TANL-1: Coreference Resolution by Parse Analysis and Similarity Clustering

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
attardi@di.unipi.it

Stefano Dei Rossi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
deirossi@di.unipi.it

Maria Simi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
simi@di.unipi.it

Abstract

Our submission to the Semeval 2010 task on coreference resolution in multiple languages is based on parse analysis and similarity clustering. The system uses a binary classifier, based on Maximum Entropy, to decide whether or not there is a relationship between each pair of mentions extracted from a textual document. Mention detection is based on the analysis of the dependency parse tree.

1 Overview

Coreference resolution can be described as the problem of clustering noun phrases (NP), also called *mentions*, into sets referring to the same discourse entity.

The "Coreference Resolution in Multiple Languages task" at SemEval-2010 is meant to assess different machine learning techniques in a multilingual context, and by means of different evaluation metrics. Two different scenarios are considered: a *gold standard* scenario (only available for Catalan and Spanish), where correct mention boundaries are provided to the participants, and a *regular* scenario, where mention boundaries are to be inferred from other linguistic annotations provided in the input data. In particular the linguistic annotations provided for each token in a sentence are: position in sentence (ID), word (TOKEN), lemma and predicted lemma (LEMMA and PLEMMA), morpho-syntactic information, both gold and/or predicted (POS and PPOS, FEAT and PFEAT), dependency parsing annotations (HEAD and PHEAD, DEPREL and PDEPREL), named entities (NE and PNE), and semantic roles (PRED, PPRED, and corresponding roles in the following columns). In the gold scenario, mention boundaries annotations (in column COREF) can also be used as input.

Our approach to the task was to split coreference resolution into two sub-problems: mention identification and creation of entities. Mention recognition was based on the analysis of parse trees produced from input data, which were produced by manual annotation or state-of-the-art dependency parsers. Once the mentions are identified, coreference resolution involves partitioning them into subsets corresponding to the same entity. This problem is cast into the binary classification problem of deciding whether two given mentions are coreferent. A Maximum Entropy classifier is trained to predict how likely two mentions refer to the same entity. This is followed by a greedy procedure whose purpose is to cluster mentions into entities.

According to Ng (2005), most learning based coreference systems can be defined by four elements: the *learning algorithm* used to train the coreference classifier, the *method of creating training instances* for the learner, the *feature set* used to represent a training or test instance, and the *clustering algorithm* used to coordinate the coreference classification decisions. In the following we will detail our approach by making explicit the strategies used in each of above mentioned components.

The data model used by our system is based on the concepts of *entity* and *mention*. The collection of mentions referring to the same object in a document forms an *entity*. A mention is an instance referring to an object: it is represented by the *start* and *end* positions in a sentence, a type and a sequence number. For convenience it also contains a frequency count and a reference to the containing sentence.

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 108–111,
Uppsala, Sweden, 15-16 July 2010.

2 Mention detection

The first stage of the coreference resolution process tries to identify the occurrence of mentions in documents.

In the training phase mentions are obtained from the NE (or PNE) column of the corpus and are partitioned into entities using the information provided in the COREF column.

In the regular setting, we used an algorithm for predicting boundaries that relies on the parse tree of the sentence produced from the gold annotations in columns HEAD and DEP, if available, or else from columns PHEAD and PDEP, the output of a dependency parser provided as input data.

This analysis relied on minimal language knowledge, in order to determine possible heads of sub-trees counting as mentions, i.e. noun phrases or adverbial phrases referring to quantities, times and locations. POS tags and morphological features, when available, were mostly taken into account in determining mention heads. The leaves of the sub-trees of each detected head were collected as possible mentions.

The mentions identified by the NE column were then added to this set, discarding duplicates or partial overlaps. Partial overlaps in principle should not occur, but were present occasionally in the data. When this occurred, we applied a strategy to split them into a pair of mentions.

The same mention detection strategy was used also in the gold task, where we could have just returned the boundaries present in the data, scoring 100% in accuracy. This explains the small loss in accuracy we achieved in mention identification in the gold setting.

