

RelaxCor: A Global Relaxation Labeling Approach to Coreference Resolution

Emili Sapena, Lluís Padró and Jordi Turmo

TALP Research Center

Universitat Politècnica de Catalunya

Barcelona, Spain

{esapena, padro, turmo}@lsi.upc.edu

Abstract

This paper describes the participation of RelaxCor in the Semeval-2010 task number 1: “Coreference Resolution in Multiple Languages“. RelaxCor is a constraint-based graph partitioning approach to coreference resolution solved by relaxation labeling. The approach combines the strengths of groupwise classifiers and chain formation methods in one global method.

1 Introduction

The Semeval-2010 task is concerned with intra-document coreference resolution for six different languages: Catalan, Dutch, English, German, Italian and Spanish. The core of the task is to identify which noun phrases (NPs) in a text refer to the same discourse entity (Recasens et al., 2010).

RelaxCor (Sapena et al., 2010) is a graph representation of the problem solved by a relaxation labeling process, reducing coreference resolution to a graph partitioning problem given a set of constraints. In this manner, decisions are taken considering the whole set of mentions, ensuring consistency and avoiding that classification decisions are independently taken.

The paper is organized as follows. Section 2 describes RelaxCor, the system used in the Semeval task. Next, Section 3 describes the tuning needed by the system to adapt it to different languages and other task issues. The same section also analyzes the obtained results. Finally, Section 4 concludes the paper.

2 System Description

This section briefly describes RelaxCor. First, the graph representation is presented. Next, there is an explanation of the methodology used to learn constraints and train the system. Finally, the algorithm used for resolution is described.

2.1 Problem Representation

Let $G = G(V, E)$ be an undirected graph where V is a set of vertices and E a set of edges. Let $\mathbf{m} = (m_1, \dots, m_n)$ be the set of mentions of a document with n mentions to resolve. Each mention m_i in the document is represented as a vertex $v_i \in V$ in the graph. An edge $e_{ij} \in E$ is added to the graph for pairs of vertices (v_i, v_j) representing the possibility that both mentions corefer.

Let C be our set of constraints. Given a pair of mentions (m_i, m_j) , a subset of constraints $C_{ij} \subseteq C$ restrict the compatibility of both mentions. C_{ij} is used to compute the weight value of the edge connecting v_i and v_j . Let $w_{ij} \in W$ be the weight of the edge e_{ij} :

$$w_{ij} = \sum_{k \in C_{ij}} \lambda_k f_k(m_i, m_j) \quad (1)$$

where $f_k(\cdot)$ is a function that evaluates the constraint k and λ_k is the weight associated to the constraint. Note that λ_k and w_{ij} can be negative.

In our approach, each vertex (v_i) in the graph is a variable (v_i) for the algorithm. Let L_i be the number of different values (labels) that are possible for v_i . The possible labels of each variable are the partitions that the vertex can be assigned. A vertex with index i can be in the first i partitions (i.e. $L_i = i$).

