NEREA: Named Entity Recognition and Disambiguation Exploiting Local Document Repositories

Ángel L. Garrido, Sergio Ilarri
IIS Department
University of Zaragoza
Zaragoza, Spain
Email: {**garrido, silarri**}**@unizar.es**

Susana Sangiao, Adrian Gañán, Alejandro Bean, Óscar Cardiel
Computer Department
GRUPO HERALDO
Zaragoza, Spain
Email: {**ssangiao,aganan,abean,ocardiel**}**@heraldo.es**

***Abstract*—In this work, we describe the design, development, and deployment of NEREA (Named Entity Recognizer for spEcific Areas), an automatic Named Entity Recognizer and Disambiguation system, developed in collaboration with professional documentalists. The aim of NEREA is to keep accurate and current information about the entities mentioned in a local repository, and then support building appropriate infoboxes, setting out the main data of these entities. It achieves a high performance thanks to the use of classification resources belonging to the local database. With this aim, the system performs tasks of named entity recognition and disambiguation by using three types of knowledge bases: local classification resources, global databases like DBpedia, and its own catalog created by NEREA. The proposed method has been validated with two different datasets and its operation has been tested in English and Spanish. The working methodology is being applied in a real environment of a media with promising results.**

***Index Terms*—Named Entity Recognition, Infoboxes, Disambiguation, Knowledge Obtention**

I. Introduction

Thanks to advances in computing, today everyone expects to find accurate and detailed information about people, places, organizations, or events both in public and private electronic repositories. Examples of such repositories are libraries, public archives, private databases, and of course the Web. The problem is that the answers to a search based on keywords is usually a list of links to documents or web pages that the user should read one by one to find detailed information, which is a costly, tedious, and error-prone task.

To avoid this, in the last years data sheets have been created in order to provide a summary of the most important data about the entity searched by the user. Examples of this type of data sheets, commonly known as *"infoboxes"*, can be found in both general-purpose Web search engines such as Google or encyclopedic tools like Wikipedia. The construction of these infoboxes can be manual, semi-automatic, or fully automatic, which means going from a more expensive but more accurate option to another one prone to errors but with less human construction cost. Mistakes are often due to problems with the automatic extraction of data from documents. The difficulty of interpreting natural language and the frequent ambiguity of the terms are two issues that are under investigation today to improve processes such as those discussed above.

Although no one doubts the usefulness of infoboxes, their automatic or semi-automatic creation is far from a trivial task. Concerning local databases of documents, it is a problem that requires information extraction tasks from both private repositories and public repositories, because users of such information systems expect to find detailed information from both private databases and public information freely accessible through the Web.

In order to create an infobox catalog of data entities based on textual information of a documental database, it is necessary to perform different processes. One of the major tasks is the recognition of the valid named entities from the text of the documents, known as the Named Entity Recognition (NER) task. The list of named entities obtained through this process is the basis to generate infoboxes by extracting information from the documents. One important difficulty within this task is that the names of these entities are frequently ambiguous. Therefore, it is also mandatory to execute a Named Entity Disambiguation (NED) task to solve this ambiguity. Although the most recent approaches use DBpedia or Wikipedia as the main resource in order to disambiguate entities, this choice has strong limitations when the target is a private repository of documents with very specific and local information (e.g. information belonging to a specific region or city), because most of the important entities have sparse or none information in a global repository like Wikipedia. Typical examples are private collections, regional archives, or local media repositories, which can contain detailed and valuable information about characters, organizations, or places with poor information about them in Wikipedia or even on the whole Web. Moreover, these repositories can have their own classification tools, sometimes with complex and very informative formats like indexes, taxonomies, thesaurus, or even ontologies. These resources could be a very valid instrument for identifying and classifying named entities. Finally, a sort of *"slot filling task"* has to be executed in order to complete the infoboxes. Slot

filling tasks have the aim of extracting the values of specified attributes (or slots) for a given entity from large collections of natural language texts. To accomplish this task, multiple information extraction processes must be launched over the documental database, each of them specific for each attribute of the infobox. The attributes are previously defined by hand or by using automatic algorithms based on the typology of the entities.