Relying on parse trees turned out to be quite effective, especially for languages where gold parses where available. For some other languages, the strategy was less effective. This was due to different annotation policies across different languages, and, in part, to inconsistencies in the data. For example in the Italian data set, named entities may include prepositions, which are typically the head of the noun phrase, while our strategy of looking for noun heads leaves the preposition out of the mention boundaries. Moreover this strategy obviously fails when mentions span across sentences as was the case, again, for Italian.

3 Determining coreference

For determining which mentions belong to the same entity, we applied a machine learning technique. We trained a Maximum Entropy classifier written in Python (Le, 2004) to determine whether two mentions refer to the same entity.

We did do not make any effort to optimize the number of training instances for the pair-wise learner: a positive instance is created for each anaphoric NP, paired with each of its antecedents with the same number, and a negative instance is created by pairing each NP with each of its preceding non-coreferent noun phrases.

The classifier is trained using the following features, extracted for each pair of mentions.

Lexical features

- *Same:* whether two mentions are equal;
- *Prefix:* whether one mention is a prefix of the other;
- *Suffix:* whether one mention is a suffix of the other;
- *Acronym:* whether one mention is the acronym of the other.
- *Edit distance:* quantized editing distance between two mentions.

Distance features

- *Sentence distance:* quantized distance between the sentences containing the two mentions;
- *Token distance:* quantized distance between the start tokens of the two mentions;
- *Mention distance:* quantized number of other mentions between two mentions.

Syntax features

- *Head:* whether the heads of two mentions have the same POS;
- *Head POS:* pairs of POS of the two mentions heads;

Count features

- *Count*: pairs of quantized numbers, each counting how many times a mention occurs.

Type features

- *Type*: whether two mentions have the same associated NE (Named Entity) type.

Pronoun features

When the most recent mention is a pronominal anaphora, the following features are extracted:

- *Gender*: pair of attributes {female, male or undetermined};
- *Number*: pair of attributes {singular, plural, undetermined};
- *Pronoun type*: this feature is language dependent and represents the type of pronominal mention, i.e. whether the pronoun is *reflexive*, *possessive*, *relative*, …

In the submitted run we used the GIS (Generalized Iterative Scaling) algorithm for parameter estimation, with 600 iterations, which appeared to provide better results than using L-BFGS (a limited-memory algorithm for unconstrained optimization). Training times ranged from one minute for German to 8 minutes for Italian, hence the slower speed of GIS was not an issue.

3.1 Entity creation

The mentions detected in the first phase were clustered, according to the output of the classifier, using a greedy clustering algorithm.

Each mention is compared to all previous mentions, which are collected in a global mentions table. If the pair-wise classifier assigns a probability greater than a given threshold to the fact that a new mention belongs to a previously identified entity, it is assigned to that entity. In case more than one entity has a probability greater than the threshold, the mention is assigned to the one with highest probability. This strategy has been described as *best-first clustering* by Ng (2005).

In principle the process is not optimal since, once a mention is assigned to an entity, it cannot be later assigned to another entity to which it more likely refers. Luo et al. (2004) propose an approach based on the Bell tree to address this problem. Despite this potential limitation, our system performed quite well.

4 Data preparation

We used the data as supplied by the task organizers for all languages except Italian. A modified version of the Hunpos tagger (Halácsy, Kornai & Oravecz, 2007; Attardi et al., 2009) was used to add to the Italian training and development corpora more accurate POS tags than those supplied, as well as missing information about morphology. The POS tagger we used, in fact is capable of tagging sentences with detailed POS tags, which include morphological information; this was added to column PFEATS in the data. Just for this reason our submission for Italian is to be considered an open task submission.

The Italian training corpus appears to contain several errors related to mention boundaries. In particular there are cases of entities starting in a sentence and ending in the following one. This appears to be due to sentence splitting (for instance at semicolons) performed after named entities had been tagged. As explained in section 2, our system was not prepared to deal with these situations.

Other errors in the annotations of entities occurred in the Italian test data, in particular incorrect balancing of openings and closings named entities, which caused problems to our submission. We could only complete the run after the deadline, so we could only report unofficial results for Italian.

5 Results

We submitted results to the gold and regular challenges for the following languages: Catalan, English, German and Spanish.

