end up with quadruples, each of which includes the following: (i) a pattern, i.e., a lemmatized verb, a preposition, or a coordinating conjunction, (ii) a pattern type, i.e., V for verb, P for preposition, or C for coordinating conjunction, (iii) direction, i.e., relative order of the arguments in the pattern (R marks reverse), and (iv) frequency of extraction.

We concatenate the first three components of these quadruples to form typed directed patterns. We then build frequency vectors for them using the frequency of extraction to represent the semantics of the relation itself. Table 2 shows the resulting relational vector for *coffee guy*.

### 3 Experiments and Evaluation

In this section, we describe the dataset, the classifier, the similarity measures, and the way we combine relational and attributional similarity.

#### 3.1 Dataset

We use with the dataset from SemEval-1 Task 4 on *Classification of Semantic Relations between Nominals* (Girju et al., 2009), which is the most popular dataset for our problem; using it allows for a direct comparison to state-of-the-art systems that were evaluated on it.

<sup>4</sup>OpenNLP: <http://opennlp.sourceforge.net>

Each example in the dataset consists of a sentence annotated with two target nominals,  $e_1$  and  $e_2$ , which are to be judged on whether they are in a given target relation or not. In addition, manually annotated WordNet 3.0 senses for these nominals are provided. The Web query the task organizers used to mine the sentence from the Web is also made available.

Here is a fully annotated training example (note that, for the test examples, the "true"/"false" labels are hidden from the system):

"The production assistant is basically the <e1>guy</e1> who makes <e2>coffee</e2> and goes to the post office."

```
WordNet(e1) = "guy%1:18:00::",
WordNet(e2) = "coffee%1:13:00::",
Origin-Entity(e2, e1) = "true",
Query = "the * makes * coffee"
```

In our experiments, we ignored the WordNet senses and the Web query since having them is unrealistic for a real-world application.

Table 3 shows the seven semantic relations defined by the task along with the positive/negative instance distribution and one example instance for each relation. In SemEval-1 Task 4, each relation is considered in isolation, i.e., there are seven separate classification tasks, and there are separate training and testing datasets for each of them. For each relation, the examples are annotated with true/false labels, depending on whether they are instances of the relation. Each of the seven datasets consists of 140 training and 71-93 testing examples per relation, approximately 50% of which are positive.

#### 3.2 Classifier and Similarity Measures

Due to the small size of the individual training datasets and because of the heterogeneity of the examples, we found it hard to train a good model such as SVM or logistic regression. Therefore, we opted for a non-parametric classifier: kNN, and more precisely, 1-nearest-neighbor. Because of its sensitivity to the similarity function, we experimented with three weighting schemes: (1) frequency, (2) TF.IDF, and (3) TF.IDF with add-one smoothing for the IDF part. Each of these schemes was combined with the following *cosine* and *Dice* similarity functions:

Relation	Training Data		Test Data		Example
	positive	size	positive	size	
CAUSE-EFFECT	52.14%	140	51.25%	80	hormone (CAUSE) – growth (EFFECT)
INSTRUMENT-AGENCY	50.71%	140	48.71%	78	laser (INSTRUMENT) – printer (AGENCY)
PRODUCT-PRODUCER	60.71%	140	66.67%	93	honey (PRODUCT) – bee (PRODUCER)
ORIGIN-ENTITY	38.57%	140	44.44%	81	alcohol (ENTITY) – grain (ORIGIN)
THEME-TOOL	41.43%	140	40.84%	71	copyright (THEME) – law (TOOL)
PART-WHOLE	46.43%	140	36.11%	72	leg (PART) – table (WHOLE)
CONTENT-CONTAINER	46.43%	140	51.35%	74	pear (CONTENT) – basket (CONTAINER)

Table 3: **SemEval-1 Task 4**: The seven semantic relations defined by the task along with the distribution of positive/negative instances and one example for each relation.

$$\text{cosine}(A, B) = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (1)$$

$$\text{Dice}(A, B) = \frac{2 \times \sum_{i=1}^n \min(a_i, b_i)}{\sum_{i=1}^n a_i + \sum_{i=1}^n b_i} \quad (2)$$

We further experimented with the information-theoretic similarity measure of Lin (1998).

