Improving cross-document coreference

Octavian Popescu[1], Christian Girardi[1], Emanuele Pianta[1], Bernardo Magnini[1]

[1]FBK-IRST, ITALY

Abstract

In this paper we present a cross document coreference system which resolves some of the more problematic cases by taking into account pieces of evidence coming from different sources. The corpus we work with is a seven-year news collection from a local newspaper. The approach does not assume any prior knowledge about persons (e.g. an ontology) mentioned in the collection and requires basic linguistic processing (named entity recognition) and resources (a dictionary of person names). The system parameters have been estimated on a 5K corpus of Italian news documents. The evaluation, over a sample of four days news documents, shows that the error rate of the system (1.4%) is above a baseline (5.4%) for the task.

Keywords: newspaper, Italian, dictionary of person names, named entity recognition, cross document coreference.

1. Introduction

Finding information about people is likely to be one of the principal interests of someone reading a newspaper. In the web space, person search is a very popular task and, since a person is primarily identified by her/his name, the key of the search is usually the name. However, as names are very ambiguous (Artiles et al. 2007), it is a big challenge for automatic systems to cluster name occurrences in a corpus according to the persons they refer to.

The input of a coreference system is a document collection where person names are already identified; the output is a set of clusters of person names, where each cluster represents a different person.

Person coreference has been addressed in previous work (e.g. Baga and Baldwin 1998, Pederson et al. 2006), building vector-based representations of persons' names which use lists of named entities (e.g. person, locations, organizations) (NE) associated with person names - called the *association set* of a person (AS). The underlying idea is that a person can be identified by the events he/she is associated with, and that such events can be conveniently represented by the list of the named entities mentioned in a particular document. However, in all these approaches all person names have been assumed to behave equally. We will show that important evidence comes from analyzing the name distribution in the corpus. One of the major drawbacks of using ASs is that in many cases there is no control on the relevance of the NEs included. An important step forward in solving this problem is to consider pieces of information which are semantically linked to the names. We extract the information coming from the name determiners, such as profession and nationality.

The approach we propose is based on three main working hypotheses. The first one is that establishing coreference is an iterative process, where at each cycle the system attempts at establishing new coreferences, taking advantage of the coreferences established in previous cycles. When no new coreferences are realized, the algorithm stops. The second consideration that has motivated our work is that we think that the full potential of names has not been exploited yet. It is useful to distinguish between first and last names. They behave differently with respect to coreference (three times less perplexity for last names than for first names) and, as there are items that could be both first and last names, making the distinction helps in reducing false coreference. We identify rare names for which the probability of different persons carrying them is very low and therefore their coreference may be realized by loosening the conditions on the number of named entities in common. Our third working hypothesis is that the determiners of the NPs containing the name mentions constitute very good clues for coreference.

The paper is structured as follows. In Section 2 we present the architecture of our system, introducing the three main modules. Sections 3, 4, and 5 are dedicated to detailed descriptions of the algorithms we have implemented for Name Splitter, Local Coreference and Global Coreference, respectively. In Section 6 we discuss the evaluation methodology and report the results we have obtained. Section 7 presents our future goals and interests.

2. System Architecture

The input of the coreference system is a list of named entities (i.e. *Person Names*, PNs) of type Person, Location, Organization[1], automatically recognized by a Named Entity Recognition (NER) system within a document collection. The output of the system is a number of clusters of the named entities of type Person, where each cluster is interpreted as the set of PNs that refer to the same entity.

The system has three main modules, depicted in Figure 1, corresponding to three steps of the analysis: Person Name Splitter, Local Coreference module and Global Coreference module.

- The *Person Name Splitter* splits the Person Names into first_name and last_name. The motivation is that first_name and last_name are used differently in mentioning persons and play different roles in coreference.
- The *Local Coreference module* establishes which of the PNs occurring within a single document corefer. We use a probabilistic approach, estimating, inside each document, the joint probability of occurrence of any pair formed by two free names (either first or last names) and a certain complete name (first and last name). For each cluster of coreferred PNs, we choose a representing name, which we call *local head*. The set of local heads is the output of this module.
- The *Global Coreference module* tries to establish the coreference among all the local heads of the collection based on their association set (the set of named entities they are associated with) and on the probability that the same PN refers to two different persons (common vs. uncommon names).

Three resources are accessed, and dynamically enriched, during the coreference process: the Name Dictionary, the Entity Ontology and the Topic Ontology.

[1] The Named Entities we are using are defined according to the ACE standard. However, for our purposes, the GPE entities have been considered as Location entities.

