UTD-HLT-CG: Semantic Architecture for Metonymy Resolution and Classification of Nominal Relations

Cristina Nicolae, Gabriel Nicolae and Sanda Harabagiu
Human Language Technology Research Institute
The University of Texas at Dallas
Richardson, Texas
{cristina, gabriel, sanda}@hlt.utdallas.edu

Abstract

In this paper we present a semantic architecture that was employed for processing two different SemEval 2007 tasks: Task 4 (Classification of Semantic Relations between Nominals) and Task 8 (Metonymy Resolution). The architecture uses multiple forms of syntactic, lexical, and semantic information to inform a classification-based approach that generates a different model for each machine learning algorithm that implements the classification. We used decision trees, decision rules, logistic regression and lazy classifiers. A voting module selects the best performing module for each task evaluated in SemEval 2007. The paper details the results obtained when using the semantic architecture.

1 Introduction

Automatic semantic interpretations of natural language text rely on (1) semantic theories that capture the subtleties employed by human communications; (2) lexico-semantic resources that encode various forms of semantic knowledge; and (3) computational methods that model the selection of the optimal interpretation derived from the textual data. Two of the SemEval 2007 tasks, namely Task 4 (Classification of Semantic Relations between Nominals)and Task 8 (Metonymy Resolution) employed distinct theories for the interpretation of their corresponding semantic phenomena, but, nevertheless, they also shared several lexico-semantic resources, and, furthermore, both these tasks could have been cast as classification problems, in vein with most of the recent work in computational semantic processing. Based on this observation, we have designed and implemented a semantic architecture that was used in both tasks. In Section 2 of this paper we give a brief description of the semantic theories corresponding to each of the two tasks, while in Section 3 we detail the semantic architecture. Section 4 describes the experimental results and evaluation.

We have used three lexico-semantic resources: (i) the WordNet lexico-semantic database; (ii) VerbNet; and (iii) the Lexical Conceptual Structure (LCS) database. Used only by Task 4, WordNet is a lexico-semantic database created at Princeton University[1] (Fellbaum, 1998), which encodes a vast majority of the English nouns, verbs, adjectives and adverbs, and groups synonym words into synsets. VerbNet[2] is a broad-coverage, comprehensive verb lexicon created at University of Pennsylvania, compatible with WordNet, but with explicitly stated syntactic and semantic information, using Levin verb classes (Levin, 1993) to systematically construct lexical entities. Classes are hierarchically organized and each class in the hierarchy has its corresponding syntactic frames, semantic predicates and a list of typical verb arguments. The Lexical Conceptual Structure (Traum and Habash, 2000) is a compositional abstraction with language-independent properties. An LCS is a directed graph with a root. Each node is associated with certain information, including a type, a primitive and a field. An LCS captures the semantics

[1] http://wordnet.princeton.edu

[2] http://verbs.colorado.edu/verb-index/verbnet-2.1.tar.gz

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 454–459,
Prague, June 2007. ©2007 Association for Computational Linguistics

Relation	*Positive example*
1. Cause-Effect	Earplugs relieve the *discomfort* from *traveling* with a cold allergy or sinus condition.
2. Instrument-Agency	The *judge* hesitates, *gavel* poised, shooting them a warning look.
3. Product-Producer	The *boy* who made the *threat* was arrested, charged, and had items confiscated from his home.
4. Origin-Entity	*Cinnamon oil* is distilled from *bark chips* and used to alleviate stomach upsets.
5. Theme-Tool	The *port scanner* is a utility to scan a system to get the status of the TCP.
6. Part-Whole	The *granite benches* are former windowsills from the Hearst Memorial Mining Building.
7. Content-Container	The *kitchen* holds patient *drinks* and snacks.

Table 1: Examples of semantic relations.

of a lexical item through a combination of semantic structure and semantic content.

2 Semantic Tasks

The two semantic tasks addressed in this paper are: **Classification of Semantic Relations between Nominals (Task 4)**, defined in (Girju et al., 2007) and **Metonymy Resolution (Task 8)**, defined in (Markert and Nissim, 2007). Please refer to these task description papers for more details. Both are cast as classification tasks: given an unlabeled instance, a system must label it according to one class of a set specific to each task.