In this paper, we describe a new approach for the design of an automatic NER and NED system, whose main purpose is to obtain infoboxes, but with the particularity of being intended for local environments. This methodology has been applied to the development of a software called "NEREA" (Named Entity Recognizer for spEcific Areas), which is able to identify the most outstanding entities in a text-based catalog by taking advantage not only of the global web resources, but also of the specific and local knowledge resources. The main contribution of this work is to make use of both the local features of the database and the local classification resources in order to improve the entity catalog process, and hence to achieve higher-quality infoboxes. Besides, the proposed system is evaluated in the real environment of a newspaper. We have rigorously tested the system using two different datasets: one Spanish local dataset of regional news, coming from *Heraldo de Aragón*[1], and one English public access dataset (OKE 2015 dataset). The experiments performed prove its multilingual feature, and they show a very good outcome on both datasets, especially in the context of a real local document database.

This paper is organized as follows. Section II studies the state of the art related to NED and tasks of Infoboxes Generation. Our methodology is detailed in Section III. Section IV is devoted to explaining the slot filling task of the infoboxes. Section V explains the different experiments performed and interprets the outcomes. Finally, Section VI summarizes the key points of this work, provides conclusions, and explores future work.

II. Related Work

In this section, we describe the state of the art by reviewing the general aspects of named entity disambiguation and slot filling, listing some existing approaches.

Approaches that used external knowledge sources containing information about entities (e.g., Wikipedia) appeared in 2003 in [1]. Specifically, the authors attempt large-scale taxonomy based disambiguation over a collection of 264 million documents (although the number of mentions to disambiguate was limited to 550 million).

Later, in 2006, in [2], article titles, hyperlinks, and disambiguation pages are used to generate candidate entities. The main core of this work is the use of Support Vector Machines (SVM) [3] for disambiguation tasks. Because of the lack of datasets for public use, the authors defined their own test environments. In 2007, in a similar work [4], but using vector space models for disambiguating all the named entities in the text simultaneously, the authors added a global constraint that requires target Wikipedia articles candidates to come from the same category, and added to those candidates linking information whenever a given anchor-text mentions the same target entity from two different Wikipedia pages.

Afterwards, in 2009, in [5], a system with a simple text similarity approach for disambiguation was implemented, but the authors gave special attention to the step of generating candidates. The authors used Bayes and K-Nearest Neighbours classifiers. This system achieved the best result in the TAC (Text Analysis Conference) 2009[2] with a 82.2 % of accuracy.

A task that also appears in the scientific literature and is related to NED is *wikification*. This task includes deciding which keywords or concepts are relevant in a given text and then disambiguating them by linking to the correct Wikipedia article. Although the general aim of this task is different, because wikification systems target both proper and common names, their disambiguation techniques are relevant to NED. As an example, in [6], the authors used mentions in anchors to train a supervised Naive Bayes classifier for wikification.

In 2013, in [7], the authors reimplemented the three NED systems mentioned in [2], [4], and [5], combined them, and carefully analyzed the performance of their candidate generation and disambiguation components. They studied contributions from a variety of available candidate sources, including titles of articles, hyperlinks, disambiguation pages, and link anchors. Besides, they considered two additional heuristics: bold text that appears in the first paragraph and the hatnote templates from popular entities that correspond to disambiguation pages. The surprising conclusion of this thorough comparison work is that most current NED systems focus on the disambiguation stage, whereas, *stage of search and generation of candidate entities* seems to have more impact on the final accuracy. They have found that, for the last disambiguation stage, a simple vector space model performed surprisingly well compared to other more interesting, implemented disambiguation strategies, computationally more costly. Following this conclusion in a context devoted to the recognition of entities in local repositories, we consider that the main task has to be to create and maintain specific catalogs in order to help the NED system in this task of searching and generating the candidate entities. Moreover, it is important to consider that such resources have to be the basis of knowledge available within the context of the repository, but in combination with other global and external resources, in order to complement the information covered in the local database.

Regarding the issue of "Slot Filling", there are two major tasks: information extraction and generation of infoboxes. The methods for obtaining information automatically from natural language texts are framed in the context of Information Extraction [8]. The first systems were mainly based on rules [9], [10]. However, manual coding was a tedious work and the

[1]http://www.heraldo.es

[2]http://www.nist.gov/tac/

algorithms exhibited low performance, so systems started to learn rules automatically from examples [11]. Later, statistical learning techniques emerged [12], [13], and grammatical construction techniques [14] were developed. These methods, based on rules and statistical methods, are being used at the same time depending on the nature of the extraction tasks. There are also hybrid models [15], that seek to gather the benefits of both types of techniques. Among recent work on this topic, is worth mentioning [16], where the authors introduced *KnowItAll*, a system which is able to extract information from the Web without hand-labeled training examples. Furthermore, there are more specific systems that focus on extraction information from Wikipedia, like *KYLIN* [17] or iPopulation [18], whose function is to generate infoboxes from Wikipedia automatically. Another relevant project in this scope is DBpedia [19], a knowledge base that contains a vast amount of data which is obtained by extracting structured information from Wikipedia.