<p>Distance and position:</p> <p>DIST: Distance between m_i and m_j in sentences: number</p> <p>DIST.MEN: Distance between m_i and m_j in mentions: number</p> <p>APPOSITIVE: One mention is in apposition with the other: y,n</p> <p>I/J.IN.QUOTES: m_i/j is in quotes or inside a NP or a sentence in quotes: y,n</p> <p>I/J.FIRST: m_i/j is the first mention in the sentence: y,n</p>
<p>Lexical:</p> <p>I/J.DEF.NP: m_i/j is a definitive NP: y,n</p> <p>I/J.DEM.NP: m_i/j is a demonstrative NP: y,n</p> <p>I/J.INDEF.NP: m_i/j is an indefinite NP: y,n</p> <p>STR.MATCH: String matching of m_i and m_j: y,n</p> <p>PRO.STR: Both are pronouns and their strings match: y,n</p> <p>PN.STR: Both are proper names and their strings match: y,n</p> <p>NONPRO.STR: String matching like in Soon et al. (2001) and mentions are not pronouns: y,n</p> <p>HEAD.MATCH: String matching of NP heads: y,n</p>
<p>Morphological:</p> <p>NUMBER: The number of both mentions match: y,n,u</p> <p>GENDER: The gender of both mentions match: y,n,u</p> <p>AGREEMENT: Gender and number of both mentions match: y,n,u</p> <p>I/J.THIRD.PERSON: m_i/j is 3rd person: y,n</p> <p>PROPER.NAME: Both mentions are proper names: y,n,u</p> <p>I/J.PERSON: m_i/j is a person (pronoun or proper name in a list): y,n</p> <p>ANIMACY: Animacy of both mentions match (persons, objects): y,n</p> <p>I/J.REFLEXIVE: m_i/j is a reflexive pronoun: y,n</p> <p>I/J.TYPE: m_i/j is a pronoun (p), entity (e) or nominal (n)</p>
<p>Syntactic:</p> <p>NESTED: One mention is included in the other: y,n</p> <p>MAXIMALNP: Both mentions have the same NP parent or they are nested: y,n</p> <p>I/J.MAXIMALNP: m_i/j is not included in any other mention: y,n</p> <p>I/J.EMBEDDED: m_i/j is a noun and is not a maximal NP: y,n</p> <p>BINDING: Conditions B and C of binding theory: y,n</p>
<p>Semantic:</p> <p>SEMCLASS: Semantic class of both mentions match: y,n,u (the same as (Soon et al., 2001))</p> <p>ALIAS: One mention is an alias of the other: y,n,u (only entities, else unknown)</p> <p>I/J.SRL.ARG: Semantic role of m_i/j: N,0,1,2,3,4,M,L</p> <p>SRL.SAMEVERB: Both mentions have a semantic role for the same verb: y,n</p>

Figure 1: Feature functions used.

2.2 Training Process

Each pair of mentions (m_i, m_j) in a training document is evaluated by the set of feature functions shown in Figure 1. The values returned by these functions form a positive example when the pair of mentions corefer, and a negative one otherwise. Three specialized models are constructed depending on the type of anaphor mention (m_j) of the pair: pronoun, named entity or nominal.

A decision tree is generated for each specialized model and a set of rules is extracted with C4.5 rule-learning algorithm (Quinlan, 1993). These rules are our set of constraints. The C4.5rules algorithm generates a set of rules for each path from the learned tree. It then checks if the rules can be generalized by dropping conditions.

Given the training corpus, the weight of a constraint C_k is related with the number of examples where the constraint applies A_{C_k} and how many of them corefer C_{C_k} . We define λ_k as

the weight of constraint C_k calculated as follows:

$$\lambda_k = \frac{C_{C_k}}{A_{C_k}} - 0.5$$

2.3 Resolution Algorithm

Relaxation labeling (Relax) is a generic name for a family of iterative algorithms which perform function optimization, based on local information (Hummel and Zucker, 1987). The algorithm solves our weighted constraint satisfaction problem dealing with the edge weights. In this manner, each vertex is assigned to a partition satisfying as many constraints as possible. To do that, the algorithm assigns a probability for each possible label of each variable. Let $\mathbf{H} = (\mathbf{h}^1, \mathbf{h}^2, \dots, \mathbf{h}^n)$ be the weighted labeling to optimize, where each \mathbf{h}^i is a vector containing the probability distribution of v_i , that is: $\mathbf{h}^i = (h_1^i, h_2^i, \dots, h_{L_i}^i)$. Given that the resolution process is iterative, the probability for label l of variable v_i at time step t is $h_l^i(t)$, or simply h_l^i when the time step is not relevant.

```

Initialize:
  H := H0,

Main loop:
  repeat
  For each variable vi
  For each possible label l for vi
    Sil = ∑j∈A(vi) wij × hlj
  End for
  For each possible label l for vi
    hli(t + 1) =  $\frac{h_l^i(t) \times (1 + S_{il})}{\sum_{k=1}^{L_i} h_k^i(t) \times (1 + S_{ik})}$ 
  End for
  End for
  Until no more significant changes

```

Figure 2: Relaxation labeling algorithm

The support for a pair variable-label (S_{il}) expresses how compatible is the assignment of label l to variable v_i taking into account the labels of adjacent variables and the edge weights. The support is defined as the sum of the edge weights that relate variable v_i with each adjacent variable v_j multiplied by the weight for the same label l of variable v_j : $S_{il} = \sum_{j \in A(v_i)} w_{ij} \times h_l^j$ where w_{ij} is the edge weight obtained in Equation 1 and vertex v_i has $|A(v_i)|$ adjacent vertices. In our version of the algorithm for coreference resolution $A(v_i)$ is the list of adjacent vertices of v_i but only considering the ones with an index $k < i$.

