Local Entity Detection and Recognition Task EVALITA 2009

Silvana Marianela Bernaola Biggio[1], Claudio Giuliano[2], Massimo Poesio[3], Yannick Versley[4], Olga Uryupina[5], and Roberto Zanoli[6]

FBK-irst,
via Sommarive 18, I-38123 Povo (TN), Italy
38100 Trento,Italy
bernaola@fbk.eu, giuliano@fbk.eu, massimo.poesio@unitn.it,
yversley@gmail.com, uryupina@gmail.com, zanoli@fbk.eu

Abstract. This article describes a system that detects and recognizes local entities for the Italian language. The system is divided into 2 modules, the Entity Mention Detection (EMD) module which detects all the mentions related to persons, organizations, geo-political entities and locations; and the Coreference Resolution module that recognizes which mentions refer to the same entity. Understanding *entity* as an object or group of objects in the world; and, mention as the textual reference of an entity. We explain the architecture of both modules and report the results of the system at the EVALITA 2009 campaign.

Keywords: entity mention detection, named entity recognition, mention detection, local coreference, intra-document coreference.

1 Introduction

The Local Entity Detection and Recognition (LEDR) task aims to detect all the entities occurring in a text. An entity is an object or group of objects in the world; while an entity mention is the textual reference of an entity. For example, the entity "*Valentino Rossi*" could be mentioned in a text as "*Rossi*", "*pilota*", "*lui*", "*che*"; and, all these mentions refer to the same entity.

The LEDR task was subdivided into two modules. The Entity Mention Detection (EMD) module, whose task consisted on the detection of entity mentions and their classification in syntactic and semantic classes; i.e. Person (PER), Organization (ORG), Geo-Political Entity (GPE) or Location (LOC). And the coreference resolution module; whose task consisted on the association of each mention with its corresponding entity.

2 Entity Mention Detection Module

The EMD task considers nested mentions, i.e. mentions that include other mentions; for instance, the mention "*La cantante Madonna*" includes other two mentions, "*cantante*" and "*Madonna*". In this case, it is a mention that includes mentions of the same type (Person); however, it is not always the case that the type of the mention coincides with

the type of the included mentions (e.g. the entity "*il presidente del consiglio regionale*" is a nominal (NOM) that refers to a person (PER); includes the entity "*presidente*" which is also a NOM and belongs to the category PER and "*consiglio regionale*" which is a proper name (NAM) that belongs to the category organization (ORG).

2.1 Data

It was used The Italian Corpus Annotation Bank (I-CAB) [1]. The training and development set were composed by 525 news stories taken from the local newspaper "*L'Adige*" belonging to four different days: September 7th, 2004; September 8th, 2004; October 7th, 2004 and October 8th, 2004; grouped in five categories: News stories, Cultural news, Economic news, Sport news and Local news. The test set, instead, was composed by new data taken from the same newspaper.

With respect to the format of the data, two types of files were put at disposition of the competitors:

- UTF-8 text files: Files with extension TXT which contain the text of the document.
- ACE Program Format Files: Files with extension APF which contain the annotation of the document using XML format. As it can be seen in figure 1, for each entity, it is indicated all its attributes (type, subtype and class) as well as all its mentions, specifying for each of them its head and its extent.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE source_file PUBLIC "SYSTEM" "apf.v5.2.0.dtd">

