Finding Constraints for Semantic Relations via Clustering

Sophia Katrenko and Pieter Adriaans

University of Amsteram, the Netherlands

Abstract

Automatic recognition of semantic relations constitutes an important part of information extraction. Many existing information extraction systems rely on syntactic information found in a sentence to accomplish this task. In this paper, we look into relation arguments and claim that some semantic relations can be described by constraints imposed on them. This information would provide more insight on the nature of semantic relations and could be further combined with the evidence found in a sentence to arrive at actual extractions.

1 Introduction

Semantic relations have been an object of study for a long time and across different disciplines (Khoo and Na 2006). Within computational linguistics, the main focus has been on identifying relations automatically (McDonald 2005) and further use of the extracted relations in various applications to improve their performance (van der Plas 2008).

When talking about relations, we distinguish between their extension and intension. For the n-ary relation its extension is determined by the set of ordered entities (of size n) that satisfy it. For example, for the relation PART - WHOLE this set would have a member <professor, faculty> but not <faculty, professor>. The intension of a relation is defined by what it means (what does it mean that x is part of y?). More generally, one can represent the extension of an n-ary relation by a subset the Cartesian product of the sets S_1, ..., S_n where each set corresponds to the particular argument of the relation. If two entities x and y are in the binary relation $\mathcal{R}$, we write $x\mathcal{R}y$ or $\mathcal{R}(x, y)$. A set of ordered entities that satisfy $\mathcal{R}$ (for instance, $< x, y >$) is referred to as instances of $\mathcal{R}$ or its mentions. Note that the same extensions do not necessarily mean the same intension. Two relations such as STUDY-IN and LIVE-IN may have the same extension but their intension is different.

Past research has led to studying similarity effects that can be imposed on relations. For instance, Chaffin and Herrmann (2001) distinguish between *item similarity* and *relation similarity* whereby the first measure, in contrast to the second, is constant and does not depend on the relation at hand. More precisely, if one is given a word pair $< x, y >$, item similarity is defined between arguments x and y, while relation similarity measures how close $< x, y >$ is to the target relation. Item similarity plays a significant role only for some relations like SYNONYMY or ANTONYMY where such similarity effects are clearly involved. Chaffin and

Proceedings of the 20th Meeting of Computational Linguistics in the Netherlands
Edited by: Eline Westerhout, Thomas Markus, and Paola Monachesi.

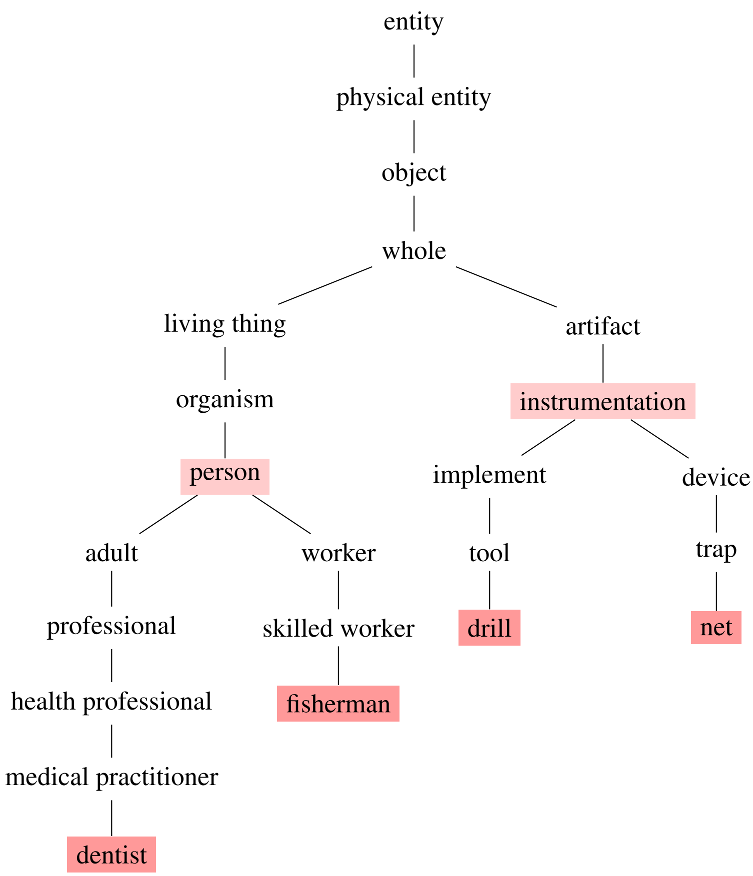

Figure 1: Part of the WordNet hierarchy.

Herrmann (2001) have studied yet another relation, PART-WHOLE, and showed that even though a part is not necessarily similar to a whole, there are still effects similar to those reported on other semantic relations. The authors concluded that relation similarity facilitates relation recognition and impedes negative response. As expected, item similarity did not contribute much to the recognition task and when held constant, relation similarity still affected the resulting performance.

