IdentityRank: Named Entity Disambiguation in the Context of the NEWS Project

Norberto Fernández[1], José M. Blázquez[1], Luis Sánchez[1], and Ansgar Bernardi[2]

[1] Carlos III University of Madrid, Leganés, Madrid, Spain
{berto,jmb,luiss}@it.uc3m.es
[2] German Research Center for Artificial Intelligence, DFKI GmbH, Kaiserslautern, Germany
ansgar.bernardi@dfki.de

Abstract. In this paper we introduce the IdentityRank algorithm, developed as part of the EU-funded project NEWS to address the problem of named entity disambiguation in the context of semantic annotation of news items. The algorithm provides a ranking of the candidate instances within an ontology which can be associated to a certain entity. In order to do so, it uses as context the metadata available in a certain news item. The algorithm has been evaluated with promising results.

1 Introduction

The EU-IST funded project NEWS[1] (News Engine Web Services) [3], which has recently been completed, aimed at providing solutions which help news agencies to overcome limitations in their current workflows and increase their productiveness and revenues by using a Web Service based architecture and Semantic Web technologies.

In order to apply Semantic Web technologies to the news domain, in the NEWS project a set of components were developed. One of them is the NEWS ontology [4], a lightweight RDFS[2] ontology providing a formal model of the domain. Another one is an annotation component, developed by Ontology Ltd., which uses natural language processing techniques to provide capabilities such as categorization and named entity extraction.

Within the semantic annotation process, one of the key problems that we found in NEWS was the disambiguation of the entities detected by the natural language processing engine. This engine extracts named entities out of the news items, but, in order to allow a fine-grained semantic search for the user of the NEWS system, these entities have to be matched against instances of the NEWS ontology. That is, the natural language processing engine can detect that a certain occurrence of the piece of text *Bush* represents a person, but we also need to deduce that this person is represented in the NEWS ontology by a certain URI like *http://www.news-project.com/2005/1*.

[1] Contract number: FP6-001906. Web site: http://www.news-project.com
[2] http://www.w3.org/TR/rdf-schema/

E. Franconi, M. Kifer, and W. May (Eds.): ESWC 2007, LNCS 4519, pp. 640–654, 2007.

In this paper we describe the IdentityRank algorithm (a.k.a. IdRank) that we designed in order to address the entity disambiguation problem in the news domain. Our algorithm, inspired by PageRank [10], exploits the metadata currently provided by news agencies (like news item timestamp) and the information provided by the natural language processing engine (categories and entities) as a context for named entity disambiguation. Using all this information, IdRank allows to match news items' entities to ontology instances automatically.

The rest of this paper is organized as follows: section 2 describes with more detail the IdRank operational scenario within the NEWS workflow. Section 3 describes the algorithm. Section 4 shows the results of an experimental evaluation of the algorithm. Section 5 takes a deeper look at related work, and finally, section 6 gives concluding remarks and finalizes the paper.

2 Scenario

In order to give a clearer idea of the operational environment of IdRank, we describe in this section the NEWS workflow, which acts as an scenario for the entity disambiguation problem. The NEWS workflow design has taken into account that the journalists in the news agencies want to have control over all the content production process in order to ensure the quality of the results. This leads to a supervised solution, where the journalist can validate the results obtained in the different processing stages of a news items. These are:

1. The journalist creates a news item using the NEWS GUI. The news item is represented in XML and some metadata like author and timestamp are added to it.
2. The news item is processed by the natural language processing component. It annotates the news item with some entities and categories. The vocabulary used for categorization is taken from the NEWS ontology. Basically this vocabulary is an RDFS representation of the International Press Telecommunication (IPTC) standard Subject Codes NewsCodes[3]. These Subject Codes constitute a three level taxonomy that, at the moment, contains about 1300 different categories. In such taxonomy, each category is identified by a fixed eight decimal-digit string. The first two digits represent the first level of the taxonomy, which consists of 17 different categories. For instance, the Subject Code 01000000 represents the category *arts, culture and entertainment*, the Subject Code 01011000 represents the subcategory *music* and the Subject Code 01011006 represents the subsubcategory of news items talking about *rock music*.

   With respect to entities, these are also added to the news item. For each entity the natural language processing engine provides the tagged text and

[3] http://www.iptc.org/NewsCodes

the entity type, which in our case is one of the three possibilities: person, place or organization. For instance the following piece of XML:

```
<meta content="11000000" name="srs-category" />
<meta content="Gargano" name="entity-person" />
<meta content="Mexico" name="entity-places" />
<meta content="Liberal Party" name="entity-organization" />
```

would be added by the natural language processing engine to state that the news item belongs to category 11000000 *politics* and mentions the entities *Gargano*, a person, *Mexico*, a place, and *Liberal Party*, an organization.