- The *Name Dictionary* is a list of first_names and last_names. We start with a Name Dictionary of Italian names which contains 68,654 last_names and 3,230 first_names. There are 1267 ambiguous names, which can be both first and last names.
- The *Entity Ontology* contains all the known entities. We start with an empty Entity Ontology, which, at the end of the coreference process, will contain all the person entities identified in the corpus.
- The *Topic Ontology* is a resource specific for the news domain, where each document of the collection is classified under a limited number of topics.

3. The Name Splitter

This module identifies first_names and last_names in a sequence of tokens composing a PN and selects the most representative token, i.e. the one with the highest probability to be used for identifying the person. For instance, given the PN "Luca Cordero di Montezemolo", the module recognizes that "Luca" is a first_name, that "Cordero" and "Montezemolo" are last_names and that "Montezemolo" is the most representative one. There are at least two relevant issues that make the task challenging: (i) the high proportion of tokens that are ambiguous (i.e. that can be used both as first and last names), which makes their classification difficult. For instance, in Italian, the name "Viola" can be used both as first_name and as last_name; (ii) the presence of first_names of famous people, such as "Michelangelo", that are actually used as last_names. This phenomenon makes the identification of the most representative token difficult.

The Person Name Splitter is based on a multi layer perceptron that, for each token in a PN, assigns a score corresponding to the probability of the token to be a last_name. The perceptron has six input nodes and three hidden nodes (Figure 1).

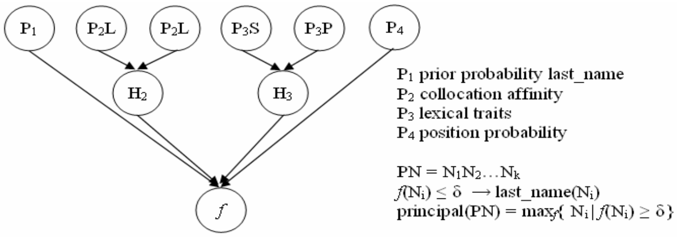


Fig. 1. Last_Name Perceptron

The parameters considered for the input nodes of the perceptron are:

- *P1*: the prior probability of a token of being either first_name or last _name;
- *P2* the probability to be followed on the left (P2L) or on the right(P2R) by a last_name;
- *P3*: the prior probability of the prefix(P3P) and suffix(P3S) to signal a last_name;
- *P4*: the prior probability that a last_name occupies the rightmost position in a PN.

The perceptron has been trained on the set of PNs present in the Name Dictionary (see Section 2) in order to determine the weights for each parameter and the threshold. If a token N_i passes then it is considered a last_name, otherwise N_i is classified as a first_name. If a PN contains more than one last_name, the one with the highest score is chosen as the most representative one. For example, given the PN "Tommaso Padoa Schioppa", the Person Name

Splitter correctly classifies “Padoa” and “Schioppa” as last_names with “Padoa” as the most representative token.

In order to estimate *P1*, the prior probability of a token of being either a first_name or a last_name, we used the Web. First, we extracted from the corpus the names that collocate with known names (existing in the Name Dictionary), from which we obtained a list of some 15,000 unknown names. From a public webpage we extracted the list of the top twenty frequent first and last names. Then, we used the Google API interface to see how many times each unknown name appears with one of the twenty last_names and we computed the average frequency μ without considering the highest two values. We take this caution in order to reduce the risk of considering a name frequent just because it happens that a famous person carries it. Nevertheless, we also include the highest values if they are within a standard deviation interval from μ. The probabilities estimated over the Web are finally weighted according to a number of parameters obtained normalizing corpus data, including how many times a certain token occurs by itself, how many times it occurs as first and last name and how many times it occurs on the left and on the right of other names.

The parameter P2 is estimated as frequency ratio: the number of occurrences on the left, and on the right respectively, divided by the total number of occurrences.

The parameter *P3* encodes the intuition that, generally, last_names have a distinct form, which is predictable by observing both prefixes and suffixes carried by names. For example, in Italian, “-ni”, “-olli”, “-ucci”, “ellini” are typical last_names suffixes. We have derived a list of prefixes and suffixes which discriminate first_names from last_names comparing the last and first names included in the Name Dictionary. We have considered only those suffixes and prefixes which are discriminative in more than 90% of the cases and having more than 100 occurrences. The fact that *P2* is a reliable indicator can be seen for the fact that the great majority of suffixes and prefixes which score over 90%, scores also over 98%. For example, there are 255 suffixes respecting the above conditions out of which 154 are 100% discriminative and 203 score better than 98%.