Regarding the infoboxes generation task, in 2008 a system called Freebase [20], a knowledge base which maintains structured and interlinked data, appeared. Part of the data for each entity was obtained from Wikipedia, but the data population is based on user collaboration. Finally, IBminer [21] is a system to derive structured information from Wikipedia using natural language processing, but unlike Freebase it does not contain templates for categorizing the entities.

If we place these works in the context of creating infoboxes in a local document database, more control over customizing the construction of these infoboxes, or even the definition of templates, is missing. Moreover, the generation of infoboxes should also benefit from the use of the local resources for classifying and tagging, like taxonomies, thesaurus, or ontologies, as we have proposed in previous proposals such as [22] and [23]. Finally, it is also necessary to use information extraction tools able to obtain data concerning the entities, both from the local repository and from open and global repositories located on the Web.

III. System Overview

NEREA is a system that recognizes relevant entities from a text in a local document database, and disambiguates them thanks to a subsystem that we have denominated *POIROT*[3]. Previously, NEREA generates from the local database a lexical catalog of named entities to keep all the potential candidates from each source of information. During the identification process, when ambiguity exists, the system delegates to POIROT the task of choosing the most appropriate entity. POIROT needs extra information to perform its function, so it gets the data from two types of knowledge sources: local and global. Local information can have different formats: a close list, a thesaurus, or even an ontology. Global information is obtained from a generic and public repository, like DBpedia. The result of this process is an *Entity Catalog* that contains a set of disambiguated relevant entities within the local repository, which are used to represent people, organizations or locations. Figure 1 shows an overview of the process, which is divided in several steps, explained below.

A. Previous Processes

NEREA has some auxiliary processes that extract and calculate important information from the local documental database to be used later. One of these processes, called *lemmatization*, extracts the canonical form, or lemma[4], of the words from the texts with the aim of simplifying them. Further information about the advantages of lemmatization in certain languages such as French, German, or Spanish can be found in [24]. Besides, the lemmatization process detects *all the named entities* thanks to Freeling[5], an open source language analysis tool suite available for several languages. This process generates a *local named entity catalog*, which will be a lexical resource used when the recognition and disambiguation of entities is performed. Another auxiliary process calculates the most relevant words (keywords) in the text using the TF-IDF and TF-WP algorithms [25]. These keywords are the basis to build the elements that POIROT needs to disambiguate the named entities.

B. Inputs of the System

NEREA receives as the input a named entity, detected by using Freeling, and its context, i.e., the whole text where that named entity is located. The context will help to select the entity referenced by the named entity. For each named entity detected in a text, the system checks if it already exists in the named entity catalog. In case it does not, NEREA searches all the information related with the named entity in the local and global knowledge sources. In each of these sources, the system tries to locate all the possible candidates that match with the named entity, and a list of candidates is created. Each of the candidates is identified according to its origin: labels, thesaurus descriptors, local URIs (in case of using a local ontology), or global URIs in the case of DBpedia.

Besides, NEREA adds, to the set of DBpedia URIs, every URI contained in a collection extracted from DBpedia Spotlight[6], a tool for automatically annotating mentions of DBpedia resources in text. This collection has a set of named entities related with some precalculated DBpedia URIs, and it is very useful at the beginning of the Entity Catalog generation process to obtain extra information with good performance.

With the most relevant words of the text where the named entity appears, NEREA calculates the Bag of Words (BOW) and builds the context vector with weights previously calculated by TF-IDF and TF-WP algorithms for each word. This vector will be compared with the context vector generated by the same procedure for each candidate, using the cosine similarity, a common method used to measure the similarity using the BOW model. NEREA measures the similarity between

[3]In honor of the famous detective of Agatha Christie's books.

[4]In English, for example, *sing, sings, sang, sung, and singing* are forms of the same lexeme, with *sing* as the lemma.