<source_file URI="adige20040907_id405381.txt" SOURCE="unknown" TYPE="text" VERSION="5.0" AUTHOR="Evalita"
ENCODING="UTF-8">
  <document DOCID="adige20040907_id405381">
    <entity ID="adige20040907_id405381-E1" TYPE="PER" SUBTYPE="Group" CLASS="SPC">
      <entity_mention ID="adige20040907_id405381-E1-1" TYPE="NOM" PRIMARY="true" METONYMY_MENTION="FALSE">
        <extent>
          <charseq START="235" END="245">dei parenti</charseq>
        </extent>
        <head>
          <charseq START="239" END="245">parenti</charseq>
        </head>
```

Fig. 1. APF file

2.2 Architecture of the system

The task was divided into the following sub tasks:

- Detection of the heads of the entity mentions

– Syntactic classification of the heads: proper name(NAM), nominal(NOM) or pronoun(PRO)
– Semantic classification of the heads: Person(PER), Organization(ORG), Geo-Political Entity(GPE) or Location(LOC)
– Extraction of the extents of the detected heads

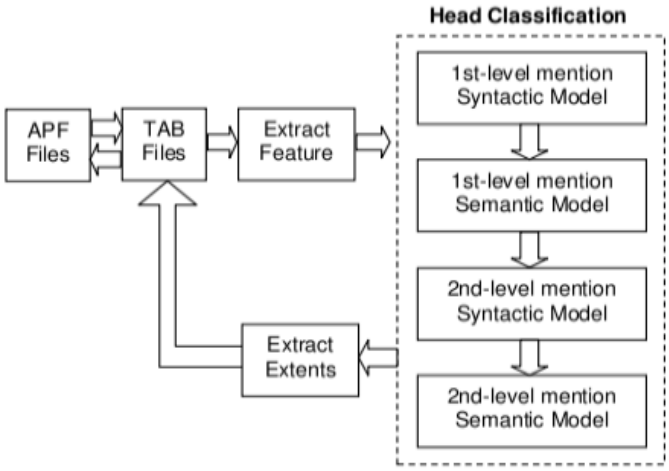

Fig. 2. Architecture of the system

To construct the classification models, it was taken into account the fact that nested mentions (also called second level mentions) should include at least one first level mention.

Figure 2 shows the architecture of the system. To detect and classify the heads of the mentions, it was necessary to select and extract all the important features from the training corpus. To interchange the data between the EMD module and the coreference resolution module, all the XML files (APF files) were converted into a tabular format (TAB files). Four-scalar models were constructed, based on two hypothesis: (1) the semantic classifier performance could improve if it takes into consideration the syntactic mention type, (2) the second-level mention classifier needs to learn that a mention includes other mentions, while it is not the case of the first-level one. This means that the syntactic classifier of first-level mention gives information to the semantic classifier of the same level mention; and these two give information to the syntactic classifier of the second level; and finally, these three give information to the semantic classifier of the second-level mention.

Once all the heads were detected and classified syntactically and semantically, the next step consisted on using MaltParser [2] to obtain the extent of each detected head. Once the entity boundaries and types were available, we used another SVM model to classify the subtypes; using tf-idf to weight the terms and chi-squared for feature selection. Finally, the tabular format file was converted into the original format file.

2.3 Experiments

Figure 3 shows the procedure followed to create the system to Detect Entity Mentions (DEMention). We used Yamcha to train the system, trying different configurations of features (e.g. changing static and dynamic features, the size of the windows, etc.) in order to get the best classifier. It is important to mention that not using the information of the syntactic classifier to construct the semantic classifier, generates inconsistent results; i.e. some tokens were semantically classified but not syntactically and vice versa.

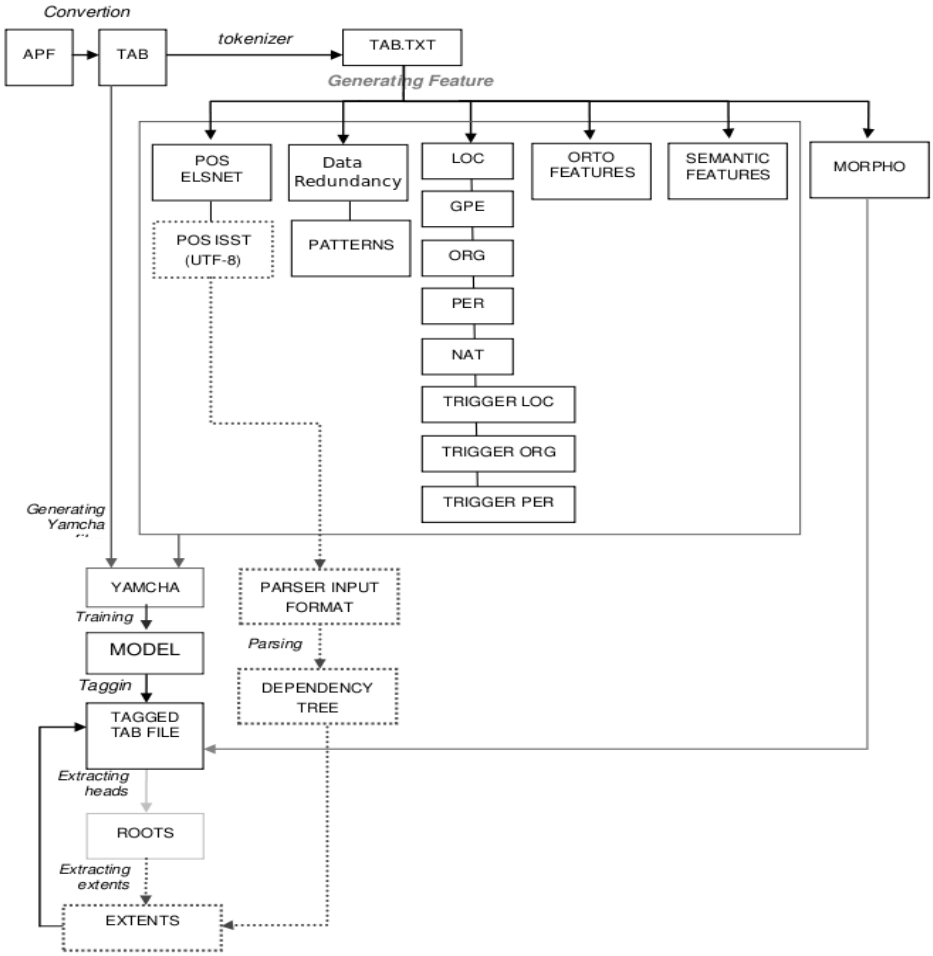

Fig. 3. Entity Mention Detection Procedure

After some experiments, it was observed that the best configuration for the second level classifier was the same as the one taken for the first level classifier. Some

of features used were the token itself, the part-of-speech, the length of the token, the stem, is capitalized, is abbreviation, among others; most of them were obtained through regular expressions; others with the TextPro tool [5] (tokenization, part-of-speech and morphological features); and we also used a large and unannotated text corpus (aprox. one billion words) to obtained two interesting features: *Data Redundancy* and *Patterns*, which improved considerably the performance of the system. The features were obtained with Typhoon, a Named Entity Recognition system that took part at EVALITA 2009 [4]. Table 1 shows the result for the DEMention system.

Table 1. Evaluation of the DEMention System at EVALITA 2009