The fourth parameter, *P4*, takes into account that generally, if a PN has a last_name, then usually it is the rightmost token. A more precise estimation on a corpus of 200 unambiguous PNs (two tokens each, both of them unambiguously either first or last name) is that the general rule is not obeyed in 7,5% of the cases, with a 95% confidence of being in the interval (7.5-5.15, 7.5+5.15) = (2.35, 12.65). Assuming that *P3* has a Poisson distribution with μ = 7.52, then we can determine whether an unknown name is a first or last name solely on its position, even if its companioning names are unknown. For example, the probability of a token to be a last_name, given that it appears only three times in a corpus, two times at the left of other names and one time by itself, is only 0.015 (1,5%).

4. The Local Coreference Module

This module addresses coreference among PNs within a single document. The input is made of all the PNs of the document, tagged by the Name Splitter as described in Section 3, while the output of the module is a set of clusters of PNs, where each cluster refers to a (document) different person. The local coreference algorithm we propose assumes the “one entity per document” hypothesis (OED). According to this hypothesis, within a document, the same PN always refers to the same person.

However, a major issue in establishing local coreferences is that a relevant proportion of PNs in a corpus are not complete, i.e. they are composed by either just the first_name (e.g.

"Silvio") or just the last_name (e.g. "Berlusconi"). We call incomplete PNs *free PNs*, and distinguish between *free_first_names* (the PN is only a first_name) and *free_last_names* (the PN is only a last_name). A free name may have a completion, and consequently corefered, coming from another free name or complete names. For example a free_last_name could be corefered with another free_last_name or with a last_name belonging to a complete_name. Initially all the last_names are possible candidates. The coreference is a probabilistic event - we calculate the most probable completion as explained below. The unknown_names are treated similarly. If they are corefered with a last_name then they are declared first_names, and vice-versa.

The completion is a probabilistic event, modeled by the following random variables: first name (*l*), last name (*L*), first_last name (*l_L*) and topic (*t*). Finding the most probable last_name for a free_first_name is finding the value of *L* which maximizes the following joint probability:

$$max_L p(l , L , l\_L , t) = max_L p(t)\, p(L \mid t)\, p(l \mid L, t)\, p(l\_L \mid l , L , t) \qquad (1)$$

where:

- *p(t)* is a constant, it does not count in computation.
- *p(L | t)*, the probability that a last_name is in that news document, shows the connection between a topic and a last_name and may be computed by using the Bayes theorem. As p(t) is a constant, p(L | t) = p(t | L) p(L). The probabilities on the right hand are computed as ratio and normalized.
- *p(l | L, t)* is the probability that a certain person is referred to by first_name in a news document. If nothing is known we consider it 1/2. Otherwise, given the corpus evidence, we compute a yes/no function, considering whether it is more probable for that person of being called by first_name than not, and consequently, a quantity is added/subtracted from 1/2.
- *p(l_L | l, L, t)* is the probability that a person's name is made out of first_name and a last_name. If there is a known link between l_L and t, like "very frequent", or "frequent", then we approximate the p(l_L | l, L, t) with the probability p(L | t). The "0" value means that we have unmatched abbreviations.

All the above probabilities have been calculated over the whole document collection. The reason to normalize each of the probabilities in (1) is twofold. Firstly, the probabilities on the right hand side can be very low and we are unable to correctly set a probabilistic space (the sum of marginal probabilities to be 1) Secondly, the number of occurrences of a collocation, such as last_name and topic, or first_name and last_name, is relevant once a certain threshold is passed. We cannot say that one person is more of a politician than another once we know that both are, nor can we say that a legal name is more legal than another.

The set of PNs that corefer represent a cluster. To each cluster we associate a name, which we call cluster name. This name represents the name of one person and all the PNs inside the respective cluster are (different) ways to refer to that name. A set of NEs is associated to this name and this information will be further used by the Global Coreference Module. The global coreference regards only the cluster names, not directly the PNs inside the clusters. The name of the cluster is the longest name we build out of different names inside the same cluster.

5. The Global Coreference Module

Given our probabilistic approach to coreference, the fundamental question addressed by the global coreference module is to estimate the probability of two PNs that have the same form to refer to different persons in the world. We are interested in estimating for any PN the probability that different persons share it (compare, for instance, the probability of a person being called by two extremely common names like "*Paolo Rossi*" vs. "*Roldano Not*" which very probably will identify uniquely a person).

