Searching for Semantic Relations between Named Entities in I-CAB.

Fabio Celli
CLIC, University of Trento
September 23, 2009

Abstract
This paper describes the experiments for the extraction from I-CAB of three different semantic relation types (role, social, location) holding between two named entities in Italian. The method adopted exploits the information about the part-of-speech of the context before and between the two entities. This method, called Part-Of-Speech Context Count (POSCo) obtained an F1-measure of 0.812 using simply decision trees and decision lists algorithms. The rules extracted from this experiment are implemented in a semantic relation annotator (called RA) compatible with the TextPro tool suite.

1 Introduction and related work
I-CAB [6] (Italian Content Annotation Bank) is a Italian corpus made of 525 news documents taken from a local newspaper named "LAdige", annotated with temporal expressions and 4 named entity types (person, organization, location and geo-political entity), the same entity types can be detected automatically by using TextPro [7]. The task presented in this paper is to retrieve semantic relations (defined here as underspecified information holding between two entities in text) from I-CAB and to formalize the annotation procedure in a way such that it can be performed automatically. This is a classification task and it is divided into three parts: 1) semantic relation selection, 2) feature selection and 3) formalization of the annotation procedure. The three parts of the task are described in detail below.

1.1 Semantic relations. The semantic relation classes selected for this task are location[1] (x is located with respect to y), social[2] (x has a social interaction with respect to y) and role[3] (x plays a role with respect to y); they are taken from a previous experiment reported in [1], where they proved to yield to a good performance of the classifier if compared to the 7 semantic relations found in ACE 2004 [2] and from which they are derived. One example per class is reported below.

(1) Trento - la federazione prova a cambiare [Trento - the federation tries to change]
1.2 Features. The method for extracting features used in this task is inspired to the "global context" method used, among others, by Giuliano et Al. 2006 ([5]) and 2007 ([4]). This method consists in taking as features all the words appearing fore-between, in-between or between-after the two named entities. Since the procedure (see next paragraph) here assigns a semantic relation when it takes as input the entity2, in this task has been used the fore-between global context only. Since the annotation is done only on a small part of I-CAB, using all words as features would lead to sparseness of data, so some generalization from the data is needed. The generalization adopted is based on the part-of-speech (ELRA tagset): it has been developed a program, called "POSCo" (Part Of Speech Context Counter), that counts the hits of ten different part-of-speech in the sentence, resetting the count at every end-of-sentence symbol. Numeric features, retrieved with POSCo, are: 1) nouns, 2) verbs, 3) prepositions, 4) adjectives, 5) adverbs, 6) pronouns, 7) articles, 8) determiners, 9) acronyms, 10) punctuation. Nominal features are: 11) entity1, 12) entity2, 13) semantic relation. In the end there are thirteen different features, ten retrieved automatically, two previously annotated in the corpus, and the last one, the semantic relation, to be manually annotated.

1.3 Annotation Guidelines and formalization. In order to formalize the annotation procedure it was followed the algorithm reported below as pseudocode. It was also implemented in a program called RA (Relation Annotator). During the annotation phase the semantic relations that do not fall under one of the three semantic relation classes described above, such as the IS-A relation, were discarded.

```plaintext
1 variable "entity1", empty
2 variable "feature", empty
3 while input a line...
3.1 if line matches a named entity and "entity1" is empty (3.1.1 then put the named entity in "entity1")
3.2 if line matches a feature (3.2.1 then put that feature in the feature variable)
3.3 if line matches a named entity and "entity1" is not empty (3.3.1 then evaluate the relation between entity1 and named entity 3.3.2 then put the named entity in "entity1")
2.4 if line matches a end-of-sentence symbol (2.4.1 then reset the values of "entity1" and "feature")
```

2 Experiments and Discussions

The dataset contains 1000 training and 500 testing sentences randomly sampled from I-CAB and manually annotated with semantic relations by an italian native
speaker. There are 284 semantic relation instances in the training set and 96 in the test set. The J4.5 decision tree and the PART decision list algorithms (see[9]) were chosen for running the experiments since they allow the experimenter to extract if-then rules easily implementable in a program.