### 3.3 Experimental Setup

For each example in the SemEval-1 Task 4 dataset, we removed all modifiers from the target entities  $e_1$  and  $e_2$ , retaining their head nouns only; below we will still refer to them as  $e_1$  and  $e_2$  though. We then mined the Web to extract features, as described in Section 2 above:

- (1) **relational features**: verbs, prepositions, and coordinating conjunctions connecting  $e_1$  and  $e_2$  (see Table 2);
- (2) **attributinal features**: hypernyms and co-hyponyms of  $e_1$  and  $e_2$  (see Table 1).

We used the type (1) features as a baseline, and we studied the impact of combining them with type (2) features using the following five linear weights:  $w_{mod}$  for the modifier,  $w_{head}$  for the head,  $w_{rel}$  for the relation,  $w_{hyp}$  for the hypernyms, and  $w_{coh}$  for the co-hyponyms.

We tuned the values of these parameters using leave-one-out cross-validation on the development set, trying all values in [0.0; 1.0] with a step of 0.1, subject to the following two constraints:

$$\begin{aligned} w_{mod} + w_{head} + w_{rel} &= 1 \\ w_{hyp} + w_{coh} &= 1 \end{aligned}$$

These tuned weights were then used to calculate the final similarity score  $s$  as follows:

$$\begin{aligned} s &= w_{mod}s_m + w_{head}s_h + w_{rel}s_r \\ s_m &= w_{hyp}s_{hyp}(m_1, m_2) + w_{coh}s_{coh}(m_1, m_2) \\ s_h &= w_{hyp}s_{hyp}(h_1, h_2) + w_{coh}s_{coh}(h_1, h_2) \end{aligned}$$

where  $s_{hyp}(m_1, m_2)$  is the similarity between the hypernyms of the modifiers,  $s_{coh}(m_1, m_2)$  is the similarity between the co-hyponyms of the modifiers,  $s_{hyp}(h_1, h_2)$  is the similarity between the hypernyms of the heads,  $s_{coh}(h_1, h_2)$  is the similarity between the co-hyponyms of the heads, and  $s_r$  is the relational similarity.

We also did two restricted experiments: (a) with hypernyms only, i.e., setting  $w_{hyp} = 1$ , and (b) with co-hyponyms only, i.e., setting  $w_{coh} = 1$ .

## 4 Results and Discussion

Following the experimental setup for SemEval-1 Task 4, we trained and evaluated a separate system for each of the seven relations.

The macro-averaged accuracy over all relations is shown in Table 4. Several interesting observations can be made about it. First, we can see consistent improvements over the corresponding baseline for all three combined systems, for all similarity measures and for all weighting schemes, ranging from 0.5% to 19.5% absolute. Second, in 15 of the 21 experimental conditions involving attributinal patterns, the improvements over the corresponding baselines are statistically significant as measured by the  $\chi^2$  test. Third, we improve by 1.4% absolute even over our strong baseline, *Dice w/ TF.IDF, smoothed*, which achieves 68.1% accuracy. Note that this baseline is better than the best accuracy of 66.0% achieved at SemEval-1 Task 4 for systems of *type A*, which do not use the Web query or the WordNet senses (Girju et al., 2007).

Similarity measures	Baseline	+(Hyp.&Co-hyp.)		+Hypernyms		+Co-hyponyms	
	Accuracy	Accuracy	$\Delta$	Accuracy	$\Delta$	Accuracy	$\Delta$
cosine w/ frequency	62.2	*67.8	+5.5	*68.3	+6.1	* <b>68.4</b>	<b>+6.2</b>
cosine w/ TF.IDF	59.4	*69.3	+9.9	*68.6	+9.2	* <b>70.3</b>	<b>+10.9</b>
cosine w/ TF.IDF, smoothed	63.9	* <b>70.1</b>	<b>+6.2</b>	*67.8	+3.9	*69.3	+5.4
Dice w/ frequency	62.5	* <b>68.9</b>	<b>+6.4</b>	*68.1	+5.6	*67.0	+4.5
Dice w/ TF.IDF	51.8	* <b>71.3</b>	<b>+19.5</b>	*67.4	+15.6	*66.8	+15.0
Dice w/ TF.IDF, smoothed	68.1	<b>69.5</b>	<b>+1.4</b>	68.7	+0.6	69.3	+1.2
Lin's measure	66.2	68.0	+1.8	<b>68.2</b>	<b>+2.0</b>	66.7	+0.5

Table 4: **Overall macro-averaged results for all seven relations.** The baseline system uses relational patterns only, while the following systems combine relational and attributional features using linear interpolation. Shown are the accuracy and the absolute difference (in %) compared to the baseline. The highest results in each row appear in bold. Statistically significant improvements over the baseline are marked with a star.

