LCC-SRN: LCC's SRN System for SemEval 2007 Task 4

Adriana Badulescu
Language Computer Corporation
1701 N Collins Blvd #2000
Richardson, TX, 75080
adriana@languagecomputer.com

Munirathnam Srikanth
Language Computer Corporation
1701 N Collins Blvd #2000
Richardson, TX, 75080
srikanth@languagecomputer.com

Abstract

This document provides a description of the Language Computer Corporation (LCC) SRN System that participated in the SemEval 2007 Semantic Relation between Nominals task. The system combines the outputs of different binary and multi-class classifiers build using machine learning algorithms like Decision Trees, Semantic Scattering, Iterative Semantic Specialization, and Support Vector Machines.

1 Introduction

The Semantic Relations between Nominals task from SemEval 2007 focuses on identifying the semantic relations that hold between two arguments manually annotated with word senses (Girju et al, 2007).

The previous work in identifying semantic relations between nominals focuses on finding one or more relations in text for specific syntactic patterns or constructions (like genitives and noun compounds) using semi-automated and automated systems. An overview of some of these methods can be found in (Badulescu, 2004).

The LCC SRN system, developed during the SRN training period, was for us, the beginning of a different approach to semantic relations detection: detecting semantic relations in text without using a syntactic pattern. Our existing work on semantic relation detection was on detecting semantic relations in text (one or more at a time) at different levels in the sentence using different syntactic patterns like genitives, noun compounds, verb-arguments, etc.

For SRN, we built a new system that combines the output of the pattern dependent classifiers with the new pattern-independent classifiers for better results.

The remainder of this paper is organized as follows: Section 2 describes our system, Section 3 details the experimental results, and Section 4 summarizes the conclusions.

2 System description

The system consists of two types of classifiers: classifiers that do not use the syntactic parsed tree and that were built specifically for the SemEval 2007 Task 4(SRN) and classifiers that use specific syntactic pattern to determine the semantic relations and there were previously developed at LCC and then adapted to the SRN task (SRNPAT).

The classifiers for each type were built from annotated examples using supervised **machine learning algorithms** like Decision Trees (***DT***)[1], Support Vector Machines (***SVM***)[2], Semantic Scattering (***SS***) (Moldovan and Badulescu, 2005) , Iterative Semantic Specialization (***ISS***) (Girju, Badulescu, and Moldovan, 2006), Naïve Bayes (***NB***)[3] and Maximum Entropy (***ME***)[4].

The outputs of different classifiers (built using different types of machine learning algorithms were **combined** and ranked using predefined rules.

Figure 1 shows the **architecture** of our SRN system.

[1] C5.0., http://www.rulequest.com/see5-info.html
[2] LIBSVM, www.csie.ntu.edu.tw/~cjlin/libsvm/
[3] jBNC, http://jbnc.sourceforge.net
[4] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 215–218,
Prague, June 2007. ©2007 Association for Computational Linguistics

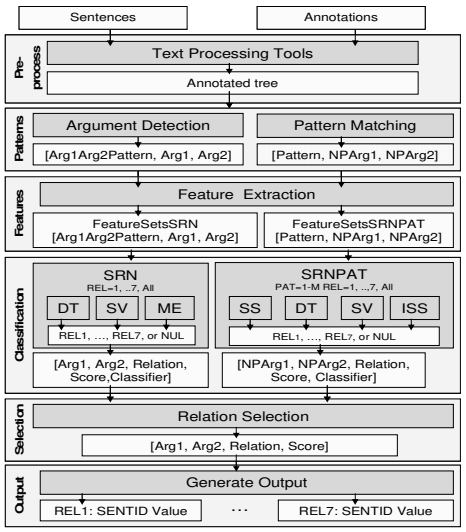

Figure 1. The architecture of our SRN system.

2.1 Text Preprocessing

The sentences were processed using an in-house text tokenizer, Brill's part-of-speech tagger, an in-house WordNet–based concept detector, an in-house Named Entity Recognizer, and an in-house syntactic parser.

Then, the syntactic and semantic information obtained using these tools (concepts, part of speech, named entities, etc) or obtained from the sensekeys for the arguments as provided by the Task 4 organizers (e.g. word senses, lemmas, etc) were mapped into the syntactic trees. If an argument corresponds to more than one tree node, the annotation was mapped to the phrase containing the two nodes.

2.2 Learning and Classification Methods

The core of our system is the learning and classification module.

We used two types of methods: pattern-dependent that uses the syntactic parsed trees for extracting and assigning a label to the arguments and pattern-independent that creates classifiers form all the examples disregarding the pattern in the tree.

2.2.1 Pattern-independent Methods (SRN)

Considering the limited number of examples for each pattern, we developed pattern-independent methods for classifying the semantic relations using the provided argument annotations and the context from the sentence.