While item similarity may not be applicable to all types of semantic relations, one can assume that argument fillers can be grouped according to their similarity. Consider, for example, the following two sentences.

(4.1) I saw a *fisherman* cleaning his *net*.

(4.2) One of the instruments a *dentist* uses often is a *drill*.

Here, we have two examples of the INSTRUMENT - AGENCY relation, <fisherman, net> and <dentist, drill>. If we look at the fragment of the WordNet hierarchy depicted in Figure 1, we could arrive at the generalization of these examples, such as <person, instrumentation>. It can serve as a semantic constraint for INSTRUMENT - AGENCY, because knowing that there is a person and an instrument we may conclude that it is likely for a person to use an instrument.

In this paper, we propose a method to derive constraints for semantic relations. In other words,

> *Given positive examples of a binary relation $\mathcal{R}(x, y)$ and a taxonomy $\mathcal{T}$, our goal is to find all possible pairs $< \mathcal{G}_x, \mathcal{G}_y >$ such that $\mathcal{G}_x$ is a semantic type of x and $\mathcal{G}_y$ is a semantic type of y, respectively.*

The paper is structured as follows. We introduce the method in Section 2 and proceed with the description of seven generic semantic relations (Section 3.1) and the experimental results (Section 3.2). In Section 4, we discuss our findings in more detail.

2 Methods

To find constraints for semantic relations, we describe a method which is based on positive examples only and does not make use of negative examples. To determine generalized semantic types of relation arguments, one has to be able to form clusters based on the existing information. Such clusters can be formed using semantic measures defined over WordNet (Fellbaum 1998). In particular, given argument x of $\mathcal{R}(x, y)$ and n corresponding synsets collected from the training data set $((x_1, y_1),\ldots,(x_n, y_n))$, we create a matrix S of size $n \times n$ by comparing each pair of synsets (s_i, s_j), $i = 1 \ldots n$. Each element of this matrix is equal to a similarity score of (s_i, s_j), with the diagonal elements equal to 1 (we assume that a similarity measure returns values from 0 to 1 with 1 being an identity score).

To form the matrices that can be used for clustering, we have to compare each pair of synsets for x and do the same for argument y. There exists a range of similarity measures that allow to compare a pair of synsets (Budanitsky and Hirst 2006). For our purpose, we selected wup measure which uses a notion of path length between two synsets. Given two synsets s_1 and s_2 connected by a path of length $len(s_1, s_2)$ and their least common subsumer $LCS(s_1, s_2)$, the wup score is calculated as follows (Palmer and Wu 1995):

$$wup(s_1, s_2) = \frac{2 * depth(LCS(s_1, s_2))}{len(s_1, s_2) + 2 * depth(LCS(s_1, s_2))} \tag{4.3}$$

Once a matrix S is obtained, we perform clustering. Ideally, the resulting clusters should reflect the semantic types of a given argument. However, to be able to use such clusters in future, we have to label them. This can be accomplished by using the least common subsumer. For each cluster C, $c_i \in C, i = 1, \ldots, k$,

$LCS(c_1, \ldots, c_k)$ is computed. This LCS corresponds to the semantic type $\mathcal{G}_x$ we are looking for.

Definition 1 (Least Common Subsumer)**.** *Given two nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ in a taxonomy $\mathcal{T}$, their least common subsumer LCS is an ancestor of both $\mathcal{N}_1$ and $\mathcal{N}_2$ such that there is no node $\mathcal{C}$ which is ancestor of $\mathcal{N}_1$ and $\mathcal{N}_2$ and a child of LCS.*

Recall that such generalization is done per argument and we need to find pairs of clusters that would correspond to $R(x, y)$. Let l be a number of clusters for x and m be a number of clusters for y. To find cluster pairs, we introduce a strength coefficient between any pair of clusters as follows. For each cluster $C_i, i = 1, \ldots, l$ for the argument x, and for each cluster $C_j, j = 1, \ldots, m$ for the argument y, the strength coefficient $s(C_i, C_j)$ is calculated in the following way:

$$s(C_i, C_j) = \frac{\#links(C_i, C_j)}{min(|C_i|, |C_j|)} \tag{4.4}$$

In Equation 4.4, $\#links(C_i, C_j)$ stands for the number of links between elements of C_i and C_j. It is easy to see that if a mapping from C_i to C_j is injective, the strength coefficient can be at most 1 (all elements of one cluster are connected with some/all elements of the other) and at least 0 (there are no elements in both clusters that are connected with each other). If a mapping is surjective, then s can be larger than 1.