Fourth, our best overall accuracy of 71.3% represents a statistically significant improvement not only over our corresponding baseline of 51.8% but also over the best result of 66.0% achieved at SemEval-1 Task 4 for systems of *type A*. It is also higher (but no statistically significant difference) than the state-of-the-art result of Davidov and Rappoport (2008), who achieved 70.1%.

The evaluation results for each of the seven individual relations are shown in Table 5. We can see that not all relations benefit equally well from using attributional patterns in addition to relational ones. The most sizable improvements are for THEME-TOOL, which shows statistically significant improvements for all evaluation measures, ranging from +7.1% to +23.9% absolute. Very large consistent improvements can be also observed for PRODUCT-PRODUCER and ORIGIN-ENTITY. The results are somewhat mixed for relations like CAUSE-EFFECT, CONTENT-CONTAINER, INSTRUMENT-AGENCY and PART-WHOLE; still, the improvements are more sizable than the decreases.

We can further see that relations like THEME-TOOL and ORIGIN-ENTITY are best characterized by the properties of their arguments, which makes them a good fit for attributional methods. In contrast, relations like INSTRUMENT-AGENCY and PRODUCT-PRODUCER, are better expressed by patterns: verbs, prepositions and coordinations.

The weights in Table 5 suggest that, overall, the co-hyponyms are more important than the hypernyms, and the relations are typically determined primarily by the modifier and the relational similarity. There is also a lot of variety for the individual relations. For example, for THEME-TOOL, it is the head that matters most.

Note that for two of the relations, we achieve results that are better than the best results achieved at SemEval-1 Task 4, even by systems that used WordNet and the original search engine query. In particular, for ORIGIN-ENTITY, we achieve up to 77.8% accuracy, which is statistically significantly better than the 72.8% at SemEval-1 Task 4. We also improve for THEME-TOOL, but our 74.7% is only marginally better than 74.6%.

## 5 Conclusion and Future Work

We have studied the combination of relational and attributional similarity for the task of semantic relation classification in text. Using the dataset for SemEval-1 Task 4, we have shown statistically significant improvements over a strong baseline that uses relational similarity only, and even a small improvement over the state-of-the-art. We have further studied the extent of the improvement across seven individual relations.

In future work, we plan to do a similar study for the dataset for SemEval-2 Task 8, where, given its size and the specifics of the relation definitions, which are much more context-dependent, we will need to model the local context, in addition to relational and attributional similarity measures.

## Acknowledgments

This research is partially supported (for the first author) by the *SmartBook project*, funded by the Bulgarian National Science Fund under Grant D002-111/15.12.2008.

We would like to thank the anonymous reviewers for their detailed and constructive comments, which have helped us improve the paper.