The training and testing datasets for the metonymy resolution task are annotated in an XML format. There are 1090 training and 842 testing instances for companies, and 941 training and 908 testing instances for locations. Each training instance corresponds to a context in which a single name is annotated with its reading (*metonymic/literal/mixed*) and, in case of metonymy, its type (*metotype*). The testing dataset for this task is annotated in a similar manner, only the reading of the name is left unknown and must be decided by the system.

For the classification of semantic relations between nominals, there exist seven training sets of 140 instances each for the seven semantic relations, and seven corresponding testing sets of around 70 instances each. A training instance is annotated with information about the boundaries of the two nominals whose relation must be determined, the truth value of their relation, the WordNet sense of each nominal, and the query that was employed by the annotators to retrieve this example from the Web. The testing instances are similar, with the only difference being that the truth value of the relations is unknown and must be determined.

3 Semantic Architecture

The semantic architecture that we have designed is illustrated in Figure 1, which contains the basic modules and resources used in the various phases of processing the input data towards the final submission format. The grayed-out modules are all used only for the semantic relations classification task, while the part of the figure represented by dotted lines appears only in the metonymy resolution algorithm. The input to the system, for both tasks, comprises the annotated instances, either from the training or the testing dataset. Before any feature is extracted, the data passes through a pipeline of preprocessing modules. The text is first split into tokens in a heuristic manner. The resulting tokenized text is given as input to Brill's part of speech tagger[3], which associates each word with its part of speech (e.g., *NN*, *PRP*). The data further goes through Collins' syntactic parser[4], which builds the syntactic trees for all the sentences in the text.

Additionally, for semantic relations classification, the system creates the dependency structures for all the sentences, using the dependency parser built at Stanford[5] and described in (de Marneffe et al., 2006). The dependency parser extracts some of 48 grammatical relations for each pair of words in a sentence. A second module that is specific only to this task is (Surdeanu and Turmo, 2005)'s semantic role labeler, which extracts the shallow semantic structure for each sentence, that is, the predicates and their arguments.

In order to extract the features for the machine learning algorithm, the modules described above are used, and, in addition, information from WordNet, VerbNet and the LCS Database is incorporated,

[3] http://www.cs.jhu.edu/~brill/

[4] http://people.csail.mit.edu/mcollins/code.html

[5] http://nlp.stanford.edu/downloads/lex-parser.shtml

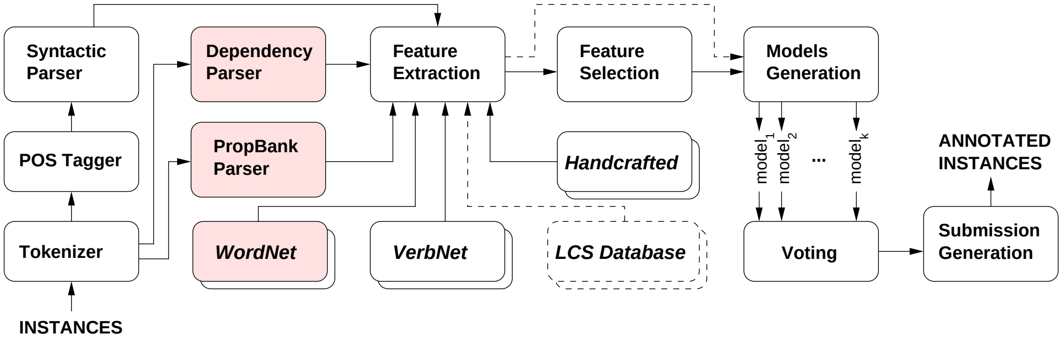

Figure 1: Semantic architecture.

Category	*Feature name*	*Feature description*
syntactic	*prevpos*	part of speech of previous word in the sentence
	nextpos	part of speech of next word in the sentence
	determiner	if the word has a determiner
	prepgoverning	if the word is governed by a prepositional phrase (PP), we extract the preposition
	insidequotes	if the word is inside quotes
	lemmapost	if the word is postmodifier for a noun, take the lemma of the noun
	lemmapre	if the word is premodifier for a noun, take the lemma of the noun
	possession	if the word is a possessor, and what it possesses
semantic	*role*	the role(s) of the name in the sentence: subject, object, under PP
	rolelemma	the combination between the role and the lemma of the verb whose argument the word is
	rolevn	same as above, but using the VerbNet class instead of the verb's lemma
	rolelevin	same as above, but using the Levin class instead of the verb's lemma
	rolelcs	same as above, but using LCS primitives from the LCS database instead of the verb's lemma

Table 2: Features for metonymy resolution.

along with other features, based on the manual annotations for both the training and testing datasets by the task organizers. These other features use the grammatical annotations for the possibly metonymic name, in the case of metonymy resolution, and the query that was used to retrieve that particular instance and the disambiguated WordNet sense for the two nominals, in the case of semantic relations classification.