[5]http://nlp.lsi.upc.edu/freeling

[6]http://spotlight.dbpedia.org/

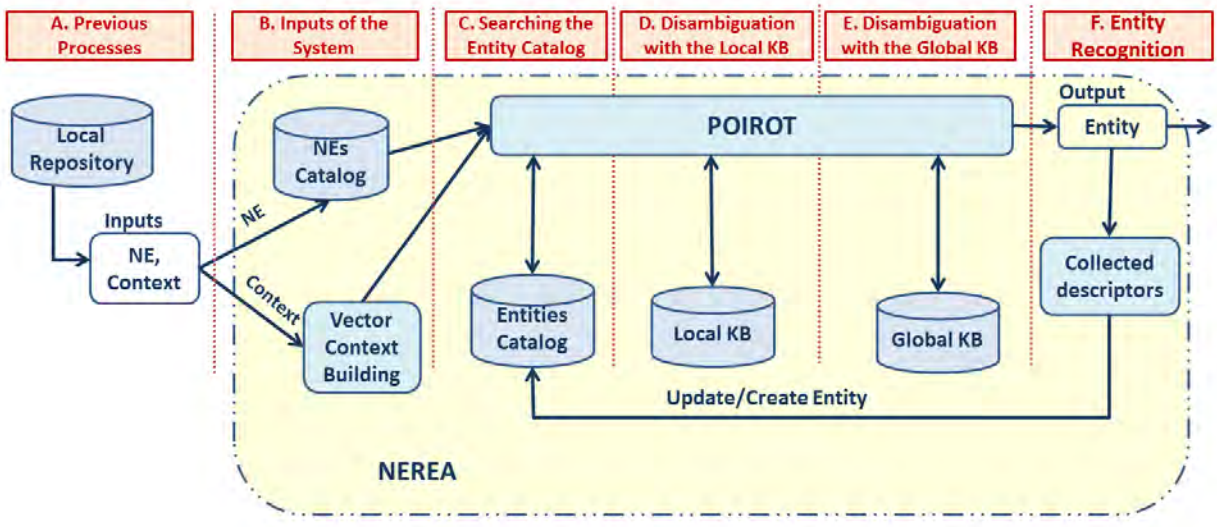

Fig. 1. System Overview of NEREA.

two BOWs by creating context vectors for each BOW and calculating the cosine similarity between the vectors according to the following formula:

$$sim(V_{context}, V_{candidate}) = \cos\theta = \frac{V_{context} V_{candidate}}{|V_{context}||V_{candidate}|}$$

$$= \frac{\sum_{i=1}^{n} V_{context}(i) V_{candidate}(i)}{\sqrt{\sum_{i=1}^{n} V_{context}(i)^2}\sqrt{\sum_{i=1}^{n} V_{candidate}(i)^2}}$$

During the system tests, the system has been tested with different context vector sizes. The best results have been obtained for vector sizes between 50 and 120, so the vector size has been set to 50 to improve the system performance.

C. Searching the Entity Catalog

The system obtains the entities that have the current named entity as a named entity associated from the Entity Catalog. For each candidate entity, the context vector of each candidate is generated, which will be compared to the context vector of the named entity by computing the cosine similarity. The entity with the highest similarity will be selected, if its similarity exceeds the threshold established (see Algorithm 1), and the process will finish. In case none accomplishes this condition, the process will continue the disambiguation with the Local KB step. The following steps will be performed by POIROT, the subsystem devoted to disambiguation tasks.

D. Disambiguation with the Local Knowledge Base

At this point, the system has a set of candidates with some identifiers of the local knowledge bases (Local KB) that can represent the entity, but a disambiguation process is necessary to select the correct one, following a similar disambiguation process than in the previous step: for each identifier of the set of candidates of the Local KB, the most relevant texts tagged with these local resources are obtained and a new context vector is generated for each candidate. The context vector of the named entity and the context vectors of each candidate will be compared with the cosine similarity, and the Local KB identifier of the text with the highest similarity will be selected, if its similarity value exceeds the similarity threshold. Once an identifier has been selected, the process tries to locate, in the Entity Catalog, the entity that is assigned to one element of the Local KB. In case it is located, it will be selected as the disambiguated entity and the process will finish. Otherwise, the system must go to the Disambiguation with the Global KB step, keeping the local identifier selected.