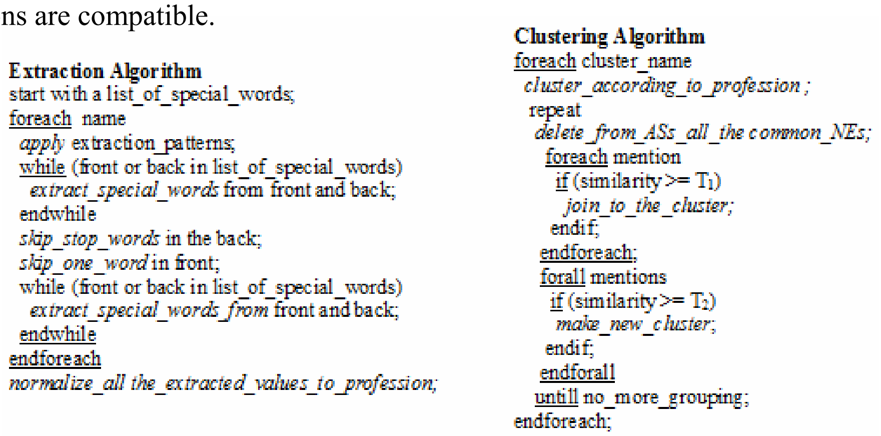
A first intuition is that the probability above is correlated to the frequency of the PNs: for a rare name it will be almost impossible that two different persons having that name, made it into the newspaper. According to this intuition we have divided the PNs in the corpus into five categories, and in Table 1 we show the figures and percentiles of the five categories computed on the Adige corpus.

Name Category	# First name	#Last Name
extremely uncommon	4 834 (7.21%)	14 319 (13.44%)
uncommon	6 294 (9.38%)	16 079 (15.09%)
common	52 167 (77.80%)	73 886(69.38%)
very common	2 634(3.92%)	1 485(1.39%)
extremely common	1 116 (1.66%)	717 (0.6%)
Total	**67 045**	**106 486**

Table 1. PN Categories clustered according to their frequency.

Since the probability of a person in the world carrying a "common" name to be mentioned in a newspaper is relatively low, we can assume that the probability of two different persons having the same "common name" is very small. In addition, even if they do appear in the newspaper, it is very probable that they are not related to the same entities, such as organizations and locations. Therefore, a common name appearing in two different news documents that have in common also at least one specific entity, may refer to the same person with a high probability.

The algorithm presented in Figure 2 is used in order to detect the names' determiners. Inside those determiners special words, denoting profession, are detected. After the local coreference level, the same name may have more than one profession. For example ("don", "monsegniore"). However, a part of these pairs are errors. The method we use for filtering out errors is motivated by the trigram distribution. If the number of occurrences of a trigram (name, profession_1, profession_2) is higher than the predicted one, then we assume that there is no mistake in identifying the profession, therefore the two professions are compatible.



```
Extraction Algorithm
start with a list_of_special_words;
foreach  name
  apply extraction_patterns;
  while (front or back in list_of_special_words)
    extract_special_words from front and back;
  endwhile
  skip_stop_words in the back;
  skip_one_word in front;
  while (front or back in list_of_special_words)
    extract_special_words_from front and back;
  endwhile
endforeach
normalize_all the_extracted_values_to_profession;
```

```
Clustering Algorithm
foreach cluster_name
  cluster_according_to_profession ;
   repeat
    delete_from_ASs_all_the common_NEs;
     foreach mention
      if (similarity >= T1)
       join_to_the_cluster;
      endif;
     endforeach;
     forall mentions
      if (similarity >= T2)
       make_new_cluster;
     endif;
     endforall
    untill no_more_grouping;
endforeach;
```

Fig 2. Extraction and Clustering Algorithm

The global coreference module starts corefering the mentions which have the same profession. These are seed clusters. In the second step the seed clusters are enriched by adding mentions which do not have any profession associated with them. The similarity between a mention and a seed cluster is computed by numbering the number of common NEs in the ASs. The mention is included in a seed cluster if a threshold T_1 is reached. The NEs which are common to two seed cluster are indiscriminative and therefore they are not counted. In this way a high purity of clusters is maintained. The mentions which are not similar to any seed cluster are clustered together if a second threshold, T_2, is reached. The process is repeated till no new coreferences are possible.

The similarity measure is computed according to the number of NEs in common between two mentions. Initially, all the NEs present in the same news enter in the ASs of a mention. After the first step of coreference – the creation of seed cluster – the NEs which are not discriminative are deleted. The similarity score takes into account the distance between the name and a NE. We consider three categories: sentence level, paragraph level and news level.

The thresholds, T_1 and T_2, are dynamically modified according to the name frequency class and the topic. According to the name frequency category, the T_1 and T_2 respectively are increased from "extremely uncommon" to "extremely common". The topic scores both positively and negatively. A third factor that contributes to the value of thresholds is the time period. The probability of two different persons carrying the same name being in the newspaper in a short period of time is extremely low and therefore the respective occurrences can be corefered.