	Baseline	Accuracy	$\Delta$	$w_{mod}$	$w_{head}$	$w_{rel}$	$w_{hyp}$	$w_{coh}$
<b>CAUSE-EFFECT</b>								
cosine w/ frequency	66.3	65.0	-1.3	1.0	0.0	0.0	0.1	0.9
cosine w/ TF.IDF	62.5	*70.0	+7.5	0.5	0.0	0.5	0.4	0.6
cosine w/ TF.IDF, smoothed	67.5	68.8	+1.3	0.6	0.0	0.4	0.4	0.6
Dice w/ frequency	63.7	65.0	+1.3	0.6	0.1	0.3	0.1	0.9
Dice w/ TF.IDF	68.8	68.8	0.0	0.9	0.0	0.1	0.1	0.9
Dice w/ TF.IDF, smoothed	71.3	70.0	-1.3	0.6	0.0	0.4	0.3	0.7
Lin's measure	68.8	66.3	-2.6	0.0	0.0	1.0	0.0	1.0
<b>INSTRUMENT-AGENCY</b>								
cosine w/ frequency	67.9	71.8	+3.9	0.0	0.4	0.6	0.5	0.5
cosine w/ TF.IDF	62.8	*70.5	+7.7	0.0	0.1	0.9	1.0	0.0
cosine w/ TF.IDF, smoothed	73.1	70.5	-2.6	0.1	0.1	0.8	0.0	1.0
Dice w/ frequency	67.9	66.7	-1.2	0.2	0.1	0.7	0.6	0.4
Dice w/ TF.IDF	56.4	*69.2	+12.8	0.0	0.4	0.6	1.0	0.0
Dice w/ TF.IDF, smoothed	61.5	65.4	+3.9	0.1	0.2	0.7	0.1	0.9
Lin's measure	61.5	55.1	-6.4	0.1	0.0	0.9	0.0	1.0
<b>PRODUCT-PRODUCER</b>								
cosine w/ frequency	58.1	*65.6	+7.5	0.2	0.7	0.1	0.5	0.5
cosine w/ TF.IDF	57.0	*72.0	+15.0	0.3	0.0	0.7	0.4	0.6
cosine w/ TF.IDF, smoothed	60.2	*71.0	+10.8	0.1	0.2	0.7	0.1	0.9
Dice w/ frequency	62.4	*66.7	+4.3	0.7	0.1	0.2	0.0	1.0
Dice w/ TF.IDF	58.1	*73.1	+15.0	0.5	0.1	0.4	0.3	0.7
Dice w/ TF.IDF, smoothed	68.8	72.0	+3.2	0.2	0.1	0.7	0.8	0.2
Lin's measure	74.2	74.2	0.0	0.2	0.2	0.6	0.2	0.8
<b>ORIGIN-ENTITY</b>								
cosine w/ frequency	56.8	*72.8	+16.0	0.5	0.5	0.0	0.3	0.7
cosine w/ TF.IDF	55.6	*70.4	+14.8	0.3	0.2	0.5	0.2	0.8
cosine w/ TF.IDF, smoothed	66.7	*71.6	+4.9	0.2	0.0	0.8	0.0	1.0
Dice w/ frequency	58.0	*74.1	+16.1	0.5	0.5	0.0	0.1	0.9
Dice w/ TF.IDF	50.6	*77.8	+27.2	0.5	0.5	0.0	0.1	0.9
Dice w/ TF.IDF, smoothed	69.1	71.6	+2.5	0.3	0.2	0.5	0.1	0.9
Lin's measure	60.5	*69.1	+8.6	0.5	0.4	0.1	0.2	0.8
<b>THEME-TOOL</b>								
cosine w/ frequency	54.9	*69.0	+14.1	0.1	0.9	0.0	0.2	0.8
cosine w/ TF.IDF	47.9	*69.0	+21.1	0.0	1.0	0.0	0.3	0.7
cosine w/ TF.IDF, smoothed	56.3	*74.7	+18.4	0.1	0.9	0.0	0.4	0.6
Dice w/ frequency	57.7	*64.8	+7.1	0.2	0.8	0.0	0.3	0.7
Dice w/ TF.IDF	42.3	*66.2	+23.9	0.0	0.8	0.2	0.4	0.6
Dice w/ TF.IDF, smoothed	62.0	*71.8	+9.8	0.0	1.0	0.0	0.3	0.7
Lin's measure	54.9	*67.6	+12.7	0.1	0.9	0.0	0.1	0.9
<b>PART-WHOLE</b>								
cosine w/ frequency	72.2	63.9	-8.3	0.8	0.0	0.2	0.8	0.2
cosine w/ TF.IDF	70.8	68.1	-2.7	0.6	0.0	0.4	0.9	0.1
cosine w/ TF.IDF, smoothed	62.5	*68.1	+5.6	0.5	0.0	0.5	0.3	0.7
Dice w/ frequency	70.8	*80.6	+9.8	0.5	0.0	0.5	0.3	0.7
Dice w/ TF.IDF	40.3	*80.6	+40.3	0.7	0.0	0.3	0.7	0.3
Dice w/ TF.IDF, smoothed	75.0	75.0	0.0	0.0	0.0	1.0	0.0	1.0
Lin's measure	69.4	72.2	+2.8	0.2	0.0	0.8	0.9	0.1
<b>CONTENT-CONTAINER</b>								
cosine w/ frequency	59.5	*66.2	+6.7	0.8	0.1	0.1	0.5	0.5
cosine w/ TF.IDF	59.5	*64.9	+5.4	0.0	0.0	1.0	0.0	1.0
cosine w/ TF.IDF, smoothed	60.8	*66.2	+5.4	0.0	0.1	0.9	0.0	1.0
Dice w/ frequency	56.8	*64.9	+8.1	0.3	0.0	0.7	0.4	0.6
Dice w/ TF.IDF	45.9	*63.5	+17.6	0.5	0.2	0.3	0.7	0.3
Dice w/ TF.IDF, smoothed	68.9	60.8	-8.1	0.6	0.0	0.4	1.0	0.0
Lin's measure	74.3	71.6	-2.7	0.5	0.1	0.4	0.7	0.3