3. The annotated document is sent back to the GUI and the journalist is allowed to check the annotations. The validated document is sent to other of the NEWS components: the Heuristic and Deductive Database (HDDB).
4. The HDDB stores the news item, indexes its textual content to allow keyword based search, stores the news item metadata, including the categories and entities, and then runs IdRank to disambiguate the entities to instances in the NEWS ontology.
5. The results of IdRank, a set of assignments (entity, instance), are then shown to the journalist. (S)he may confirm them, select a different instance for some entity (creating a new one if needed) or might simply drop the assignment and leave the entity without associated instance.
6. The results of the validation process are sent back to the HDDB, where are stored and used to train IdRank. All the information generated and stored in this process and the NEWS ontology are used by the HDDB to allow intelligent content distribution services.

3 The Algorithm

As we have seen in the previous section, the NEWS natural language processing engine is able to extract basic entities from text. But in order to allow fine-grained semantic search over the news item repository stored in the HDDB it is not enough to figure out, that the extracted text string *Alonso* represents a person, we need to know who is that person by mapping the entity to an instance in the NEWS ontology. For instance, for the entity (*Alonso*,*person*) there are the following candidates in the NEWS ontology:

```
Fernando Alonso, Airbus flight testing vice-president.
http://www.news-project.com/2005/11
Fernando Alonso, Formula 1 driver.
http://www.news-project.com/2005/12
Mikel Alonso, soccer player.
http://www.news-project.com/2005/13
Xabi Alonso, soccer player.
http://www.news-project.com/2005/14
Jose Antonio Alonso Suarez, Spanish politician.
http://www.news-project.com/2005/15
Alonso Cano, Spanish painter, architect and sculptor.
http://www.news-project.com/2005/16
Alonso de Ercilla y Zuniga, Spanish poet.
http://www.news-project.com/2005/17
```

So a problem of ambiguity arises: which is the best candidate instance to be assigned to a certain entity? Finding that instance is the main task of the IdRank algorithm, which is based on two principles:

Semantic coherence: Instances typically occur in news items of certain categories, e.g., the politician *Jose Antonio Alonso* in news items of *politics* category. Also the occurrence of a certain instance gives information about the occurrence of other instances. For example, the soccer player *Xabi Alonso* usually appears in news items in which the soccer team where he plays, *Liverpool*, is also mentioned.

News trends: Important events typically are described with several news items covering a certain period of time. For instance when the Formula 1 driver *Fernando Alonso* won the F1 world championship, several news items describing such event where composed, most of them including instances as *Fernando Alonso* and *Renault*, his F1 team.

In this section we will describe in detail the main processes involved in the IdRank algorithm. As we have said, IdRank is partially inspired by PageRank, so we will start by briefly describing PageRank before going into the IdRank details.

3.1 PageRank and Relation with IdRank

The PageRank algorithm [10] exploits the information in web links to compute the ranking of a certain web page. The basic idea is mentioned in [10]: *a page has high rank if the sum of the ranks of its backlinks is high.* So in PageRank the ranking or importance of a certain page depends on the ranking and number of the pages which point to it (backlinks). Mathematically this is represented by the following equation (see [10]):

$$R(u) = \lambda \sum_{v \epsilon B_u} \frac{R(v)}{N_v} + \lambda E(u) \tag{1}$$

Where:

- λ is a factor used for normalization.
- $R(u)$ represents the ranking of the web resource u. The L_1 norm of the vector R, composed of all $R(u)$, is such that $||R||_1 = 1$.
- B_u is the set of backlinks of u.
- N_v is the cardinality of F_v, the set of pages v points to (forward links of v).
- E is a vector that corresponds to a source of rank. As is indicated in [10], each component $E(u)$ can be used to adjust the rank of a certain resource u, for instance for personalization purposes (give more weight to certain pages).

This equation can be represented in a matricial manner:

$$R = \lambda AR + \lambda E \tag{2}$$

Where A is a matrix, $A_{uv} = 1/N_v$ if $v \epsilon B_u$ or 0 otherwise, $||A||_1 = 1$.