Using clustering to detect semantic types of arguments poses a problem of defining a number of resulting clusters. If the number of clusters is high, we expect to obtain specific generalizations and high precision/low recall. Reducing a number of clusters will most likely lead to less precise generalizations but higher recall.

There exist many clustering methods and it is clear that a choice of a clustering method may affect the results. However, we abstract away from a clustering approach by choosing a simple agglomerative method (Zhao and Karypis 2002).

The shortest path Provided that semantic constraints are identified with high recall, one may combine this outcome with syntactic evidence. One way of doing this is to consider all positive predictions by both syntax-based method and semantic constraints as positive in the final model, while the rest should be labeled as negative examples. To obtain predictions based on syntactic information, we use the shortest path kernel, which represents a kernel-based approach for relation extraction and explores information found in dependency trees (Bunescu and Mooney 2005). As input for this method serve dependency paths connecting two relation arguments. The more similar these paths are, the more likely two relation examples belong to the same category. Given the data sparseness problem, the authors generalize over existing paths by adding information sources, such as part of speech (PoS) categories or named entity types.

The shortest path between relation arguments is extracted and a kernel between two sequences (paths) $\mathbf{x} = \{x_1, \ldots, x_n\}$ and $\mathbf{x}' = \{x'_1, \ldots, x'_m\}$ is computed as follows:

$$k_B(\mathbf{x}, \mathbf{x}') = \begin{cases} 0 & m \neq n \\ \prod_{i=1}^{n} f(x_i, x'_i) & m = n \end{cases} \tag{4.5}$$

In Equation 4.5, $f(x_i, x'_i)$ is the number of features shared by x_i and x'_i. The features which are used as input are the following: word (e.g., $protesters$), part of speech tag (e.g., NNS), generalized part of speech tag (e.g., $Noun$), and entity type (e.g., $PERSON$) if applicable. In addition, a direction feature ($\rightarrow$ or $\leftarrow$) is employed.

3 Evaluation

In this section, we will describe seven semantic relations used in the experimental set-up and elaborate on the results on both training and test data sets of the SEMEVAL-2007 competition (Girju et al. 2007).

3.1 Data

For semantic type detection, we use 7 binary relations from the training set of the SEMEVAL-2007 competition, all definitions of which share the requirement of the syntactic closeness of the arguments. Further, they have various restrictions on the nature of the arguments. The definitions of the relation types together with the restrictions imposed on them are reproduced below (based on the SEMEVAL-Task 4 definitions).

CAUSE - EFFECT(X,Y) takes place if, given a sentence S, it is possible to entail that X is the cause of Y. X, Y can each be a nominal denoting an occurrence (e.g., event, state, activity), or a noun denoting an entity, as a metonymic expression of an occurrence. In case an effect is caused by a combination of events, each such event is considered a separate cause for the effect. Indirect causation is considered positive, e.g. CAUSE-EFFECT(earthquake, aftershock).

INSTRUMENT - AGENCY(X,Y) is true if the situation described in S entails the fact that X is the instrument of Y (Y uses X). Further, X is an entity and Y is an explicit actor or an implied activity (there exists an activity even if the close context for X and Y includes no verb). The relation is true if the sentence context implies that Y uses X, Y has used X, or X will likely use Y in the future.

PRODUCT - PRODUCER(X,Y) is true if the situation described in S entails the fact that X is a product of Y, or Y produces X. The producer should be actively involved in the process of bringing the product into existence and not just serve as a raw material. The product can be any abstract or concrete object.

ORIGIN - ENTITY(X,Y) is true if the situation described in S entails that X is the origin of Y. Y is the entity derived from the origin. The origin can be spatial/geographical or material but it should not be actively involved in the process of bringing the entity into existence (e.g., "light bulb"). The entity should not be part of the origin, e.g. "apple seed", when the "seed" is separated from the "apple". In the case of material origin X, X should undergo considerable processing in order to produce Y. A person/company can be identified as origins if they were not involved in the production of the entity. Objects emitting radiation/heat/light are regarded as producers of such emissions, not just origins. An entity can have several origins, and each of them separately will count as an origin. News and information is conveyed, rather than produced, and its source will be the origin.

THEME - TOOL(X,Y) is true if the situation described in S entails the fact that Y (the tool) is (or was) intended (or designed or used) for some kind of action (V-ing, where V is some verb) in which X (the theme) is the thing that is acted upon (the object of the verb V) or the result of the action. X (the theme) should be an object (e.g., "*wine* glass"), an event ("*concert* hall"), a state of being ("*migraine* drug"), an agent ("*artist* award") or a substance ("*water* filtration"). Y (the tool) should be an object (e.g., "migraine *drug*"), an action (e.g., "service *charge*"), an agent ("military *police*"), or a substance ("salad *oil*"). Psychological features are not allowed as tools (e.g., "death *wish*"). The theme and tool must be two completely different and separate things. Plans, missions, strategies, advice, proposals, methods, process, and similar things are not allowed as tools. Requirements, groundwork, foundations, preliminaries, preconditions, and similar things are not allowed as tools for the theme either.