The aim of the algorithm is to find a weighted labeling such that global consistency is maximized. Maximizing global consistency is defined

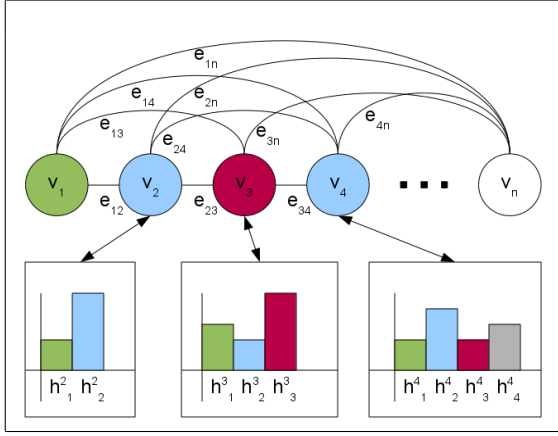


Figure 3: Representation of Relax. The vertices representing mentions are connected by weighted edges e_{ij} . Each vertex has a vector \mathbf{h}^i of probabilities to belong to different partitions. The figure shows \mathbf{h}^2 , \mathbf{h}^3 and \mathbf{h}^4 .

as maximizing the average support for each variable. The final partitioning is directly obtained from the weighted labeling \mathbf{H} assigning to each variable the label with maximum probability.

The pseudo-code of the relaxation algorithm can be found in Figure 2. The process updates the weights of the labels in each step until convergence, i.e. when no more significant changes are done in an iteration. Finally, the assigned label for a variable is the one with the highest weight. Figure 3 shows an example of the process.

3 Semeval task participation

RelaxCor have participated in the Semeval task for English, Catalan and Spanish. The system does not detect the mentions of the text by itself. Thus, the participation has been restricted to the gold-standard evaluation, which includes the manual annotated information and also provides the mention boundaries.

All the knowledge required by the feature functions (Figure 1) is obtained from the annotations of the corpora and no external resources have been used, with the exception of WordNet (Miller, 1995) for English. In this case, the system has been run two times for English: English-open, using WordNet, and English-closed, without WordNet.

3.1 Language and format adaptation

The whole methodology of RelaxCor including the resolution algorithm and the training process is totally independent of the language of the document. The only parts that need few adjustments are

the preprocess and the set of feature functions. In most cases, the modifications in the feature functions are just for the different format of the data for different languages rather than for specific language issues. Moreover, given that the task includes many information about the mentions of the documents such as part of speech, syntactic dependency, head and semantic role, no preprocess has been needed.

One of the problems we have found adapting the system to the task corpora was the large amount of available data. As described in Section 2.2, the training process generates a feature vector for each pair of mentions into a document for all the documents of the training data set. However, the great number of training documents and their length overwhelmed the software that learns the constraints. In order to reduce the amount of pair examples, we run a clustering process to reduce the number of negative examples using the positive examples as the centroids. Note that negative examples are near 94% of the training examples, and many of them are repeated. For each positive example (a corefering pair of mentions), only the negative examples with distance less than a threshold d are included in the final training data. The distance is computed as the number of different values inside the feature vector. After some experiments over development data, the value of d was assigned to 3. Thus, the negative examples were discarded when they have more than three features different than any positive example.

Our results for the development data set are shown in Table 1.