The relation between PageRank and IdRank comes from the application of one of the basic principles that inspire IdRank. The Semantic Coherence principle states that the appearance of an instance gives certain information about the occurrence of other instances. Paraphrasing the sentence in [10] we can say that: *an instance has high rank if the sum of the ranks in the news item of the instances that typically cooccur with it is high.* As PageRank does with web pages, the objective of IdRank is to obtain a ranking: the ranking of the possible *identities* (candidate instances) of a certain entity.

The next subsection will describe with more detail how IdRank works in practice.

3.2 IdRank

Three are the main steps needed to run IdRank on a certain news item: *finding* the candidates instances in the ontology for each entity in the news item, *ranking* that candidate instances using a modified version of PageRank and *retraining* the algorithm with the journalist feedback once the process is finished. The next subsections will describe each of these steps in more detail.

Find candidate instances. This process takes as input the entities detected by the natural language processing engine in a certain news item and produces as output a set of candidate instances for each of the input entities. For instance, given the entity (*Alonso,person*) the following steps are executed:

1. Given the entity type, the HDDB code is configured to decide which is the upper class in the NEWS ontology taxonomy which maps to the entity type. In our example the mapping is as follows: the entity type *person* maps to the ontology class *Human*. The other possible mappings are: the entity type *place* maps to the ontology class *Location* and the entity type *organization* maps to the ontology class of the same name.
2. The HDDB computes the transitive closure of the *subclassOf* property to find all the subclasses of the class of interest. For instance, in our example, the deductive part of the HDDB computes the transitive closure of the class *Human* finding the two subclasses of this class: *Man* and *Woman*.
3. An SQL query is automatically generated to query the database where the NEWS ontology A-box is stored. With this query we find the candidate instances that match the entity text and belong to the classes *Human, Man* or *Woman*. For instance, in the example introduced above, the SQL looks like:

```
SELECT DISTINCT(uri) from Instances
WHERE (
        label LIKE '% Alonso' OR label LIKE 'Alonso %' OR
        label LIKE '% Alonso %' OR label = 'Alonso'
)
AND (
        type IN (
                'http://www.news-project.com/Ontology/Content#Human',
                'http://www.news-project.com/Ontology/Content#Man',
                'http://www.news-project.com/Ontology/Content#Woman'
            )
);
```

The same process is repeated with all the entities detected in the news item by the natural language processing engine.

Rank the candidates using a modified version of PageRank. Once the candidate instances of all the entities are obtained, a semantic network with all these instances is defined. In such semantic network the nodes represent the different candidate instances for the entities in the news item. If an instance is candidate for more than one entity, it only appears once. The arcs between two nodes appear when the two instances have cooccurred in the past in at least one news item, that is, if at least one news item exists in the HDDB that is annotated with occurrences of both instances.

Then we apply a modified version of PageRank to the semantic network. In our algorithm, instead of dividing the importance of an instance among its forward links evenly, as PageRank does with the quotient $R(v)/N_v$ in equation (1), we will give weights to the links. That is, in IdRank, the occurrence of an instance can give more weight to certain instances than to others. These weights depend on the cooccurrence frequency of the involved instances.

Mathematically we have the following set of equations:

$$R(I_i) = \lambda \sum_{j \epsilon C_i} \alpha_{ij} R(I_j) \tag{3}$$

Where:

- λ is a factor used for normalization.
- $R(I_i)$ represents the ranking of the candidate instance I_i in the context of the news item.
- C_i is the set of candidate instances in the semantic network that cooccur with I_i in at least one news item apart from the one being analyzed.
- α_{ij} represent the weight of the link from I_j to I_i, that is, the proportional part of the I_j importance or ranking which is given to I_i. Mathematically, this can be expressed as:

$$\alpha_{ij} = \frac{f_{ij}}{\sum_{k \epsilon C_j} f_{kj}} \tag{4}$$

Where f_{ij} is the coocurrence frequency of I_i and I_j, that is, the number of news items where both I_i and I_j occur divided by the number of news items where I_j occurs. With this definition: $\alpha_{ij} \epsilon [0, 1]$ and:

$$\sum_{\forall i \epsilon C_j} \alpha_{ij} = 1 \tag{5}$$

Note that, as has been previously indicated, the weights α_{ij} and α_{ji} are, in general, not equal due to equation (4).

At this point, we have described how the coocurrence of instances is used by the algorithm, but still two contributions remain unclear: the *Semantic Coherence* principle is also dependent on the news item categories, and the *News Trends* principle, which uses the timestamp, has also not been exploited.