The features implemented for the two tasks are described in Tables 2 and 3. Their types are: syntactic, semantic, lexical and other. The *syntactic features* express the relationships between the target words and words from the rest of the sentence (e.g., the part of speech of the previous word in the sentence, or the dependency relations between two words). The *semantic features* make use of the information given by the resources used by the system (e.g., the VerbNet class of the verb whose argument the word is, or the lexicographic category of a word in WordNet). The *lexical feature* is the lemma of the word. The *other feature* is the query provided by Task 4.

Using these sets of features, a number of models were generated by different machine learning techniques included with the Weka data mining software (Witten and Frank, 2005). The machine learning classifiers comprise decision trees, decision rules, logistic regression, and ”lazy” classifiers like k-nearest-neighbor. Because of too many features generated for a relatively small training dataset, feature selection is performed by Weka before creating the models. Metonymy resolution uses in addition the entire set of features, since the dataset has seven times more instances than the other task. For the classification of semantic relations, the initial total and the number of features that remain after the selection are printed in Table 4.

For metonymy resolution, there are six subtasks to be resolved, which result from all combinations between *organization/location* and *coarse/medium/fine* granularity of the label. For the classification of nominal relations, there are 28 subtasks, resulting from the processing of the seven se-

Category	***Feature name***	***Feature description***
syntactic	*dependency*	the dependency relations between the two words
	modifier	if one word is a modifier of the other
	prepositions	the prepositions immediately before and after both words
	determiners	the determiners of the two words
	pattern	the simplified pattern that exists in the sentence between the two words
lexical	*lemmas*	the lemmas of the words
semantic	*predicates*	the predicates whose arguments the two words are
	predtypes	the predicate types of the predicates above
	samepred	if the two words are arguments of the same predicate, which one that is
	lexname	the lexicographic category of each word in WordNet
	hyponym	if one word is a hyponym of the other in WordNet
	partof	if one word is a part of the other in WordNet
	shareholonym	if the two words share a holonym in WordNet
	shareparent	if the two words share a parent in WordNet
other	*query*	the query that was used by the annotators to retrieve the training example from the Web

Table 3: Features for classification of semantic relations between nominals.

	R1	***R2***	***R3***	***R4***	***R5***	***R6***	***R7***
before	682	1200	913	898	861	849	677
after	13	19	10	15	15	8	16

Table 4: The number of features before and after Weka selection, for each semantic relation dataset: R1 Cause-Effect, R2 Instrument-Agency, R3 Product-Producer, R4 Origin-Entity, R5 Theme-Tool, R6 Part-Whole, and R7 Content-Container.

Base type	***Coarse***	***Medium***	***Fine***	***BA***
Locations	84.1	84.0	82.2	79.4
Organizations	73.9	71.1	71.1	61.8

Table 5: Accuracy for the metonymy resolution system at three granularity levels.

Base type	***Reading***	***P***	***R***	***F***	***BA***
Locations	literal	88.2	92.4	90.2	79.4
	non-literal	64.1	52.4	57.6	20.6
Organizations	literal	75.8	84.8	80.0	61.8
	non-literal	69.6	56.2	62.2	38.2

Table 6: Performance for the metonymy resolution system for the coarse level.

mantic relations, in which four experiments are conducted, each with an increasing number of training instances. We treated each subtask as a separate classification problem. Its training set and features are fed into Weka to create several models. Each classification algorithm mentioned before is employed to obtain one model. For each subtask, the voting module selects the best performing model on 10-fold crossvalidation, which is used to classify the test instances. These annotated instances make up the submission dataset for that particular subtask. To note is that the coarse metonymic level and the semantic relations classification are binary classifications, while the rest of the metonymic subtasks are multi-class classifications, performed in a single stage.