PART - WHOLE(X,Y), where X is part of Y and this relation can be one of the following five types: Place-Area, Stuff-Object, Portion-Mass, Member-Collection and Component-Integral object.

CONTENT - CONTAINER(X,Y) takes place when a sentence S entails the fact that X is (or was) stored (or carried) inside Y. Moreover, X is not a component of Y and can be removed from it. The container must be clearly delineated in space (sea or cloud are locations rather than containers). There is strong preference against treating legal entities (people and institutions) as content. There is weak preference against treating buildings and vehicles as containers.

Table 4.1 shows a number of training/test examples per each relation type and a number of positive instances per relation (+(train) and +(test)) used in the official SEMEVAL-Task 4 competition. All sentences were collected by quering the Web with some hand-written patterns (Hearst 1992, Nakov 2008). It was assumed that using patterns would result in the extraction of not only positive instances, but also negative ones (near misses). We removed examples unannotated with WordNet 3.0 from the training and test data sets while conducting this experiment (column 5). CONTENT - CONTAINER turned out to be the only relation type whose examples are fully annotated. In contrast, PRODUCT - PRODUCER is a relation type with the most information missing (9 examples removed).

Ideally, we would expect constraints to be of the following types:

1. Both arguments have a very specific type

2. One of the arguments is specific, whereas the other allows for a wider range

of semantic types

The third option where both arguments are general does not seem to be appropriate because in such cases it would be difficult to discriminate between different relation types. If we consider such relations as PRODUCT-PRODUCER or CONTENT-CONTAINER, they seem to fall in the second category. For instance, in a typical scenario, a producer would be a human being, while a product can be anything (e.g., a thought, an idea, a table). Similarly, a container in the CONTENT-CONTAINER relation is most likely of the limited type but content may vary substantially.

relation type	all (train)	+(train)	+(test)	+(test, a/w WordNet)
ORIGIN - ENTITY	140	54	81	77
PRODUCT - PRODUCER	140	85	93	84
THEME - TOOL	140	58	71	66
INSTRUMENT - AGENCY	140	71	78	74
PART - WHOLE	140	65	72	71
CONTENT - CONTAINER	140	65	74	74
CAUSE - EFFECT	140	73	80	74

Table 4.1: Distribution of the SEMEVAL examples.

We hypothesize that ORIGIN-ENTITY and THEME-TOOL are the relation types which may require sentential information to be detected. These two relations allow a greater variety of arguments and semantic information alone might be not sufficient. Such relation types as PRODUCT-PRODUCER or INSTRUMENT-AGENCY are likely to benefit more from the knowledge found in ontologies.

3.2 Experiments

To conduct experiments, we collected arguments of all positive examples from the training set and clustered them. In total, there are 14 clustering solutions (2 solutions per relation). An optimal number of clusters is not known in advance and for this reason we set it to 3, 5 and 10. Further, the strength coefficient from Section 2 was used to determine pairs of clusters that cover the training data the best. On the one hand, we may expect a pair of clusters with the strength coefficient 1 to yield 100% precision on the training data but this assumption is, however, misleading. As we use LCS on cluster's elements, it may lead to a very general concept in the WordNet hierarchy and, consequently, to lower precision. On the other hand, if all resulting pairs of clusters are used (as long as the strength coefficient is larger than zero), we should reach 100% recall. The most desirable solution is a pair of clusters that has a high strength coefficient, has a good coverage and whose clusters describe reasonably general concepts.

Selecting the pairs of clusters that should ideally lead to the best performance in the future can be done by by fixing the strength coefficient and by doing so, restricting ourselves to a subset of clustering pairs. To determine which strength coefficient is the best for a given semantic relation, either the highest accuracy or the F_1 score can be used. The general tendency that one can observe in Figure 2, 3, and 4 is that by increasing the strength coefficient recall increases while precision drops. The X-axis in these figures has to be read as follows. The values there indicate which subset of the entire set of cluster pairs is used. For instance, '0.8' means that all cluster pairs that have the strength coefficient larger than 0.8 are employed. Note that the highest F_1 is not necessarily to be found on the intersection of the precision and recall curves. This happens due to the fact that the strength value of 1.00 does not guarantee the highest precision (as explained above). Table 4.2 shows which values of s were selected per relation and what the performance on the training set is. The solutions with 10 clusters turned out to be the best for all seven relations. Two semantic relations whose scores substantially differ from the rest are THEME - TOOL and ORIGIN - ENTITY.