2.1 Experiment 1: testing the model In order to test the Part-Of-Speech-counting global context method, an experiment with the following settings was run to find a baseline: 3 features (entity1, relation, entity2), training and test sets, decision trees (Quinlan 1993 [8]) and decision lists (Frank and Witten 1998 [3]) weka algorithms. The result, taken as the baseline, is an average f-measure of 0.750, exactly the same both with the decision tree and decision lists algorithms. The confusion matrix, reported in Table 1, reveals that there are too few social relations, probably due to the language type in I-CAB, and they tend to be confused with the "role" class. Then the experiment was run again with all 13 features described above, training and test sets. Results of this experiment are an average f-measure of 0.763 with decision trees and an average f-measure of 0.782 with decision lists and show that the method adopted helps in rising the performance and the decision list algorithm works better than the decision tree. Still remains to test whether the performance could be improved or not by adding more data to the training set and more features to the model.

<table>
<thead>
<tr>
<th></th>
<th>rol</th>
<th>loc</th>
<th>soc</th>
</tr>
</thead>
<tbody>
<tr>
<td>rol</td>
<td>38</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>loc</td>
<td>36</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>soc</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Confusion matrix of experiment 1 (3 features).

2.2 Experiment 2: adding more data. A second experiment was run adding new annotated data to the training set. The experimental settings are: 13 features, training and test sets, decision trees (J4.5) and decision list (PART) weka algorithms. Results, reported in Table 2, reveal that the performance increases much constantly with decision lists but not with decision trees. Still more data are needed to test the upper bound of the learning curve, this could be material for future work.

<table>
<thead>
<tr>
<th>Training set</th>
<th>avg F (trees)</th>
<th>avg F (lists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>284</td>
<td>0.763</td>
<td>0.782</td>
</tr>
<tr>
<td>394</td>
<td>0.792</td>
<td>0.802</td>
</tr>
<tr>
<td>470</td>
<td>0.763</td>
<td>0.812</td>
</tr>
</tbody>
</table>

Table 2: Results of experiment 2 (13 features).

2.3 Experiment 3: adding more features. Experiment 3 was designed in order to test the hypothesis whether a higher number of feature in the model would lead or not to a better classification performance. POSCo has been modified in order to count the following numerical feature, then added to the
settings presented above: 14) prepositions “di” (of), 15) prepositions “a” (to), 16) prepositions “da” (from), 17) prepositions “in” (in), 18) prepositions “con” (with), 19) prepositions “su” (on), 20) prepositions “per” (for), 21) prepositions “tra/fra” (within), 22) word count. The experiment was run with the following settings: 22 features, lists and trees algorithms, training (both the one with 284 and the one with 470 instances) and test sets. Results are shown in table 3 below. This time results show a very poor performance going below the baseline (except in the case of decision trees with 470 instances in the training set), clearly indicating that the Part-Of-Speech Context Count model works better without adding more detailed features.

### 3 Conclusions and Future work

In this paper has been described the Part-Of-Speech Context Count model, a computational method for information extraction that has proved to be useful for semantic relations extraction. A program called POSCo has been developed in order to implement the model and the features extracted from I-CAB with this method were used in the experiments that tested the model itself. Results showed that the Part-Of-Speech Context Count model increased the semantic relation classification performance. The classification rules extracted in the experiments are implemented in a semantic relation annotator compatible with the TextPro tool suite.

### References


<table>
<thead>
<tr>
<th>training set</th>
<th>avg F (trees)</th>
<th>avg F (lists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>284</td>
<td>0.741</td>
<td>0.644</td>
</tr>
<tr>
<td>470</td>
<td>0.753</td>
<td>0.680</td>
</tr>
</tbody>
</table>

Table 3: Results of experiment 3 (22 features).