We built two types of **classifiers**: **binary** that focuses on building a classifier for a specific relation (SRNREL) and **multi-class** methods that build classifiers for all the SRN relations (SRN). Table 1 presents the accuracy of the classifiers built using different machine learning algorithms.

Relation	DT	SVM	ME
1	52.10	46.15	46.67
2	41.40	30.76	60.00
3	61.70	51.61	63.33
4	59.30	52.17	53.33
5	58.60	39.99	50.00
6	71.70	24.99	73.33
7	50.00	57.13	43.33
Avg	56.40	43.26	55.71

Table 1. The accuracy of the SRNREL classifiers built using different machine learning algorithms.

The classifiers were built using lexical, semantic, and syntactic features of the arguments, their phrases, their clauses, their common phrase/clause, and their modifier or head phrase. The system uses WordNet, an in-house Named Entity Recognizer, and an in-house Syntactic Parser for determining the values of some of these features. Table 2 presents the list of features used by the SRN classifiers.

Argument's lexical, semantic, and syntactic features: the surface form, the label (POS tag or phrase label), the named entity (human, group, location, etc), the WordNet hierarchy (entity, group, abstraction, etc), the Semantic Scattering class (e.g. object, substance, etc), the grammatical role (subject or object of the clause), the syntactic parser structure, the POS Pattern (the sequence of POS of the words from the argument), and the phrase pattern (the sequence of labels of the phrases, words from the argument);
Argument's phrase features: surface form, label, grammatical role, named entity, POS pattern, Phrase patterns;
Argument's Modifier/Head features: the label, surface forms, NE, and WN Hierarchy for the first modifier, post modifier, pre-modifier, and head;
Arguments' common tree node features: label, named entity, grammatical role, POS pattern, and phrase pattern, the tree path between arguments, and their order in tree;
Arguments' clause: label, verb, voice, POS pattern, phrase pattern.

Table 2. The list of features used for the SRN classifiers.

2.2.2 Pattern-dependent Methods (SRNPAT)

The second type of methods we used, were for particular patterns frequent in the training corpus. Table 3 shows the list of most frequent **patterns** in the training corpus. For having general pattern and covering the arguments that correspond to more

than one node in a tree, we considered as argument the noun phrase that contains the nominal instead of the node for the nominal.

Pattern name	Example
Noun compounds: NN_1 NN_2	If you are cleaning a *<e1>coffee</e1> <e2>maker</e2>* that hasn't been cleaned regularl271076ad y, repeat this step again with a fresh vinegar and water mixture.
Of-genitives: NP_1 of NP_2	The incoming *<e1>chairman</e1>* of *the <e2>committee</e2>* is promising an array of oversight investigations that could provoke sharp disagreement with Republicans and the White House.
S-genitives: NP_1 's NP_2	This is the *<e1>government</e1>*'s *<e2>effort</e2>* to encourage more employers to open up childcare centres at the respective ministries and government departments.
Prepositional constructions: NP_1 IN NP_2	I believe that unless we take this issue seriously, the red squirrel is facing eventual *<e1>extinction</e1>* from *the <e2>woods</e2> of Scotland*.
Verbal constructions: NP_1 VB NP_2	On both of my systems, *the <e1>reboot</e1>* produced *the ominous <e2>message</e2> 'Missing operating system'*.
Verbal prepositional constructions: NP_1 VB IN NP_2	Manila radio station DZMM quoted survivors as saying that *the <e1>fire</e1>* started with *an <e2>explosion</e2>* in the cargo hold and spread across the ship within minutes.

Table 3. The most frequent patterns found in the training corpus.

For the pattern-dependent methods we adapted some of our existing binary and multi-class **classifiers** to work with the SRN relations.

For the SRN system we used only one **binary** classifier built for the Part-Whole relation (relation 6) using the ISS learning algorithm and trained/tested on the examples used in (Girju, Badulescu, and Moldovan, 2006) and different **multi-class** classifiers for the first 4 patterns from Table 3 built using DT, SVM, SS, and NB learning algorithms trained on a corpus annotated with 40 semantic relations (extracted from Wall Street Journal articles from the TreeBank collection and LATimes articles from TREC 9 collection) that includes the 7 SRN relations (or equivalents). (Badulescu, 2004) gives more details on this list of relations (definitions, examples, distribution on corpus, etc). Table 4 shows the **accuracy** of these classifiers on other WSJ and LAT articles for the 40 LCC relations and respectively Part-Whole relation for the most frequent patterns from the SRN corpus (Table 3).

Pattern cluster	SS	DT	NB	SVM	ISS
Noun compounds	52.54	47.8	53.45	74.79	73.59
S-genitives	62.27	56.2	58.27	72.66	87.26
Of-genitives	67.55	53.1	54.63	72	87.26
Prepositional constructions	43.48	43.3	41.92	64.52	75.97

Table 4. The accuracy of the SRNPAT classifiers for the list of 40 LCC relations and the Part-Whole Relation.