Relation type	Precision	Recall	Accuracy	s
ORIGIN - ENTITY	56.8	98.0	71.1	wup10, 0.71
CONTENT - CONTAINER	69.2	100	79.6	wup10, all
CAUSE - EFFECT	74.0	100	81.6	wup10, all
INSTRUMENT - AGENCY	84.6	82.1	83.6	wup10, 1.0
PRODUCT - PRODUCER	73.3	94.9	75.4	wup10, 0.29
THEME - TOOL	55.0	88.0	67.4	wup10, 0.29
PART - WHOLE	76.7	87.5	81.8	wup10, 0.5
avg.	69.9	92.9	77.2	

Table 4.2: Performance on the SEMEVAL training data set, where s stands for the strength coefficient.

The results on the test set are given in Table 4.3. Here, we can observe the same tendency as on the training data, namely, CAUSE - EFFECT and INSTRUMENT - AGENCY are among the relations with the highest scores, while THEME - TOOL and ORIGIN - ENTITY belong to semantic relations that cannot be easily identified by using the rules that we derived. It is worthwhile to recall that some constraints are consistent with the part of the test data and not necessarily with all examples. If syntactic context is not used, they are bound to extract false positives. Semantic relations that we consider here can be roughly divided into two groups, where CAUSE - EFFECT, CONTENT - CONTAINER, PRODUCT - PRODUCER, INSTRUMENT - AGENCY, PART - WHOLE form a group of relations that can be relatively easy identified solely on the semantic types of the arguments, whereas ORIGIN - ENTITY and THEME - TOOL cannot. Some constraints found by our method are listed in Table 4.4.

Relation type	Precision	Recall	Accuracy
ORIGIN - ENTITY	51.2	61.8	57.1
CONTENT - CONTAINER	70.3	68.4	68.9
CAUSE - EFFECT	68.4	72.2	70.3
INSTRUMENT - AGENCY	77.8	56.8	70.3
PRODUCT - PRODUCER	73.3	80.0	67.9
THEME - TOOL	30.7	33.3	48.5
PART - WHOLE	58.3	53.8	69.0
avg.	61.4	60.9	64.6

Table 4.3: Performance on the SEMEVAL test data set.

Relation	$(\mathcal{G}_x, \mathcal{G}_y)$
INSTRUMENT - AGENCY	(unit#6, person#1)
	(unit#6, medical_man#1)
CAUSE - EFFECT	(event#1, human_action#1)
	(knowledge#1, human_action#1)
	(event#1, state#2)
	(phenomenon#1, physical_process#1)
PRODUCT - PRODUCER	(object#1, person#1)
	(object#1, group#1)
	(matter#3, physical_entity#1)
	(communication#2, person#1)
ORIGIN - ENTITY	(group#1, object#1)
	(object#1, object#1)
	(object#1, person#1)
THEME - TOOL	(abstract_entity#1, event#1)
	(knowledge#1, abstract_entity#1)
	(event#1, communication#2)
PART - WHOLE	(person#1, group#1)
	(body_part#1, thing#12)
	(substance#1, matter#3)
	(person#1, person#1)

Table 4.4: Some constraints per relation type.

Combining semantic and syntactic evidence

In the previous section, we have shown that if accurately generated, semantic types should provide high precision. In the previous work, we have also noticed that the shortest path introduced by Bunescu and Mooney (2005) usually boosts recall (Katrenko et al. 2010). If we put two pieces of evidence together, we might expect better performance.

Clearly, there are several relations that are likely to benefit from use of semantic types, e.g., CAUSE - EFFECT, INSTRUMENT - AGENCY and PRODUCT -

PRODUCER. For these relations, recall is high (Table 4.5) and this means that accurate constraints would hopefully increase precision without significant decrease in recall. For other relation types, one might attain better performance but this should mostly affect results only slightly because recall by shortest path method is already low.

Relation type	Precision	Recall	Accuracy
ORIGIN - ENTITY	40.0	16.7	51.9
CONTENT - CONTAINER	50.0	23.7	48.6
CAUSE - EFFECT	54.7	100.0	57.5
INSTRUMENT - AGENCY	53.9	92.1	57.7
PRODUCT - PRODUCER	69.9	93.6	68.8
THEME - TOOL	56.3	31.0	62.0
PART - WHOLE	42.9	23.1	61.1
avg.	52.5	54.3	58.2

Table 4.5: Shortest path kernel's results per relation type.

Relation type	Precision	Recall	Accuracy
ORIGIN - ENTITY	57.1	11.1	56.8
CONTENT - CONTAINER	70.0	18.4	54.1
CAUSE - EFFECT	72.1	75.6	72.5
INSTRUMENT - AGENCY	80.0	41.7	70.5
PRODUCT - PRODUCER	77.4	77.4	69.9
THEME - TOOL	50.0	10.3	64.8
PART - WHOLE	60.0	11.5	65.3
avg.	66.7	35.1	64.8

Table 4.6: Results on the SEMEVAL test data set achieved by combining syntactic and semantic evidence.