In order to exploit also that information, we will use the E component included in the original PageRank formula (equation (2)) where it was used to personalize the ranking. In our case, we are going to use it with a similar meaning, *personalizing* the ranking computation for the context of the concrete news item. Mathematically we will have now:

$$R(I_i) = \lambda \sum_{j \epsilon C_i} \alpha_{ij} R(I_j) + \lambda E(I_i) \tag{6}$$

In practice, the vector E, composed of all the $E(I_i)$, is computed as a normalized sum of contributions:

$$E = \sum_{\forall c} E_{cnorm} = \sum_{\forall c} \frac{E_c}{||E_c||_1} \tag{7}$$

At the moment, the set of contributions which are being considered in the context of NEWS are:

E_{tim}**: instance occurrence in last D days.** The value of each element $E_{tim}(I_i)$ of E_{tim} is computed taking into account the frequency of occurrence of the candidate instance I_i in the news items of the last D days, taking as time origin the timestamp of the news item being analyzed. D is a constant empirically determined (we worked with D=7).

E_{cat}**: instance occurrence in news items of certain category.** Takes into account the occurrence of the instance in news items belonging to a certain top level category (01000000-17000000). A news item belongs to a certain category if the annotation engine assigns it that category or one of its subcategories. As a news item can belong to several different top level categories, in practice E_{cat} is composed of the sum of several components.

For each top level category, *tlc*, in the news item, the value of each element of the vector $E^{tlc}_{cat}(I_i)$ is computed taking into account the frequency of occurrence of the candidate instance I_i in news items belonging to *tlc*. The final vector E_{cat} is just a linear combination of the different vectors E^{tlc}_{cat}. Less frequent categories have a higher weight in that linear combination, because they provide more information about the news item.

Taking into account these contributions and equation (7), we can represent the equation (6) in a matricial manner, as in 3.1:

$$R = \lambda AR + \lambda E = \lambda AR + \lambda E_{catnorm} + \lambda E_{timnorm} \tag{8}$$

Where, in the same way as in [10], A is a matrix, $A \in M_{nxn}$, $A_{ij} = \alpha_{ij}$ and $R, E_{catnorm}$, $E_{timnorm}$ are vectors, R, $E_{catnorm}$, $E_{timnorm} \in R^n$ and n is the total number of different candidate instances in the news item.

Due to equation (5) $||A||_1 = 1$ and due to equation (7) $||E_{catnorm}||_1 = 1$ and $||E_{timnorm}||_1 = 1$. The consequence of this fact is that if we use directly the equation (8), we give the same weight in the computation of the R vector to the components depending on instance coocurrence, A, depending on instance-category coocurrence, $E_{catnorm}$ and depending on the temporal information, $E_{timnorm}$. In order to control the effect in the final ranking of each contribution, we have assigned weights to each component, resulting the equation:

$$R = \lambda(k_a AR + k_{cat} E_{catnorm} + k_{tim} E_{timnorm}) \tag{9}$$

Where $k_a + k_{cat} + k_{tim} = 1$. As is indicated in [10], since $||R||_1 = 1$ the equation (9) can be rewritten as:

$$R = \lambda(k_a A + (k_{cat} E_{catnorm} * \mathbf{1}) + (k_{tim} E_{timnorm} * \mathbf{1}))R \tag{10}$$

Where $\mathbf{1}$ represents a row vector of all ones, $\mathbf{1} \in R^n$, and * represents the matrix product.

Analyzing equation (10) we conclude that, as happens in the original PageRank algorithm, we can compute the vector R simply by determining the main eigenvector of a matrix. In our case the matrix is: $k_a A + (k_{cat} E_{catnorm} * \mathbf{1}) + (k_{tim} E_{timnorm} * \mathbf{1}) \in M_{nxn}$. In our implementation, that eigenvector is computed using a numerical method: the power method.

Once R is computed, we know the ranking of each candidate instance in the context of the news item being analyzed: the weight of the instance I_i is simply the component i of the vector R. For each entity in the news item, the algorithm returns a vector with all the pairs (candidate instance, weight) for such entity. This vector is sorted using the weight, so the candidate with the biggest weight is the one shown to the journalist as best candidate instance for the entity. If more than one candidate has the biggest weight, the algorithm randomly selects one of them as the first one.