2.3 Relation Selection

Any of the SRN or SRNPAT classifiers can return a relation for a pair of arguments. The best relation is selected by weighting them using the following predefined rules:

- The relations returned by the SRN classifiers weight more than the ones returned by SRNPAT classifiers because they were trained on the task annotated examples
- The relations returned by the binary classifiers weight more than the ones returned by multi-class classifiers because they focus on one relation and therefore are more precise.

3 Experimental Results

3.1 Experiments on Testing

During the competition we performed several experiments to assess the correct combination of classifiers that leads to the best results.

The organizer provided 140 examples for each of the 7 relations. For testing the classifiers we trained the system on the first 110 examples and tested it on the last 30 of them.

We performed different sets of experiments.

- **Experiments with one type of classifiers**. These experiments showed that ME has a best performance (55.1) 10.05 more than DT and 8.05 more than SV. ME also got the highest score for Cause-Effect, while DT obtained the best score for Product-Producer.
- **Experiments with multiple classifiers**. These experiments showed that DT+SV+SS+ISS has the best score (66.72) followed by DT+SS+ISS with 55.66. Also by adding the SS and ISS classifiers the DT score increased with 10.51, the SV score with 5.81 and the DT+SV with 20.57.

- **Experiments with types of methods**. These experiments showed that the SRN methods (with a score 0.44) are better than the SRNPAT methods (with a score of 0.41) with 0.03 which was expected since SRN were trained on provided examples.

Table 5 shows the results of our SRN system when using specific classifiers or a combination of classifiers. The time did not permit us to do any experiments with the ME and NB classifiers.

Classifier Combination	Average F-measure
DT	45.05
SV	47.05
ME	55.10
DT+SV	46.15
DT+SS+ISS	*55.66*
SV+SS+ISS	52.96
DT+SV+SS+ISS	66.72
SRN	44.31
SRNPAT	41.15

Table 5. The results of some of our experiments with the different classifiers on the testing corpus.

We submitted the DT+SS+ISS version because of its closeness to the normal distribution rather than DT+SV+SS+ISS that had a better f-measure but it was closer to All-True. The evaluation results showed that the testing examples we used were representative and the DT+SV+SS+ISS produce better results.

3.2 Results

Table 6 shows the results obtained by our system on the evaluation corpus for the B4 case (using WordNet but not the query and all the training examples.

Relation	Precision	Recall	F-measure	Accuracy
1	50.8	73.2	60.0	50.0
2	54.5	31.6	40.0	53.8
3	66.7	100.0	80.0	66.7
4	80.0	22.2	34.8	63.0
5	42.2	65.5	51.4	42.3
6	39.6	80.8	53.2	48.6
7	57.1	31.6	40.7	51.4
Avg	55.9	57.8	51.4	53.7

Table 6. The results of our system on the evaluation corpus.

Table 7 shows a comparison of our results with the following baseline systems: *All-True*, a system that always returns true, *Majority*, a system that always returns the majority value from the training, and *Prob-Match*, a system that randomly generate the value. We have obtained a larger precision and accuracy than the All-True and the Prob-Match systems. However, we obtained a lower recall and therefore an F-measure.

System	Precision	Recall	F-measure	Accuracy
All-True	48.5	100.0	64.8	48.5
Majority	81.3	42.9	30.8	57.0
Prob-Match	48.5	48.5	48.5	51.7
LCC-SRN	55.9	57.8	51.4	53.7

Table 7. Comparison with the baselines.

3.3 Discussions

The results are promising. However, there is still room for improvement. The system was developed in a limited time, and therefore it could have been benefited from more features, feature selection, more experiments, a more complex relation selection scheme (using learning), more patterns, and more types of machine learning algorithms (especially unsupervised ones).

4 Conclusion

We presented a system for classifying the semantic relations between nominals that combines the results of different methods (pattern-dependent or pattern-independent) and machine learning algorithms (decision tree, support vector machines, semantic scattering, maximum entropy, naïve bayes, etc). The classifiers use lexical, semantic, and syntactic features and external resources like WordNet and an in-house Named Entity dictionary.

References

Adriana Badulescu. 2004. Classification of Semantic Relations between Nouns. PhD Dissertation. University of Texas at Dallas.

Dan Moldovan and Adriana Badulescu. 2005. A Semantic Scattering Model for the Automatic Interpretation of Genitives. In *Proceedings of HLT/EMNLP 2005*.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2006. Automatic Discovery of Part-Whole Relations. *Computation Linguistics*, 32:1.

Roxana Girju et al. 2007. Classification of Semantic Relations between Nominals: Description of Task 4 in SemEval-1, In *Proceedings of ACL-2007, SemEval-1 Workshop*.