Table 5: **Results for the individual relations.** The baseline uses relational patterns only; the rest combine relational and attributional patterns for hypernyms & co-hyponyms. Shown are the accuracy and the absolute difference (in %) compared to the baseline. Statistically significant improvements are marked with a star.

## References

- Michael Cafarella, Michele Banko, and Oren Etzioni. 2006. Relational Web search. Technical Report 2006-04-02, University of Washington, Department of Computer Science and Engineering.
- Dmitry Davidov and Ari Rappoport. 2008. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of ACL*, pages 227–235.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Comput. Speech Lang.*, 4(19):479–496.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of SemEval*, pages 13–18.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2009. Classification of semantic relations between nominals. *Lang. Resour. Eval.*, 43(2):105–121.
- Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, pages 539–545.
- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval*, pages 33–38.
- Eduard Hovy, Zornitsa Kozareva, and Ellen Riloff. 2009. Toward completeness in concept extraction and classification. In *Proceedings of EMNLP*, pages 948–957.
- Sophia Katrenko, Pieter W. Adriaans, and Maarten van Someren. 2010. Using local alignments for relation recognition. *J. Artif. Intell. Res.*, 38:1–48.
- Su Nam Kim and Timothy Baldwin. 2006. Interpreting semantic relations in noun compounds via verb semantics. In *Proceedings of COLING/ACL*, pages 491–498.
- Su Nam Kim and Timothy Baldwin. 2007. Interpreting noun compounds via bootstrapping and sense collocation. In *Proceedings of PAFLING*, pages 129–136.
- Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL*, pages 1048–1056.
- Mirella Lapata and Frank Keller. 2004. The Web as a baseline: Evaluating the performance of unsupervised Web-based models for a range of NLP tasks. In *Proceedings of HLT-NAACL*, pages 121–128.
- Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304.
- Preslav Nakov and Marti Hearst. 2006. Using verbs to characterize noun-noun relations. In *Proceedings of AIMSA*, pages 233–244.
- Preslav Nakov and Marti Hearst. 2008. Solving relational similarity problems using the web as a corpus. In *Proceedings of ACL*, pages 452–460.
- Preslav Nakov. 2008a. Improved statistical machine translation using monolingual paraphrases. In *Proceedings of ECAI*, pages 338–342.
- Preslav Nakov. 2008b. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Proceedings of AIMSA*, pages 103–117.
- Diarmuid Ó Séaghdha and Ann Copestake. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of EACL*, pages 621–629.
- Diarmuid Ó Séaghdha. 2009. Semantic classification with WordNet kernels. In *Proceedings of HLT-NAACL*, pages 237–240.
- Marius Paşca. 2007. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *Proceedings of WWW*, pages 101–110.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of ACL*, pages 113–120.
- Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of AAAI Spring Symposium*, pages 88–93.
- Barbara Rosario, Marti Hearst, and Charles Fillmore. 2002. The descent of hierarchy, and selection in relational semantics. In *Proceedings of ACL*, pages 247–254.
- Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259.
- Peter Turney and Michael Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.
- Peter Turney. 2006. Similarity of semantic relations. *Comput. Linguist.*, 32(3):379–416.