We use the Stanford parser [1] to analyze data, and combined syntactic and semantic evidence as discussed in Section 2. If semantic types cannot be applied to the test data (because of the lack of annotations with WordNet synsets), predictions by shortest path kernel are used. The results of such combination are presented in Table 4.6. As expected, combination of the pieces of evidence boosts performance for CAUSATION, PRODUCT-PRODUCER and INSTRUMENT-AGENCY but also has a positive influence on other relations in terms of accuracy.

4 Discussion

Using clustering in the method we have proposed requires making some additional decisions, e.g., how many resulting clusters should be considered. To estimate the number of clusters, we use either precision or accuracy on the training data set. Our approach also depends on positive examples in the training set and on the semantic

[1] `http://nlp.stanford.edu/software/lex-parser.shtml`

hierarchy we use. If some parts of the hierarchy are more flat, the resulting patterns may be too general.

Our method has some similarities with the methods by Moldovan et al. (2004) proposed in the past. For instance, we also employ the idea of restricting the WordNet hierarchy, but it is done in the different way and for a different purpose. We do not contrast one semantic relation against the other but rather look for the semantic types of the argument per semantic relation.

Recall also the work by Chaffin and Herrmann (2001) who studied different similarity effects on semantic relations. In contrast to their proposal, we do not rely on similarity between two argument fillers but rather detect similarity on the paradigmatic level. Item similarity (or in our case similarity between two arguments of the same relation) is undoubtedly important for SYNONYMY but we believe that it is nearly useless if other semantic relations are considered.

Semantic constraints and the SEMEVAL guidelines Constraints yielded by our method largely correspond to the relation description given in the guidelines of the SEMEVAL competition. For instance, for CONTENT-CONTAINER containers are often deliniated in space as required by the guidelines. Similarly to what was stated in the SEMEVAL guidelines, we obtained generalizations for several subtypes of MERONYMY. The most successful generalizations were of the type MEMBER-COLLECTION (e.g. <person#1, group#1> in Table 4.4). Most pairs that were judged as good candidates for INSTRUMENT-AGENCY and PRODUCT-PRODUCER relations correspond to the definitions of these relations as well. In particular, the constraints detected for INSTRUMENT-AGENCY have an actor as a type for Agency and an entity for Instrument. In line with the definition of PRODUCT-PRODUCER, products can be either abstract or concrete entities (e.g., <object#1, person#1>). Most generalizations consist of products as concrete entities, there are however examples of abstract entities for the products as well (e.g., <communication#2, person#1>). CAUSATION is one of the relations where almost any type is allowed for its arguments and it can be seen in a variety of generalizations that were detected. The definitions of the ORIGIN-ENTITY and the THEME-TOOL relations do not list the types of arguments that are allowed for this relation but rather their counterexamples (i.e. plans and strategies are not allowed as tools).

Differences among semantic relations It is tempting to conclude that ORIGIN - ENTITY and THEME - TOOL are relations where semantic constraints do not play such a significant role. Indeed, even on the training set, these semantic relations could not be validated with precision that would be comparable to the other relations. Since our method, which exploits only WordNet information, does not seem to be very useful, a question becomes whether context would help more. The best performing system in category “C4” (no WordNet, but queries are used) designed by Nakov (2007) yields much lower scores for these two relations and for PART - WHOLE, which is in line with our results. One explanation for why THEME-TOOL seems to be so different from other relations would be in the way it is defined. According to its definition, two arguments are related to each other via "some kind of action" (Section 3.1). In our view, this relation may be treated as ternary rather than binary.

If used alone, constraints generated by both methods may provide accuracy similar or higher to the existing solutions based on syntactic information such as the shortest dependency kernel. Given performance on the test data set, our method provides rather coarse generalizations which results in a relatively high recall.

5 Conclusions and Future Work

We have presented a method to derive constraints for semantic relations and studied it on seven relations. As can be seen from the experimental results, some relations can be described by such constraints in a relatively precise manner (e.g., INSTRUMENT-AGENCY, PRODUCT-PRODUCER, CONTENT-CONTAINER), while others cannot (e.g., ORIGIN-ENTITY, THEME-TOOL).

There are several directions which might be explored on this topic in the future. One of them includes finding constraints based on probabilistic clustering. In particular, we may use distributional clustering of arguments given information from large data sets. This would enable finding constraints in the cases when data is not annotated with the WordNet synsets. Yet another direction would be incorporating relational similarity.

Acknowledgments

The authors thank the anonymous reviewers and Simon Carter for their comments. This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). This project was supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and was part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

References

Budanitsky, Alexander and Graeme Hirst (2006), Evaluating WordNet-based Measures of Lexical Semantic Relatedness, *Computational Linguistics* **32** (1), pp. 13–47, MIT Press, Cambridge, MA, USA.