4 Experimental Results and Evaluation

Both the metonymy resolution system and the system for classification of semantic relations performed well in the SemEval 2007 competition. The experiments presented in this paper were done on the training and testing datasets for each subtask. To note is that no other training data was collected or used than the one provided by the organizers.

4.1 Results for Metonymy Resolution

This system was scored by measuring its accuracy at three granularity levels (*coarse, medium,* and *fine*) and the precision, recall and F score for all combinations of *locations/organizations* and *literal/non-literal*. These results are tabulated in Tables 5, 6, 7 and 8.

All results are compared with the baseline accuracy values (BA). In Table 5, the baselines are computed by taking all readings to be literal; for the rest, the baseline is the percentage in the gold test data of each reading. As can be observed, the readings

Base type	***Reading***	***P***	***R***	***F***	***BA***
Locations	literal	87.8	93.5	90.5	79.4
	mixed	0.0	0.0	0.0	2.2
	metonymic	63.6	52.3	58.0	18.4
Organizations	literal	74.3	90.0	81.4	61.8
	mixed	28.6	13.1	18.0	7.2
	metonymic	66.8	47.1	55.3	31.0

Table 7: Performance for the metonymy resolution system for the medium level.

Base type	***Reading***	***P***	***R***	***F***	***BA***
Loc	literal	85.7	94.6	89.9	79.4
	mixed	0.0	0.0	0.0	2.2
	othermet	0.0	0.0	0.0	1.2
	obj-for-name	0.0	0.0	0.0	0.0
	obj-for-repr	0.0	0.0	0.0	0.0
	place-for-people	57.1	45.4	50.6	15.5
	place-for-event	0.0	0.0	0.0	1.1
	place-for-prod	0.0	0.0	0.0	0.1
Org	literal	74.4	90.4	81.6	61.8
	mixed	50.0	3.33	6.25	7.1
	othermet	0.0	0.0	0.0	1.0
	obj-for-name	80.0	66.7	72.7	0.7
	obj-for-repr	0.0	0.0	0.0	0.0
	org-for-members	61.3	64.0	62.6	19.1
	org-for-event	0.0	0.0	0.0	0.1
	org-for-prod	60.6	29.9	40.0	8.0
	org-for-fac	0.0	0.0	0.0	1.9
	org-for-index	0.0	0.0	0.0	0.4

Table 8: Performance for the metonymy resolution system for the fine level.

for locations were more reliably identified than the ones for companies. An explanation for this difference in performance lies in the fact that locations, in their literal readings, are inactive entities, whereas in their non-literal readings they are very often active, especially in the annotated instances of the training dataset. This cannot be said for organizations– they can be active in their literal readings. The active vs. inactive criterion, therefore, functions better for locations. Furthermore, since the training set contains a ratio *literals/non-literals* of 1.7 for organizations and 3.9 for locations, the models were skewed, identifying literal readings more easily than non-literal ones, as shown in Table 6.

4.2 Results for Classification of Semantic Relations between Nominals

This task's performance was measured by accuracy, precision, recall and F-measure, the latter constituting the score for ranking the systems in the competition. Table 9 presents these scores by semantic relation. The column entitled "Inst" contains the number of instances in the testing sets corresponding to each relation. The average baseline values were computed by guessing the label to be the majority in the dataset for each relation. From this table it can be observed that the PRODUCT-PRODUCER, INSTRUMENT-AGENCY, and CAUSE-EFFECT relations were detected with a relatively very high performance score, whereas the THEME-TOOL relation classification yielded a relatively small score. This can be explained as the effect of their specifications; the three best-ranked relations are well-defined by human standards, while the THEME-TOOL relation is more ambiguous.

Semantic relation	***P***	***R***	***F***	***Acc***	***Inst***
Cause-Effect	65.5	87.8	75.0	70.0	80
Instrument-Agency	68.3	73.7	70.9	70.5	78
Product-Producer	66.7	96.8	78.9	65.6	93
Origin-Entity	62.9	61.1	62.0	66.7	81
Theme-Tool	70.0	24.1	35.9	64.8	71
Part-Whole	55.6	76.9	64.5	69.4	72
Content-Container	82.4	36.8	50.9	63.5	74
Average	67.3	65.3	**62.6**	67.2	78.4
Avg baseline	81.3	42.9	56.2	57.0	78.4

Table 9: Performance of the semantic relations classification system for each semantic relation.