Measure	Result
Value	65.7%
Precision	78.1%
Recall	74.1%
FB1	76.1%

2.4 Discussion

During the development of the task a lot of issues were detected; for instance, the word *che* is a pronoun when it makes reference to a previous mentioned entity; but, it should be annotated as a pronoun only if it refers to an entity that belongs to the list of entities to annotate (PER, ORG, LOC, GPE). However, the main issue is related to context reference; continuing with the previous example, if the word *che* refers to an entity that has been mentioned in a previous sentence, which is out of the context of the classifier, it is very probably that the classifier does not recognize it as a pronoun. On the other hand, a name can refer to a person sometimes and others, to a location, geopolitical entity and even an organization. For example, the word *Trento* can be a person, an organization or a geopolitical entity, in the sentence: *Trento ha detto che (Trento has said that)*, most probably *"Trento"* refers to a Person, since *"Trento"* is an Italian surname and usually the verb *"said"* is related to persons.

3 Coreference Resolution Module

We have designed a coreference resolution system for Italian based on BART [6]. BART is a modular toolkit for coreference resolution that supports state-of-the-art statistical approaches to the task and enables efficient feature engineering. BART has originally been created and tested for English, but its flexible modular architecture ensures its portability to other languages and domains. The toolkit has four main components: preprocessing pipeline, feature extraction module, decoder and encoder. In addition, an independent *LanguagePlugin* module handles all the language specific information and is accessible from any component. The preprocessing pipeline converts an input

document into a sequence of mentions with assigned properties (number, gender etc). The feature extraction module describes pairs of mentions $\{M_i, M_j\},\ i < j$ as a set of features. Table 2 shows the features used in our Evalita run. All the feature values are computed automatically, without any manual intervention. The decoder generates training examples through a process of sample selection and learns a pairwise classifier. Finally, the encoder generates testing examples through a (possibly distinct) process of sample selection, runs the classifier and partitions the mentions into coreference chains given the classifier decisions. For our Evalita run, we have tested a number of machine learning algorithms and decoding/encoding techniques and have opted for the setting advocated by [3] with the maximum entropy classifier.

Table 2. Features used by our Italian version of BART for the Evalita run: each feature describes a pair of mentions $\{M_i, M_j\},\ i < j$, where M_i is a candidate antecedent and M_j is a candidate anaphor.

MentionType(M_i)
MentionType(M_j)
SemanticClass(M_i)
SemanticClass(M_j)
FirstMention(M_i)
GenderAgreement(M_i,M_j)
NumberAgreement(M_i,M_j)
AnimacyAgreement(M_i,M_j)
Alias(M_i,M_j)
Apposition(M_i,M_j)
StringMatch(M_i,M_j)
Distance(M_i,M_j)

Our work on adapting BART to the Evalita task has followed two directions. First, we have developed a new language plugin to support Italian input in general. Second, we have developed a new preprocessing pipeline to specifically improve the system's performance on the Evalita test set. Our work on the language plugin has mostly included investigating Italian-specific aliasing techniques. A list of company/person designators (e.g., "S.p.a" or "D.ssa") has been manually crafted. We have extracted from the training data several patterns of name variants for the locations (e.g. "Provincia di Verona" and "Verona" refer to the same place). Finally, we have relaxed abbreviation constraints, allowing for lower-case characters in the abbreviations – a pattern that is much more common for Italian than for English. These adjustments have improved the performance level of our aliasing module on the development set by around 3%. We have run several evaluation experiments with the different designs of the preprocessing pipeline to optimize the system's performance on the Evalita dataset. For the testing data, the preprocessing is straightforward: we input all the chunks detected by the mention tagger (cf. above) and assign relevant properties from the output of the corresponding component from TextPro [5] – part-of-speech, morphological features such as number and gender, as well as semantic type. For the training data, however, this