Bunescu, Razvan C. and Raymond J. Mooney (2005), A shortest path dependency kernel for relation extraction, *Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, BC.

Chaffin, Roger and Douglas J. Herrmann (2001), Effects of relation similarity on part-whole decisions, *Journal of General Psychology* **115(2)**, pp. 131–139.

Fellbaum, Christiane (1998), *WordNet: An Electronic Lexical Database*, MIT Press.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret (2007), SemEval-2007 Task 04: Classification of semantic relations between nominals, *ACL 2007*.

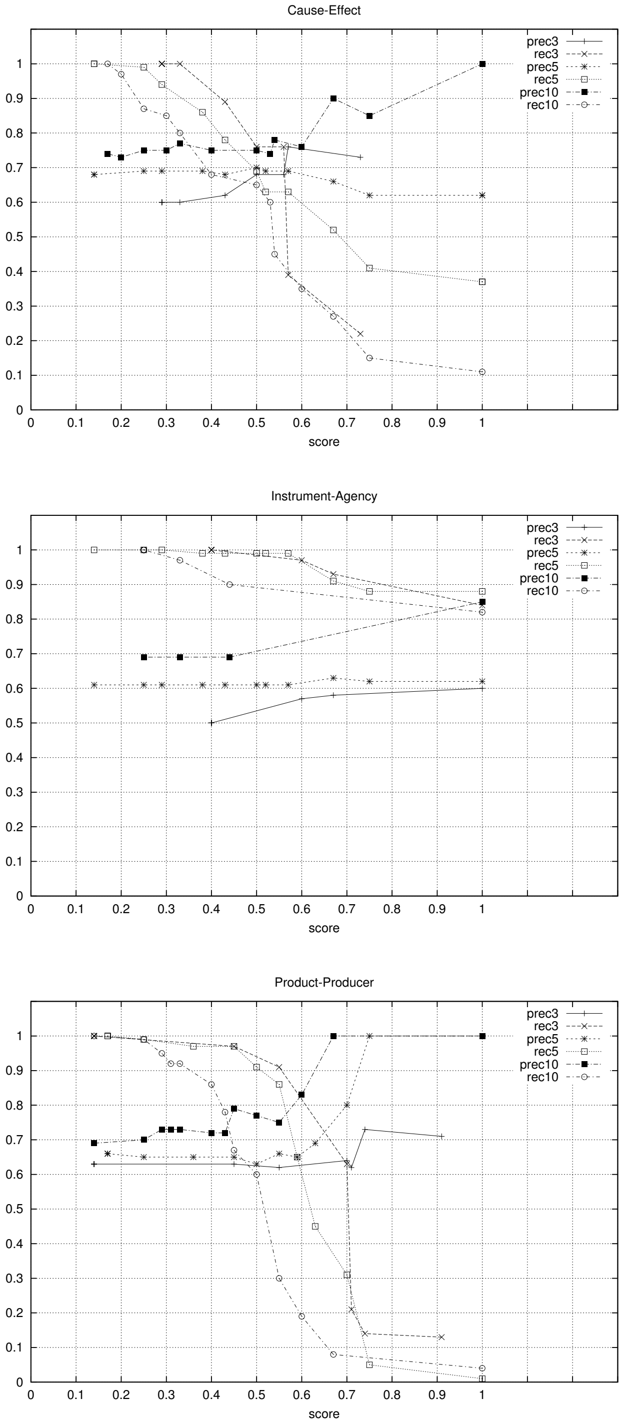

Figure 2: Clustering solutions on CAUSE - EFFECT, INSTRUMENT - AGENCY, and PRODUCT - PRODUCER.