E. Disambiguation with the Global Knowledge Base

POIROT checks if the named entity is in the DBpedia Spotlight *pair count* collection. This collection is a pair-wise occurrence count that keeps track of anchor texts in DBpedia articles, looking for different named entities that appear linked to a DBpedia article and the count of the number of times this link occurs. Then, the system adds the URIs assigned to the named entity in this collection to the set of candidates generated in the first step, with the aim of completing all potential candidates. Besides, the system exploits the information contained in the DBpedia disambiguation pages, which means that, if the system detects that one of the candidates is a disambiguation page, it collects all URIs contained on that page and adds them to the candidate set. After that, for each URI in the candidate set, POIROT obtains the HTML page of DBpedia and extracts a BOW from it, which is used to generate context vectors for each candidate. All of these context vectors are compared with the context vector of the named entity by using the cosine similarity, following Algorithm 1.

As in the other cases, the URI with the highest similarity will be selected if it exceeds the threshold. In this step,

exceptionally, if no URI has exceeded the threshold of similarity, we choose the URI that has the greatest count in the DBpedia Spotlight pair count collection. This means that when the context is not sufficient to choose one we rely on the information provided by the DBpedia Spotlight pair count collection. The next step is to check if any entity from the catalog has this URI associated, in which case that entity will be selected.

F. Entity Recognition

At this point, there are two possible situations:

1) *One entity from the Entity Catalog has been selected.* In this case, the record corresponding to the entity in the catalog must be updated with all the info harvested during the disambiguation process: local identifiers corresponding to it, URI from DBpedia, and other NEs collected from DBpedia Spotlight that are associated with the selected URI, if that is the case.
2) *No entity has been selected.* If any local or global identifier has been selected, an entity will be created with that associated identifier, and then NEREA populates the Entity Catalog. If no identifier has been selected, it is likely that a false positive has occurred in the identification process, i.e., something *that is not a named entity* has been identified as a named entity, or the named entity appears in a very different context from its usual context. In these cases, given the correct parameters of the system set in the training period (size of the context vectors and similarity threshold), the named entity will be discarded.

G. Summary

As it has been seen, the purpose of this process is to obtain unique and unambiguous entities, with local and global descriptors, which are able to provide knowledge to the entity. Applying this process to all the named entities of the local repository, the Entity Catalog is built. Then, NEREA will be able to perform the disambiguation tasks automatically, with that catalog as the main resource of the local environment.

The Entity Catalog generated will be used later to perform the slot filling task needed to create infoboxes, which is explained in the next section.

Algorithm 1 Disambiguation Algorithm for every resource: Catalog of Entities, Local KB and Global KB.

function DISAMBIGUATION(NamedEntity NE, Context C, Resource r) as Entity
  $V_{context} = createContextVector(C)$
  $maxSimilarityValue = 0$
  for each $candidate$ in $Candidates(r, NE)$ **do**
    $V_{candidate} = createCandidateVector(candidate)$
    $similarityValue = cosine(V_{context}, V_{candidate})$
    if $similarityValue > GreaterSimilarity$ **then**
      $mostSimilarCandidate = candidate$
      $GreaterSimilarity = similarityValue$
    end if
  end for
  if $GreaterSimilarity > similarityThreshold$ **then**
    return $mostSimilarCandidate$
  end if
  return $null$
end function

IV. INFOBOXES GENERATION

In this section, we explain the process of harvesting data for each entity stored in the Entity Catalog previously obtained through NEREA, with the aim of generating infoboxes for displaying the information to the final users. As the aim is to work in the specific domain of the local documental database, we have introduced a tailored tool (specific to documentalists and to technical staff) whose purpose is to create *templates*. These templates are important elements to support the process of extracting information from the documents, because they allow 1) to define specific entity categories, for example, football players, 2) to select the most suitable attributes for each infobox, and 3) to link each attribute with a specific information extraction method, able to find the desired data to accomplish the aforementioned slot filling task. A template is defined by a name, a category (person, organization, event, or topic), and a list of attributes belonging to different typologies (text, number, date, etc.), which can be grouped into several sets (for example, *personal data*; including the attributes name, surname, date of birth/death, civil status, and birthplace).

Therefore, NEREA includes a complementary software component, a *templates edition tool*, which uses the aforementioned Entity Catalog created by NEREA, and takes advantage of the local knowledge base resources (taxonomies, thesaurus, ontologies, etc.). A screenshot of this software can be seen in Figure 2. With this tool the documentalists are able to:

1) Create, modify, and delete templates.
2) Create, modify, and delete attributes within a template, and to assign a typology to each of them.
3) Define sets of attributes in a given template.
4) Link these templates with their corresponding entities in NEREA's catalog. This option *generates* an initially empty infobox related to each one.
5) Classify the template within a taxonomy of templates. This option is very useful to implement inheritance options among the templates. For example, the template *politician* can be a "child" of the template *person* and then it inherits the attributes of that template.
6) Manage this taxonomy of templates.

For accomplishing these tasks, the software aids the user with a user-friendly GUI, that provides information from the local KB when necessary. For example, if a *person* is defined in the local ontology, when the user wants to create a specific template for a group of people (for example, *athletes*), this tool prepares an automatic template based on the characteristics and the attributes of the concept of *person* in the ontology.

On the other hand, an interesting feature of the spot filling task is to seek information in various data sources: the main

Fig. 2. Screenshot of the template managing software of NEREA.

data source is the local knowledge base, although the system extracts information from other global KBs like DBpedia. In that way, we enhance our local information and improve the KB that supports the system.

The spot filling task of the previously defined infoboxes can be performed in two ways:

- *Manual:* The documentalist users fill directly the attributes. This is, of course, a time-consuming task, so it is only recommended for isolated cases.
- Automatic: The slots of the attributes are filled using automatic processes, depending on the classification of the entity and the typology of the attribute. The allocation of procedures to each infobox attribute can be a manual task made by the information extraction specialist of the computer department, or an automatic task. The second option is one of the future objectives of NEREA, although it is out of the scope of this work for the moment.

V. Experiments

For testing the system, two major experiments have been made: the first one with a well-known English data set used for testing entity-linking tools like NEREA. The second one has been a local dataset from a documental database of *Heraldo de Aragón*[7], a Spanish regional media. In this section, a comparative study of the two experiments can be found, together with some considerations about the generation of infoboxes.

A. OKE Evaluation Dataset

There is no established benchmark for NED. Most published papers define their own test environment with subsets of Wikipedia dumps or subsets of known corpus manually annotated. Other works use datasets obtained through participation in conferences, which are mostly not available for public access. Among the datasets which are available online, our system has been tested with the OKE 2015 dataset[8], which is provided every year for the Open Knowledge Extraction challenge. It contains the training and evaluation datasets, which have been built by manually annotating 196 sentences. These sentences have been selected from Wikipedia articles reporting the biographies of scholars to cover NEs for people, locations, organizations, and roles.

By using this dataset, the results of our system in the task of entity linking can be compared directly with other systems that have been tested with the same public dataset. We extract the comparison data from [26]. These results can be observed in Table I, in terms of precision (P%), recall (R%), and F-measure (F1%).

TABLE I
Comparison on the English OKE evaluation dataset in the Entity Linking Task

	OKE 2015 dataset		
	P%	R%	F1%
Adel-without pruning	49.4	46.6	48
Adel-with pruning	57.9	6.2	11.1
AIDA	51.6	43.9	47.4
TagMe	28.5	54.9	37.5
DBpedia Spotlight	28.3	45.7	34.9
Approach described in [26]	73.1	46.1	56.5
NEREA	79.4	59.5	**68**

According to the data shown in the Table I, it can be seen that our system outperforms all previous approaches, both in terms of precision and recall, and F1-measure. In Figure 3, the change in the F-measure (F1%) depending on the similarity threshold can be observed, given a context vector size equal to 50 (see explanation in Section III-B). The best system performance is obtained with a threshold value of 0.59. Due to weak contextual information present in this dataset, this figure is very similar to the one obtained with context vector sizes from 50 to 120.

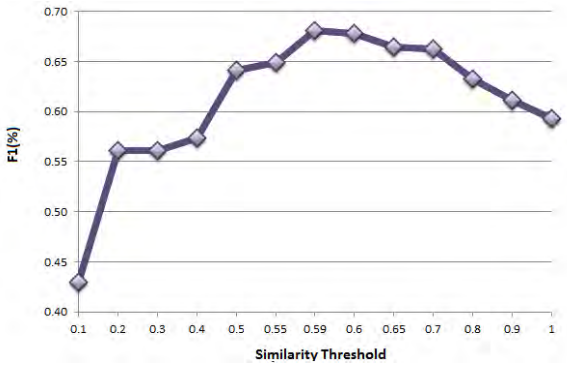

Fig. 3. F-measurements obtained depending on the similarity threshold with the size of context vectors = 50.