strategy leads to only a moderate performance level for two main reasons. First, manually annotated ("gold") mentions tend to be much longer than those extracted by the tagger ("system mentions"). This means that our matching and aliasing models, learned directly from the gold training data, may not be applicable to automatically extracted testing mentions. To rectify this problem, we have adjusted gold mention boundaries to cover only the heads, not the extents. Second, the training data contain a number of embedding mentions – chunks that span over another mention (e.g. "la popolazione del sobborgo" is a mention of the second level of embedding, as it spans over another mention, "sobborgo"). Our mention tagger can only extract mentions of the first and second level of embedding. We have, therefore, discarded all the gold mentions with the higher level of embedding to avoid unnecessary noise. We have also investigated an alternative parsing pipeline: within this strategy, the chunks, suggested by the mention tagger, are mapped into NP-like nodes in automatically constructed parse trees. To get the trees, we have relied on an Italian version of MaltParser [2]. Our experiments on the development set, however, have shown that, unlike for English, this method brings no improvement over the simpler and faster pipeline described above and even leads to some performance loss. We believe that the state of the art for parsing in Italian is not yet reliable enough – possibly due to the lack of training data for the parser. Note that morphological preprocessing for Italian, on the contrary, is much easier and more accurate, than for English: thus, we can reliably obtain mentions properties (e.g., gender) from a shallow morphological analyzer (TextPro).

To summarize, we have extended BART [6] to create a full-scale coreference resolution system for Italian. Its modular design has allowed us to port a large part of the functionality from English to Italian with no changes – we have only had to run a series of evaluation runs on the development set to pick the best decoding/encoding scheme and the most suitable machine learning algorithm from a range of solutions provided in the BART distribution. We have therefore focused our attention on improving the system's performance by taking care of language-specific properties. Our experiments on the development set have shown that a coreference resolution system based on shallow preprocessing works better for a morphologically rich language, such as Italian, compared to parsing-oriented strategies more common for English.

Acknowledgments. This work has been partially supported by the project LiveMemories, funded by the Provincia Autonoma of Trento.

References

1. Magnini, B., Pianta, E., Speranza, M., Bartalesi Lenzi, V., Sprugnoli, R.: Italian Content Annotation Bank (I-CAB): Named Entities, http://evalita.fbk.eu/doc/I-CAB-Report-Named-Entities.pdf (2007)
2. Lavelli, A., Hall, J., Nilsson, J., Nivre, J.: MaltParser at the EVALITA 2009 Dependency Parsing Task. In: Proceedings of EVALITA 2009. Reggio Emilia, Italy (2009)
3. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics (Special Issue on Computational Anaphora Resolution), vol. 27, issue 4, pp. 521–544 (2001)

4. Zanoli, R., Pianta, E.: Named Entity Recognition through Redundancy Driven Classifiers. In: Proceedings of EVALITA 2009. Reggio Emilia, Italy (2009)
5. Pianta, E., Girardi, C., Zanoli, R.: The TextPro tool suite. In: Proceedings of LREC 2008. Marrakech, Morocco (2008)
6. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: BART: A modular toolkit for coreference resolution. In: Proceedings of LREC 2008 (2008)