The output of the Global Coreference Module is a set of persons which are added to the Entity Ontology.

6. Data and Evaluation

The coreference algorithm has been applied to the Adige corpus, a collection of 586,017 news documents (having 20,199,441 Name Entities mentions) which appeared in the Italian newspaper "L'Adige" over a period of seven years. The three modules of the system have been independently evaluated on the I- CAB benchmark.

I-CAB is a four days news corpus completely annotated and all the entity mentions manually coreferred. Table 2 shows the figures for the I-CAB relevant for our evaluation: the number of occurrences of Person Mentions (second column) with, within parentheses, the proportion of mentions containing a certain attribute; the number of names correctly recognized by the NER system (third column); the number of distinct names (fourth column); the number of distinct persons carrying them (fifth column). Additional details on the I-CAB corpus, relevant for estimating the difficulty of coreference task, are reported in (Popescu et al. 2006, Magnini et al. 2006b).

#attribute	#occ in ICAB	#correct NER	#distinct	#persons
FIRST_NAME	2299 (31%)	2283	676	1592
MIDDLE_NAME	110 (1%)	104	67	74
LAST_NAME	4173 (57%)	4157	1906	2191
NICKNAME	73 (1%)	54	44	41

Table 2. I-CAB: figures of interest for evaluating name coreference.

The coverage of the dictionary with respect to the document collection is 47.68%. Therefore, for more than half of the names we meet in the corpus we are unable to tell whether they are first or last names.

We use the “all for one” baseline: all equal PNs stand for the same person. Partial_names are completed with the most frequent complete name that includes them (for instance, the free_first_name “Silvio” is completed with “Silvio Berlusconi”). The rightmost token in a PN is considered the last name. In Table 3 we present the performances of our system, comparing the error rate of the system and of the baseline.

Module	#System errors	#Baseline errors
Name Category Recognizer	151 (2.65%)	603 (10.97%)
Local Coreference	11 (0.4%)	29 (0.8%)
Global Coreference	31 (1.46%)	150 (5.49%)

Table 3. System and Baseline error rate.

The evaluation on I-CAB exhibits a very strong baseline. This is not what is expected to happen at the corpus level. The figures in Table 3 of a coreference algorithm only on I-CAB may be statistically too weak. We plan to build an evaluation corpus that is built on statistical bases, instead of covering a continuous period of time. From each name frequency category we select a number of names according to the percentage of corpus coverage of that name frequency category.

7. Conclusion and Further Research

We have presented a probabilistic framework for Person Name Coreference. The system is divided in three modules: first, the last_name of a person name is recognized, then the coreference among a single document is established and finally the coreference at the level of the whole collection is addressed. Each module has been evaluated against a corpus of manually annotated documents.

Our analysis suggests that while generally good results can be achieved, there is still a long way to go for perfection. The principal bottleneck is, in our opinion, the fact that names behave randomly and many times there is no explicit evidence in the corpus for their coreference. There are open issues in the Name Coreference tasks. The exact coreference is guaranteed only by ontological facts, but we are unable to extract this information out of news documents. We approximate uniqueness by computing the probability of occurrence considering that it is a rare event that different persons have the same names and/or share partially the same set of NEs. We improved the performances by using the determiners inside the NP of name mentions.

References

Artiles J., Gonzalo J., Sekine S. (2007). Establishing a benchmark for the Web People Search Task: The Semeval 2007 WePS Track. In *Proceedings of Semeval 2007*, Association for Computational Linguistics.

Bagga A. and Baldwin B. (1998). Entity-based cross-document co-referencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics*, 75-85.

Magnini B., Pianta E., Girardi C., Negri M., Romano L., Speranza M., Bartalesi Lenzi V., Sprugnoli R. (2006). I-CAB: the Italian Content Annotation Bank. In *Proceedings of LREC-2006*, Genova, Italy.

Pedersen T., Purandare A., Kulkarni A. (2006). Name Discrimination by Clustering Similar Contexts. In *Proceedings of the World Wide Web Conference*.

Popescu O., Magnini B., Pianta E., Serafini L., Speranza M., Tamilin A. (2006). From Mentions to Ontology: A Pilot Study. In *Proceedings SWAP06*, Pisa, Italy.

Zanoli R., Pianta E. *SVM based NER (2006)*. Technical Report, Trento, Italy.

Magnini B., Pianta E., Popescu O. and Speranza M. (2006). Ontology Population from Textual Mentions: Task Definition and Benchmark. In *Proceedings of the OLP2 workshop on Ontology Population and Learning*, Sydney, Australia, 2006. Joint with ACL/Coling.