3.2 Results analysis

Results of RelaxCor for the test data set are shown in Table 2. One of the characteristics of the system is that the resolution process always takes into account the whole set of mentions and avoids any possible pair-linkage contradiction as well as forces transitivity. Therefore, the system favors the precision, which results on high scores with metrics CEAF and B^3 . However, the system is penalized with the metrics based on pair-linkage, specially with MUC. Although RelaxCor has the highest precision scores even for MUC, the recall is low enough to finally obtain low scores for F_1 .

Regarding the test scores of the task comparing with the other participants (Recasens et al., 2010), RelaxCor obtains the best performances for Cata-

-	CEAF			MUC			B ³		
language	R	P	F ₁	R	P	F ₁	R	P	F ₁
ca	69.7	69.7	69.7	27.4	77.9	40.6	67.9	96.1	79.6
es	70.8	70.8	70.8	30.3	76.2	43.4	68.9	95.0	79.8
en-closed	74.8	74.8	74.8	21.4	67.8	32.6	74.1	96.0	83.7
en-open	75.0	75.0	75.0	22.0	66.6	33.0	74.2	95.9	83.7

Table 1: Results on the development data set

-	CEAF			MUC			B ³			BLANC		
language	R	P	F ₁	R	P	F ₁	R	P	F ₁	R	P	Blanc
Information: closed Annotation: gold												
ca	70.5	70.5	70.5	29.3	77.3	42.5	68.6	95.8	79.9	56.0	81.8	59.7
es	66.6	66.6	66.6	14.8	73.8	24.7	65.3	97.5	78.2	53.4	81.8	55.6
en	75.6	75.6	75.6	21.9	72.4	33.7	74.8	97.0	84.5	57.0	83.4	61.3
Information: open Annotation: gold												
en	75.8	75.8	75.8	22.6	70.5	34.2	75.2	96.7	84.6	58.0	83.8	62.7

Table 2: Results of the task

lan (CEAF and B³), English (closed: CEAF and B³; open: B³) and Spanish (B³). Moreover, RelaxCor is the most precise system for all the metrics in all the languages except for CEAF in English-open and Spanish. This confirms the robustness of the results of RelaxCor but also remarks that more knowledge or more information is needed to increase the recall of the system without losing this precision

The incorporation of WordNet to the English run is the only difference between English-open and English-closed. The scores are slightly higher when using WordNet but not significant. Analyzing the MUC scores, note that the recall is improved, while precision decreases a little which corresponds with the information and the noise that WordNet typically provides.

The results for the test and development are very similar as expected, except the Spanish (es) ones. The recall considerably falls from development to test. It is clearly shown in the MUC recall and also is indirectly affecting on the other scores.

4 Conclusion

The participation of RelaxCor to the Semeval coreference resolution task has been useful to evaluate the system in multiple languages using data never seen before. Many published systems typically use the same data sets (ACE and MUC) and it is easy to unintentionally adapt the system to the corpora and not just to the problem. This kind of tasks favor comparisons between systems with the same framework and initial conditions.

The results obtained confirm the robustness of the RelaxCor, and the performance is considerably good in the state of the art. The system avoids con-

tradictions in the results which causes a high precision. However, more knowledge is needed about the mentions in order to increase the recall without losing that precision. A further error analysis is needed, but one of the main problem is the lack of semantic information and world knowledge specially for the nominal mentions – the mentions that are NPs but not including named entities neither pronouns–.

Acknowledgments

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement number 247762 (FAUST), and from the Spanish Science and Innovation Ministry, via the KNOW2 project (TIN2009-14715-C04-04).

References

- R. A. Hummel and S. W. Zucker. 1987. On the foundations of relaxation labeling processes. pages 585–605.
- G.A. Miller. 1995. WordNet: a lexical database for English.
- J.R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- M. Recasens, L. Màrquez, E. Sapena, M.A. Martí, M. Taulé, V. Hoste, M. Poesio, and Y. Versley. 2010. SemEval-2010 Task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010)*, Uppsala, Sweden.
- E. Sapena, L. Padró, and J. Turmo. 2010. A Global Relaxation Labeling Approach to Coreference Resolution. *Submitted*.
- W.M. Soon, H.T. Ng, and D.C.Y. Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.