Retraining the system. The results of the ranking process are shown to the journalist at the GUI. The journalist can check the suggestions of the system and correct the wrong ones. The resulting annotations are stored into the HDDB and used to retrain IdRank. Basically the retraining process consist in storing or updating into the relational database used by the algorithm information needed for the algorithm process. Concretely, for each instance in the news item we update the following information: the occurrence of the instance in a certain timestamp (used in computing E_{tim}), the counter of number of coocurrences between the instance and all the other instances detected in the new news item (used in computing A) and the counter of the number of occurrences of the instance in each of the top level categories of the news item (used in E_{cat}).

For each new news item, we store or update also the following information: the counter of the total number of news items, the counter of the number of news items belonging to a certain top level category (used in computing E_{cat}) and the association between news item and its timestamp (used in E_{tim}).

As IdRank reads on the fly from the database the information needed to perform its computations, the next time the algorithm is run, the new training information is taken into account.

4 Evaluation

The first step in the empirical evaluation of the algorithm was to define a theoretical threshold that we can take as reference to compare our algorithm. In our case, that theoretical threshold is provided by the average accuracy of a naive disambiguation algorithm that simply assigns randomly one of the possible candidate instances to each entity. We assume that all the candidates of a certain entity have the same probability of being chosen as the right one. We also assume that the decisions of that hypothetical algorithm are independent, that is, that it chooses the candidate instance for a certain entity independently of all the other elections in the corpus. Finally, we also assume that, for each entity in the corpus, there exists in the ontology at least one candidate: the right candidate instance to be mapped. Though this last assumption seems unrealistic, in practice in the NEWS scenario, as journalists are allowed to insert new instances into the knowledge base, entities without the right instance can get one as soon as they are detected.

With these assumptions, we get the following expression for the accuracy:

$$Av[Acc] = \frac{Av[right]}{total} = \frac{\sum_{\forall e} Occ(e)P(Right/e)}{N_{ent}} \tag{11}$$

That is, the average accuracy is defined as the average number of right assignments entity/instance of our naive algorithm divided by the total number of possible assignments. The total number of assignments coincides with the number of entities in the corpus (N_{ent}) due to the assumption that each entity has at least one candidate. The total average number of right decisions is the addition of the average number of right decisions for each entity e in the corpus. Due to the assumption of independent election, the average number of right decisions for a certain entity e can be computed as the number of occurrences of the entity e in the corpus $Occ(e)$ multiplied by the probability of making a right decision on that entity $P(Right/e)$. Due to the assumption of random uniform election between the candidates for a certain entity, $P(Right/e) = 1/Ncand_e$, where $Ncand_e$ represents the number of candidates for the entity e in the ontology. As N_{ent} is constant for the summatory in the fraction, we can reformulate the equation (11) as:

$$Av[Acc] = \sum_{\forall e} \frac{Occ(e)}{N_{ent}} \frac{1}{Ncand_e} \tag{12}$$

As can be seen, and not surprisingly, the final accuracy depends on the concrete corpus used for evaluation ($Occ(e)/N_{ent}$ component) and the ontology used as source of candidates instances in the evaluation ($1/Ncand_e$ component). This

value of the average accuracy of the naive random election algorithm can also be interpreted as a measure of the degree of ambiguity of the pair corpus/ontology used in the evaluation: the bigger $Av[Acc]$ the lower the ambiguity.

Once the theoretical baseline that we use as reference for our algorithm is defined, we will describe in next subsections the concrete results of the empirical evaluation of the algorithm. We start by describing the corpus and ontology used in the evaluation process, after that, we describe the results of the accuracy evaluation and finally we include some measurements of the computation time of the algorithm.

4.1 Corpus and Ontology

As we have seen previously, the corpus and the ontology selected to perform the evaluation have direct influence on the results of the evaluation. Due to this, we have carried out our evaluation using two different corpora and two different ontologies, instead of only one.

Corpora. As we are evaluating an algorithm for ambiguity resolution, we are interested in having ambiguity in our corpora. In order to achieve this, the process of building our corpora started by selecting a possible ambiguous entity and querying the NEWS repository, which contains real news items of Spanish news agency EFE and Italian news agency ANSA, for news items where such entity appears. More in detail two entities were selected for the process, *Georgia, location* and *Alonso, person*. The query gave us 32 news items for the entity *Georgia, location* and 65 for the entity *Alonso, person*. All the entities appearing in the news items, where manually disambiguated using the NEWS ontology. The annotations were reviewed by two different persons to ensure as much as possible the quality of the evaluation corpora.