Table 10 contains the scores of the 10-fold crossvalidation experiments that were performed on the training dataset in order to select the best classification algorithm. The classifiers used in these experiments were, in the order of appearance in the table: JRip, Random Forest, ADTree, Logistic Regression, IBk, and Random Tree. The Logistic Regression classifier was chosen in the vast majority of cases, because it achieved the highest score for six out of the seven relations. For R6, PART-WHOLE, Random Forest was preferred. This ranking between the scores of classifying relations, done considering training accuracy only, does not however anticipate the final F score ranking in Table 9. In particular, the crossvalidation accuracy of R5, THEME-TOOL, is better than the accuracy for R3, PRODUCT-PRODUCER, which came first in the final results, whereas R5 came last and at a large distance from the others. These lower-than-expected results in the

Alg	***R1***	***R2***	***R3***	***R4***	***R5***	***R6***	***R7***
JRip	72.1	76.4	68.6	66.4	68.6	66.4	73.6
RandF	78.6	85.0	72.1	77.1	74.3	**70.7**	73.6
ADTree	72.9	79.3	70.0	70.7	70.7	68.6	69.3
LogReg	**79.3**	**85.7**	**72.1**	**80.0**	**76.4**	70.0	**75.7**
IBk	78.6	83.6	70.7	75.7	74.3	70.0	72.1
RandT	79.3	85.7	71.4	77.1	75.0	70.0	72.1

Table 10: Results on 10-fold crossvalidation for each relation and each classifier.

evaluation were caused in part by the drastic feature selection module that was applied before generating the models. In experiments performed on the development data, the accuracy on 10-fold crossvalidation was increased with an average of 7% by feature selection, but the same feature set on the testing data obtained a final score 4.7% less than the one obtained by using all the features (F=67.3%). The results submitted in the evaluation were based on feature selection because of this misleading performance shift observed on the development set.

The task of classification of semantic relations between nominals required data to be separated into four training sets: the first 35 instances (D1), the first 70 instances (D2), the first 105 instances (D3), and the entire set, 140 instances (D4). The letter "D" stands for systems that use both the WordNet and the query information provided by the organizers. The results on the four sets are illustrated in Figure 2. The results generally increase with the size of training data, and tend to be the same on D3 and D4, which means that the D4 set does not bring significant new information compared to D3.

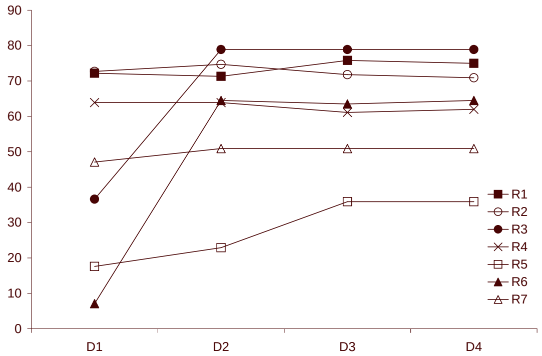

Figure 2: Results of training on different portions of the training dataset.

5 Conclusions

This paper has presented a semantic architecture that participated in the SemEval 2007 competition to evaluate two tasks, one for metonymy resolution, and the other for the classification of semantic relations between nominals. Although the tasks were very different, the architecture produced competitive results. The experimental results are reported in this paper in a detailed manner, and some interesting observations can be drawn from them.

References

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. In *Computer Speech and Language*, volume 19, pages 479–496.

Roxana Girju, Marti Hearst, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Task 04: Classification of semantic relations between nominal at semeval 2007. In *SemEval 2007*.

Beth Levin. 1993. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago and London.

Katja Markert and Malvina Nissim. 2002. Metonymy resolution as a classification task. In *the 2002 Conference on Empirical Methods in Natural LAnguage Processing (EMNLP2002)*.

Katja Markert and Malvina Nissim. 2007. Task 08: Metonymy resolution at semeval 2007. In *SemEval 2007*.

Mihai Surdeanu and Jordi Turmo. 2005. Semantic role labeling using complete syntactic analysis. In *CoNLL 2005, Shared Task*.

David Traum and Nizar Habash. 2000. Generation from lexical conceptual structure. In *Workshop on Applied Interlinguas, ANLP-2000*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition.