UNIPI participation to the Anaphora Resolution Task

Giuseppe Attardi, Stefano Dei Rossi, Maria Simi

Università di Pisa, Dipartimento di Informatica, Largo B. Pontecovo 3,
56127 Pisa, Italy
{attardi, deirossi, simi}@di.unipi.it

Abstract. The system used in the Evalita 2011 Anaphora Resolution Task is based on dependency parse tree analysis and similarity clustering. Mention detection relies on the analysis of dependency parse trees obtained by re-parsing texts with DeSR, and on some ad-hoc heuristics to deal with specific cases, where mentions boundaries do not correspond to sub-trees. The system then uses a binary classifier, based on Maximum Entropy, to decide whether there is a co-reference relationship between each pair of mentions extracted in the previous phase. Clustering of entities is performed by a greedy clustering algorithm.

Keywords: Anaphora resolution, maximum entropy, similarity clustering, parse analysis, mention detection

1 Description of the System

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 Evalita 2011 task is similar to the "Coreference Resolution in Multiple Languages task" at SemEval-2010, whose main goal was to assess different machine learning techniques in a multilingual context, and by means of different evaluation metrics. At SemEval-2010 two different scenarios were considered: a *gold standard* scenario, where correct mention boundaries were provided to participants, and a *regular* scenario, where mention boundaries had to be inferred from other linguistic annotations provided in the input data. In particular lemmas, PoS tags, morphology, dependency-parsing annotations, named entities (NE), and semantic roles were provided in both gold and predicted form and could be used in both scenarios.

Our team participated in SemEval-2010 and our system obtained top scores in the following tasks and languages: German in the gold standard scenario, and Catalan and Spanish, in the regular scenario [1]. At SemEval, we were not able to complete a run for the Italian language: several errors occurred in the Italian training corpus (mention boundaries spanning several sentences, incorrect balancing in opening and closing named entities) and our system was not robust enough to deal with these anomalies.

Our main motivation in participating in the Evalita-2011 Anaphora task was to test our system on the new Italian corpus, adapt the system to Italian and improve on previous results obtained in other languages.

The Evalita-2011 task however turned out to be quite different and we had to make a substantial amount of adjustments in order to deal with the novelties introduced by the task. In particular only the full task corresponding to the regular task of SemEval was organized. As a consequence it will also be hard to compare the results obtained with previous results in other languages.

Our approach to the task is to split co-reference resolution into two sub-problems: mention identification and entities clustering. Mention recognition is based on the analysis of parse trees.

Once the mentions are identified, co-reference 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 co-referent. 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 [2], most learning based co-reference systems can be defined by four elements: the *learning algorithm* used to train the co-reference 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 co-reference 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.

1.1 Mention Detection

The first stage of the co-reference resolution process tries to identify the occurrence of mentions in documents.

For predicting mention boundaries we tried the same strategy used in SemEval, i.e. relying on the output of the dependency parser provided as input data. This approach in fact had turned out to be quite effective, especially for languages where gold parses where available. For some other languages, the strategy was less effective, due to different annotation policies, and, in part, to inconsistencies in the data.

In this case no gold data was provided, only predicted, and in particular the system output of the dependency parser made available with the data was not accurate enough to rely on. Moreover, unlike other corpora for other languages we have dealt with, there was no clear correspondence between the NP marked as mentions and sub-trees of the parse tree. Finally, the NEs (or PNEs), that were given in the SemEval task and also in this training set, could not be taken into account because they were not present in the test set. NEs would have been really important both to help the mention extractor to detect mention boundaries, and to add a type classification label to each mention to be exploited as feature by the co-reference tagger.

We tried to address these problems by:

- Retagging lemmas, PoS and parser columns using the Tanl Suite [3];
- Adding some heuristics to:
  - check the alignment between given and re-tagged PoS;
  - detect cases in which mentions exceeds NPs boundaries.

The mention extraction strategy 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 (after re-tagging), were mostly taken into account in determining mention heads. The leaves of the sub-trees of each detected head were collected as possible mentions, then some heuristics and rules were applied to address problems related to mention boundaries exceeding NPs limits. Because of the lack of detailed and self-contained annotation guidelines those heuristics and rules were mainly inferred from the comparison between the gold co-reference column of the training set and the output of our mention extractor.

Another unexpected problem was the presence of about 1200 verbs (mainly verbs with implicit subjects or clitics), marked as one-token mention in the training set, that could not be identified using the algorithm described above. A Maximum Entropy base tagger, the Tanl tagger [4], was trained on the training set with the aim to predict those kind of entities. The Tanl tagger is a generic and customizable text chunker, which can be applied to several tasks such as POS tagging, Super-sense tagging and Named Entity Recognition. The modular architecture of the chunker offers the possibility to specify the features to extract using a textual configuration file. In particular, to train this model we used the following local features:

- *Features of Current Word*: first word of sentence and capitalized; first word of sentence and not capitalized; two parts joined by a hyphen.
- *Features from Surrounding Words*: both previous, current and following words are capitalized; both current and following words are capitalized; both current and previous words are capitalized; word is in a sequence within quotes.