The results were 343 total entities in the Georgia corpus, 169 of them distinct (different pair entity text, entity type), and 742 entities in the corpus of Alonso, 229 of them distinct. The entity *Georgia, location* appeared with two different meanings in the Georgia corpus (Georgia as U.S. state -12 times- and Georgia as country -20 times-) and the entity *Alonso, person* appeared with three different meanings in the corpus of Alonso: Fernando Alonso, a Formula 1 driver (41 times), Jose Antonio Alonso a Spanish minister (23 times), and Xabi Alonso, a soccer player (only once).

The timestamps of the Alonso news items range from the 13/Oct/2005 to the 12/May/2006 and the ones of the Georgia corpus range from 17/Oct/2005 to 12/May/2006. Their distribution is shown in table 1. The distribution of the number of news items belonging to the top level categories (01000000 to 17000000) in both the Georgia and Alonso corpora is shown in table 2. Note that the total number of categories in each corpus is bigger than the total number of news items in such corpus, because a single news item can be categorized into different categories.

Ontologies. We have also used two different ontologies for our process. One of the ontologies was the NEWS ontology, the other one was built using the

information on the annotated corpus. In order to build this second ontology, we considered as the only possible candidates for each different entity the ones that appear in the manually annotated corpus. For instance, as we have seen the entity *Alonso,person* has 7 different candidates in the NEWS ontology, but in the corpus it appears only with 3 different meanings, so the number of candidates for this entity is 7 in the NEWS ontology and 3 in the ontology built considering only the candidates that appear in the corpus.

Table 1. Number of news items in each month from Oct 2005 to May 2006

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Alonso	28	0	0	0	2	10	1	24
Georgia	27	0	0	0	0	2	0	3

Table 2. Number of news items in each of the top level categories

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
Alonso	1	9	0	1	0	0	0	2	0	0	23	0	0	1	40	4	0
Georgia	2	7	0	3	0	0	2	3	0	0	16	0	0	0	3	4	1

4.2 Accuracy Evaluation

In our accuracy evaluation process, we were interested in measuring the effect of using different ontologies and corpora. Another aspect of interest was to measure the impact on the final results of the three components involved in the IdRank computation, instance coocurrences, categories and timestamp. Due to this, for each of the four possible combinations (ontology,corpus) we ran four experiments changing the values of the k_a, k_{cat} and k_{tim} parameters. So the total number of experiments performed was 16. As IdRank has a random component, each of these experiments was run 10 times and the average accuracy of IdRank was measured. We centered our attention on two aspects of the accuracy. One was the global accuracy, measured as total number of right assignments entity/instance divided by the total number of assignments. The other one was the relative accuracy for the entity used to construct the corpus, defined as the number of right assignments on the decisions of that entity divided by the number of decisions about the entity.

The results of this evaluation are shown in table 3. The first column, labeled as *Corp, Ont, Res* indicates the corpus (A, Alonso or G, Georgia) the ontology (N, NEWS ontology or C, Corpus dependent ontology) and the results (Tot, total accuracy or A/G, Alonso/Georgia, relative accuracy). So, for instance, the value *A,C,A* indicates that these results where obtained with the corpus of Alonso, using the ontology built taking as input such corpus and that only the relative accuracy for the entity *Alonso, person* is shown. The second column shows the theoretical results. These are computed with the equation 12 for the total accuracy case. For the relative accuracy case, the theoretical average accuracy is just

the total number of occurrences of the entity, multiplied by the probability of choosing the right candidate for the entity and divided by the total number of occurrences of the entity, that is, just the probability of choosing the right candidate for the entity. As we have seen this depends on the concrete ontology, so for instance, the second column in the row *A,N,A* is 1/7*100 = 14.3%, because in the NEWS ontology, the one used in such experiments, the entity *Alonso, person* has 7 candidates. The rest of the columns of the table show the results of concrete experiences with the pair (ontology,corpus). The column A shows the results obtained when $k_a = 1$, the column E_{tim} the results when $k_{tim} = 1$, the column E_{cat} the results when $k_{cat} = 1$ and finally the column, *All* contains the results obtained when the three components of the algorithm were considered. In concrete we used: $k_a = 0.8$, $k_{tim} = 0.05$ and $k_{cat} = 0.15$. For each of the entries in the table, estimated by averaging the results of 10 executions of IdRank, the mean and the standard deviation are shown.