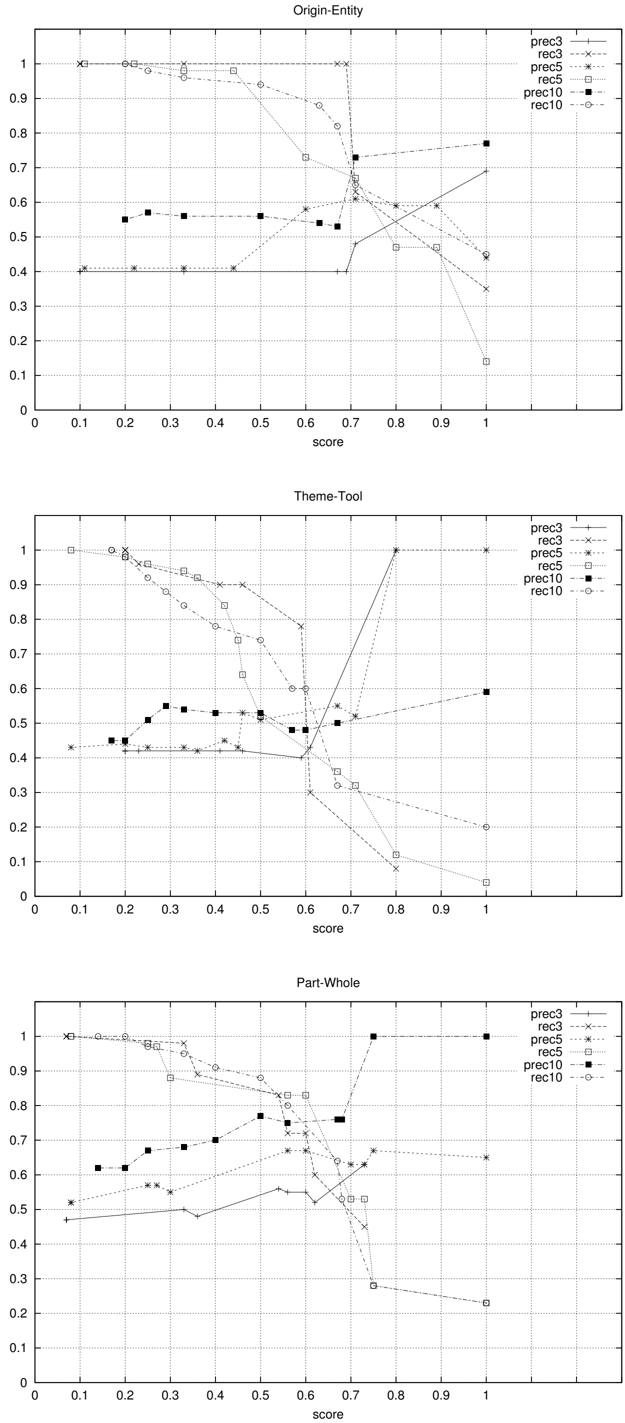

Figure 3: Clustering solutions on ORIGIN - ENTITY, THEME - TOOL, and PART - WHOLE.

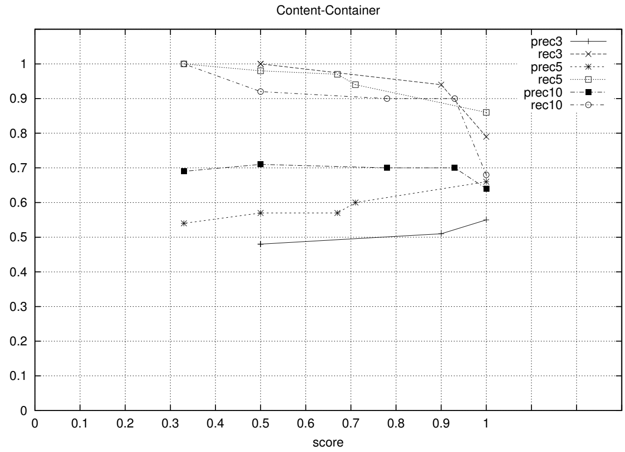

Figure 4: Clustering solutions on CONTENT - CONTAINER.

Hearst, Marti (1992), Automatic acquisition of hyponyms from large text data, *Proceedings of COLING-92*, pp. 539–545.

Katrenko, Sophia, Pieter Adriaans, and Maarten van Someren (2010), Using local alignments for relation recognition, *International Journal of Artificial Intelligence Research (JAIR)* **38**, pp. 1–48.

Khoo, Christopher S. G. and Jin-Cheon Na (2006), Semantic relations in information science, *Annual Review of Information Science and Technology* **40**, pp. 157–228.

McDonald, Ryan (2005), Extracting relations from unstructured text, *Technical Report MS-CIS-05-06*, UPenn.

Moldovan, Dan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju (2004), Models for the semantic classification of noun phrases, *HLT-NAACL 2004: Workshop on Computational Lexical Semantics*, pp. 60–67.

Nakov, Preslav (2007), UCB: System description for SemEval task #4, *SemEval-2007*.

Nakov, Preslav (2008), Paraphrasing verbs for noun compound interpretation, *Proceedings of the Workshop on Multiword Expressions (MWE'08), in conjunction with the Language Resources and Evaluation conference, Marrakech, Morocco, 2008*.

Palmer, Martha and Zhibiao Wu (1995), Verb semantics for English-Chinese translation, *Technical report*, Technical Report No. MS-CIS-95-39, Department of Computer & Information Science, University of Pennsylvania.

van der Plas, Lonneke (2008), *Automatic Lexico-Semantic Acquisition for Question Answering*, PhD thesis, University of Groningen.

Zhao, Ying and George Karypis (2002), Evaluation of hierarchical clustering algorithms for document datasets, *CIKM 2002*.