And the following attributes features:

Table 1. Attributes features

Attributes	Positional values
FORM	0
POSTAG	-2 -1 0 1 2
CPOSTAG	-1 0

where CPOSTAG is the coarse-grained POSTAG (the first letter of the POS) and the values represent the position of the attribute with respect to the current token. No dictionaries o gazetteers were used and the best results on the development set were achieved with 250 iterations of the Maximum Entropy algorithm.

Heuristic Rules and Runs. Two different intermediate files were created as output of the mention extractor both sharing almost the same set of heuristics and post

processing rules. The following PoS were considered as heads of mentions: *common nouns, proper nouns*, *personal pronouns*, *demonstrative pronouns*, *indefinite pronouns*, *possessive pronouns*.

The following heuristics and rules were applied in both runs:

- include articulated preposition at the beginning of the mention;
- stop mention expansion on adverb;
- add dates and years as mentions;
- exclude clitic pronouns at the beginning of the mention;
- add verbs identified by the ME classifier in the guided procedure described above;
- stop right mention expansion on balanced punctuation and on commas when the parser relation is copulative conjunction;
- remove articulated preposition and relative pronoun from the right boundary of mentions;
- remove preposition and balanced punctuation from the left boundary of mentions;

Moreover the following more restrictive rules were applied in run 1, in an attempt to improve precision excluding some sub-cases:

- do not consider as head of NPs a proper noun when its dependency relation is "concatenation";
- do not consider as head of NPs each PoS different from nouns and pronouns when the associated dependency relation is "modifier";

1.2 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 [5] 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.

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 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 200 iterations, which appeared to provide better results than using L-BFGS (a limited-memory algorithm for unconstrained optimization).

1.3 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 [2].

2 Results

Table 2. UniPI systems results for Run 1 and Run 2

	Run 1			Run 2		
	Recall	**Precision**	**FB1**	**Recall**	**Precision**	**FB1**
Ident. of ment.	64.01%	62.11%	63.04	64.12%	59.36%	61.65
MUC	18.38 %	46.59%	26.36	17.83 %	42.21%	25.07
B-CUB	75.69%	93.83%	83.79	75.96%	93.04%	83.64
CEAFm	72.99%	72.99%	72.99	72.53%	72.53%	72.53
CEAFe	87.64%	71.72%	78.89	86.53%	71.64%	78.38
BLANC	53.75%	64.66%	55.94	53.66%	64.38%	55.80

3 Discussion

The scorer would have been really important for the tuning of the system but the official scorer was not made available in that phase. Therefore we resorted to using the SemEval-2010 scorer. The two scorers unfortunately use the same metrics but different approaches. The Evalita scorer seems to use a more strict approach in the evaluation of mentions but, as stated in the task guidelines, it is more tolerant in coreference evaluation since it allows a partial alignment between system and gold mentions. The difference between the two scorers is significant, as shown in Table 4.

Table 3. Differences between SemEval and Evalita scorers

	Scorer SemEval		**Scorer Evalita**	
	dev	**test**	**dev**	**test**
Identification of mentions	71.83	67.34	64.21	63.04
Coreference (B-CUB)	65.99	59.37	84.74	83.79

The performance of our system is quite disappointing when compared to the results obtained in SemEval 2010 with other languages and resources. And unfortunately in this task we are not able to compare our results with other systems since we were the only ones participating. The following consideration, however, are in order:

1. the identification of mentions proved to be more difficult with respect to SemEval 2010 due to the following factors:
   - PoS, lemmas and parsing information were system predicted and not gold;
   - some heuristics that behaved well on the development set were not effective on the test set, due to our own poor understanding of annotation guidelines;
   - somewhat surprisingly, the model we trained to recognize verbs that are also mentions, failed badly to predict on the test set: 29% recall, 18% precision.
2. the coreference results are very high but cannot be compared with the results obtained in SemEval-2010 Coreference Task because the scorer is different and more tolerant (it allows also partial alignment between system and gold mentions).

References

1. Attardi, G., Dei Rossi, S., Simi, M.: TANL-1: Coreference Resolution by Parse Analysis and Similarity Clustering. In: Proceedings of SemEval 2010, Uppsala (2010)
2. Ng, V.: Machine Learning for Coreference Resolution: From Local Classification to Global Ranking. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 157-164, Ann Arbor, MI (2005)
3. Attardi, G. et al.: Tanl (Text Analytics and Natural Language Processing). SemaWiki project, http://medialab.di.unipi.it/wiki/SemaWiki (2009)
4. Attardi, G., Dei Rossi, S., Dell'Orletta, F., Vecchi E. M.: The Tanl Named Entity Recognizer at Evalita 2009. In: Proceedings of Evalita 2009, Reggio Emilia (2009)
5. Le, Z.: Maximum Entropy Modeling Toolkit for Python and C++, Reference Manual