Table 3. Average accuracy results (percentages)

Corp, Ont, Res	Theo. (%)	A (%)	E_{tim} (%)	E_{cat} (%)	*All* (%)
A,N,Tot	82.89	96.44 (0.62)	95.21 (0.52)	96.27 (0.58)	96.48 (0.52)
A,N,A	14.3	93.69 (1.35)	74.62 (1.81)	93.23 (0.79)	95.54 (0.49)
A,C,Tot	92.07	97.91 (0.25)	96.35 (0.33)	97.78 (0.25)	98.07 (0.23)
A,C,A	33.3	95.38 (0.73)	74.46 (2.63)	93.23 (1.30)	95.73 (0.68)
G,N,Tot	88.56	97.32 (0.55)	93.67 (0.65)	93.09 (0.55)	96.24 (0.57)
G,N,G	33.3	93.13 (1.98)	57.81 (8.10)	54.69 (3.68)	85.00 (1.32)
G,C,Tot	95.04	98.89 (0.18)	95.92 (0.55)	95.66 (0.47)	98.22 (0.26)
G,C,G	50	94.06 (1.77)	61.25 (6.28)	57.50 (4.70)	85.62 (1.61)

Analyzing the results in table 3, we see that the A component, related with instances coocurrence, is more accurate in giving us the right candidate instance than the E_{tim}, E_{cat} components. The category-related component, works fine in the case of the entity *Alonso, person*, because most of the news items in category 15000000 (sports) talk about Fernando Alonso, the Formula 1 driver, whereas the news item in category 11000000 (politics) are mostly related with Jose Antonio Alonso, a Spanish minister. Nevertheless, the behavior of the category-based component, is worse in the case of the entity *Georgia, location*. This is due to the fact that locations usually are not directly related with a certain subject, so we can have news talking about very different events, and thus having completely different categories, mentioning the same location. In fact, due to the bad performance of the category-based component in the Georgia case, the results of the *All* test are worse than the ones obtained by using only the instance coocurrences information. With respect to the temporal component, its poor results can be explained by the fact that in our concrete corpora the occurrences of the different candidates are interleaved, and the temporal window is relatively long (D=7) to give good results. But, as the number of news items in the corpora

is low and they are relatively disperse, we had to use long windows to get significative results for this component. So, more experiences with bigger corpora and with different values of the temporal window D must be accomplished to extract definitive conclusions.

4.3 Computation Time

As we have said in the initial sections, IdRank is designed to operate on the real production environment of a news agency. In such environment the time expended on creating and sending to the customers a new news item should be minimized, because the news agencies are interested in providing the relevant news items to the clients as soon as possible. In order to evaluate whether IdRank is adequate to operate on such an environment, we have conducted an evaluation of the time expended by the algorithm.

This evaluation consisted in running the algorithm 10 times with the Alonso corpus and the NEWS ontology, the case with bigger ambiguity, and compute the average time expended by the algorithm in each of its subprocess: candidate finding, ranking and retraining. The parameters of the evaluation were: $k_a = 0.80$, $k_{tim} = 0.05$. $k_{cat} = 0.15$. The tolerance and maximum number of iterations of the iterative method used to compute the matrix eigenvector where 0.0001 and 100 respectively. We conducted this experiment on a machine with Linux Debian 3.1 operative system, kernel 2.6.11, one Gigabyte of RAM memory and a Pentium(R) Mobile 1.60GHz processor.

Table 4. Average Computation Time

Nent	Ncat	Av[Find] (msec)	Av[Rank] (msec)	Av[Retrain] (msec)	Av[Total] (msec)
25	1	123.5	2705.7	3348.2	6177.4
26	1	270.4	1594.4	2015.6	3880.4
23	1	114.2	2246.3	2663.5	5024.0
23	1	197.1	2719.7	2347.6	5264.4
30	1	392.9	570.4	4868.4	5831.7

Table 4 shows the results for the five news items with worst total average execution times. For each news item, the number of distinct entities N_{ent}, the number of categories N_{cat} and the average time of finding, ranking and retraining, are shown. The last column shows the average total time needed by IdRank to process the news item.

As can be seen, the total time is in the order of seconds, which seems affordable for the proposed application scenario. Another conclusion is that the retraining time has a significative influence on the final results. On the positive side, we have to say that the retraining process does not have much effect on the time perceived by the journalist and the news agency client, because the retraining process is done when the edition process of the news item is finished.