Table 1 summarizes the performance of our system, according to the different accuracy scores for the gold task, Table 2 for the regular task. We have outlined in bold the cases where we achieved the best scores among the participating systems.

	Mention	**CEAF**	**MUC**	**B^3**	**BLANC**
Catalan	98.4	64.9	26.5	76.2	54.4
German	**100**	**77.7**	25.9	**85.9**	57.4
English	89.8	67.6	24.0	73.4	52.1
Spanish	98.4	65.8	25.7	76.8	54.1

Table 1. Gold task, Accuracy scores.

	Mention	**CEAF**	**MUC**	**B^3**	**BLANC**
Catalan	**82.7**	**57.1**	22.9	**64.6**	51.0
German	59.2	49.5	15.4	50.7	44.7
English	73.9	57.3	24.6	61.3	49.3
Spanish	**83.1**	**59.3**	21.7	**66.0**	**51.4**

Table 2. Regular task. Accuracy scores.

6 Error analysis

We performed some preliminary error analysis. The goal was to identify systematic errors and possible corrections for improving the performance of our system.

We limited our analysis to the mention boundaries detection for the regular tasks. A similar

analysis for coreference detection, would require the availability of gold test data.

7 Mention detection errors

As described above, the strategy used for the extraction of mentions boundaries is based on dependency parse trees and named entities. This proved to be a good strategy in some languages such as Catalan (F1 score: 82.7) and Spanish (F1 score: 83.1) in which the dependency data available in the corpora were very accurate and consistent with the annotation of named entities. Instead, there have been unexpected problems in other languages like English or German, where the dependencies information were annotated using a different approach.

For German, while we achieved the best B^3 accuracy on coreference analysis in the gold settings, we had a quite low accuracy in mention detection (F1: 59.2), which was responsible of a significant drop in coreference accuracy for the regular task. This degradation in performance was mainly due to punctuations, which in German are linked to the sub-tree containing the noun phrase rather than to the root of the sentence or tokens outside the noun phrase, as it happens in Catalan and Spanish. This misled our mention detection algorithm to create many mentions with wrong boundaries, just because punctuation marks were included.

In the English corpus different conventions were apparently used for dependency parsing and named entity annotations (Table 3), which produced discrepancies between the boundaries of the named entities present in the data and those predicted by our algorithm. This in turn affected negatively the coreference detection algorithm that uses both types of information.

ID	TOKEN	HEAD	DEPREL	NE	COREF
1	Defense	2	NAME	(org)	(25
2	Secretary	4	NMOD	_	_
3	William	4	NAME	(person	_
4	Cohen	5	SBJ	person)	25)

Table 3. Example of different conventions for NE and COREF in the English corpus.

Error analysis also has shown that further improvements could be obtained, for all languages, by using more accurate language specific extraction rules. For example, we missed to consider a number of specific POS tags as possible identifiers for the head of noun phrases. By some simple tuning of the algorithm we obtained some improvements.

8 Conclusions

We reported our experiments on coreference resolution in multiple languages. We applied an approach based on analyzing the parse trees in order to detect mention boundaries and a Maximum Entropy classifier to cluster mentions into entities.

Despite a very simplistic approach, the results were satisfactory and further improvements are possible by tuning the parameters of the algorithms.

References

G. Attardi et al., 2009. Tanl (Text Analytics and Natural Language Processing). SemaWiki project: http://medialab.di.unipi.it/wiki/SemaWiki.

P. Halácsy, A. Kornai, and C. Oravecz, 2007. HunPos: an open source trigram tagger. *Proceedings of the ACL 2007*, Prague.

Z. Le, Maximum Entropy Modeling Toolkit for Pytho and C++, Reference Manual.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla & S. Roukos. 2004. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. *Proceedings of the ACL 2004*, Barcelona.

V. Ng, Machine Learning for Coreference Resolution: From Local Classification to Global Ranking, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (ACL), Ann Arbor, MI, June 2005, pp. 157-164.

M. Recasens, L. Màrquez, E. Sapena, M. A. Martí, M. Taulé, V. Hoste, M. Poesio and Y. Versley, SemEval-2010 Task 1: Coreference resolution in multiple languages, in Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010), Uppsala, Sweden, 2010.