5 Related Work

Named entity disambiguation or proper name disambiguation is a type of *word sense disambiguation* [8], in which the words to be disambiguated are named entities. There are lots of approaches in the state of the art dealing with word sense disambiguation and also with named entity disambiguation. These different approaches can be characterized according to a number of criteria:

- The *context* used to disambiguate the entity. Some approaches use the complete document where the entity is placed to disambiguate [2]. Others use as context a number of words before and after the entity. Those can be further classified on those that take a "bag of words" context (the position of the words taken as context is not considered) like [11] and those that try to use the role of each word in the context and their relation with the entity [9].

  Although some approaches use both common words and named entities as context [11], others suggest that better results can be obtained using as context only other named entities [9].
- The use of *knowledge sources* like lexical databases, etc., that define the instances that should be matched against the entities and can provide information that can be exploited to perform the matchings. There are of course several approaches that make use of such knowledge sources [1,7]. However, a remarkable number of approaches try to cluster the named entities without any reference to an available list of possible instances [11,9].
- The *disambiguation algorithms* employed can make use of a number of techniques or a combination of them: statistical procedures [6,11,9], morphosyntactic analysis [9,2], or exploiting ontologies that provide rich linguistic and semantic information about instances of interest [7].
- The *domain*: several approaches are oriented to a particular domain like biology [5] or bibliographic citations [1,6].

The usage of a semantic network ranking algorithm, which also takes into account the temporal component and the categorization system characteristic of the news domain, are the main differences of our approach compared with the ones in the state of the art.

6 Conclusions and Future Lines

In this paper we introduce the IdRank algorithm to address the problem of entity disambiguation in the context of semantic annotation of news items. The algorithm provides a ranking of the candidate instances within an ontology which can be associated to a certain entity. In order to do so the algorithm uses as context the metadata available in a certain news item. It is based on the principles of *Semantic Coherence* (instances typically occur in similar contexts) and *News Trends* (it is common to have temporal burst of news items talking about a certain event).

We have performed an empirical evaluation of the algorithm that shows its adequacy for the news domain, both for the quality of results and the computation time.

A possible future line of development that we want to explore is the possibility of using dynamic coefficients k_a, k_{cat} and k_{tim}, instead of the constant ones. In the training process we would decide the right coefficients for the next execution, depending on the quality of results obtained in the past ones. Evaluating the algorithm in bigger corpora and adapting it to other domains are also future lines of development.

References

1. N. Aswani, K. Bontcheva, H. Cunnigham. Mining Information for Instance Unification. In 5th International Semantic Web Conference. Ed. Springer, LNCS 4273, pp. 329-342. Athens, USA. November 2006.
2. A. Bagga, B. Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In 17th International Conference on Computational Linguistics. Quebec, Canada. August 1998.
3. N. Fernández, J. M. Blázquez and J. Arias, L. Sánchez, M. Sintek, A. Bernardi, M. Fuentes, A. Marrara, Z. Ben-Asher. NEWS: Bringing Semantic Web Technologies into News Agencies. In 5th International Semantic Web Conference. Ed. Springer, LNCS 4273, pp. 778-791. Athens, USA. November 2006.
4. N. Fernández, L. Sánchez, J. M. Blázquez, J. Villamor. The NEWS Ontology for Professional Journalism Applications. A Handbook of Principles, Concepts and Applications in Information Systems. Ed. Springer, Integrated Series in Information Systems, Vol. 14. To appear in December 2006.
5. F. Ginter, J. Boberg, J.Ärvinen, T. Salakoski, New Techniques for Disambiguation in Natural Language and their Applications to Biological Text. Journal of Machine Learning Research, 5: 605-621, 2004.
6. H. Han, L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. In Joint ACM/IEEE Conference on Digital Libraries. Tucson, USA. June 2004.
7. J. Hassell, B. Aleman-Meza, I. Budak Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In 5th International Semantic Web Conference. Ed. Springer, LNCS 4273, pp. 44-57. Athens, USA. November 2006.
8. N. Ide, J. Véronis. Word Sense Disambiguation: The State of the Art. Computational Linguistics, 24(1), 1998.
9. G. S. Mann, D. Yarowski. Unsupervised Personal Name Disambiguation. In 7th Conference on Natural Language Learning. Edmonton, Canada. June 2003.
10. L. Page, S. Brin., R. Motwani, T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Stanford Technical Report available online at: http://dbpubs.stanford.edu/pub/1999-66, 1999
11. T. Pedersen, A. Purandare, A. Kulkarni. Name Discrimination by Clustering Similar Contexts. In 6th International Conference on Computational Linguistics and Intelligent Text Processing. Ed. Springer, LNCS 3406. Mexico City, Mexico. February 2005.