B. Local Documental Dataset

The main themes of the Heraldo's dataset are local politics and local sports. This is a problem to perform disambiguation tasks, because in many cases characters and locations that appear in the news do not exist on the Internet, have no entry in

[7]http://www.heraldo.es
[8]https://github.com/anuzzolese/oke-challenge

Wikipedia, and the system has no other sources from which to extract information. Hence, it is very important for NEREA to have its own databases containing all the people, places and organizations that are related to the news in the archive. For example, one of the entities that most often appears in local news is the local football team; this team is now in the second division of the Spanish football league, which means that many players are not so well known and, in most of cases, they will not be referenced in Wikipedia or in other sources of information. The same applies to other local characters, for example, the president of the Chamber of Commerce of the region, whose name (D. Manuel Teruel) appears in many news on local economy, but his profile is not in Wikipedia.

As training data, the system has harvested all the entities which appear in the news published during 2014 and 2015. With these training texts, we have populated our local knowledge base and our catalogs of Entities and Named Entities. Once we have manually evaluated the correct behavior of the system, as a dataset for Spanish entities, we have drawn randomly 5,000 news with ambiguous references to entities from the local repository of news. These ambiguous mentions in the text have been disambiguated by our system, and the correct referenced entity has been manually annotated by the Documentation Department staff in order to check if the system obtains the right answer.

We have first checked our local dataset by performing only the stage of disambiguation with a global knowledge base, in our case with the DBpedia Spotlight pair counts collection. This collection has a good behavior with well-known global entities, but not for local characters and places. For these local entities, the accuracy of the disambiguation process with our local KB is better than with a global knowledge base. The data in Table II reflect the precision, recall, and F1 scores when using the aforementioned best combination of the size of context vectors (50) and similarity threshold (0.59).

TABLE II
BEST RESULT FOR LOCAL ENTITIES DISAMBIGUATION ONLY WITH THE GLOBAL KB AND WITH THE COMPLETE SYSTEM

	Spanish Local Dataset		
	P%	R%	F1%
Global KB only	55.2	45.3	49.7
NEREA	83.5	62.4	71.4

C. Analysis of the Results

If we only consider the local entities, the performance of our system is far better than that of the others, because if we work with entities for which there is no information on global KBs, the only source of information is our local KB. This is so for one out of every eight entities in our catalog, so we can say that in 12.5% of cases our system will be the only one with the necessary resources to disambiguate a mention.

If we compare the system performance between the two datasets, we note that the performance of the system with our dataset is considerably better, because the contextual information available in our dataset is much higher than on the OKE 2015 dataset (the average length of sentences included in this last dataset is 10 words, which is insufficient information for performing an accurate context comparison). When the length of the text where the entity appears exceeds 100 words, our accuracy is always above 70%.

As a first approach to the ambiguity problem with references to entities, the performance of our system is rather acceptable (see Table I). There are still some cases where the system fails to select the correct entities. In these cases, it is often not even clear to humans which entity is referenced in the text, because the amount of contextual information is not enough or because of the high similarity between two entities (i.e., in the context *"Iglesias will give a concert in Miami"*, even humans cannot discriminate if the named entity "Iglesias" refers to the entity "Julio Iglesias" or to the entity "Enrique Iglesias").

D. Generation of Infoboxes

Regarding the generation of infoboxes, in order to implement both repositories of templates and infoboxes, and after conducting an evaluation of alternatives (XML files, relational databases, document-oriented databases, etc.) the use of a graph database was decided. The reason to adopt this technology is that templates do not have a prefixed schema. Although the most typical schema-less databases are document-oriented, graph databases are also used and, in terms of performance, graph databases are better than document-oriented databases. Besides, if one of the ultimate goals is the publication of the database in Linked Data format to be accessible through a SPARQL end-point, using a graph database will facilitate the process of creation. Anyway, this is not a critical decision and other deployment options could be evaluated. An example of the final aspect of a complete infobox produced through NEREA can be seen in Figure 4.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we have presented NEREA, a NER system specific for recognizing and disambiguating the entities referenced in a local repository. As soon as entities are discovered and disambiguated, the system also allows generating customized infoboxes for each of the entities.

The performance of NEREA is pretty good given a large enough context of the entity is available to disambiguate it, which is consistent with the conclusion of [7]: "focusing efforts on the stage of generation of the candidate entities achieves a good performance without the need to use more complex techniques, which are resource-intensive for local repositories". Besides, using the OKE-Evaluation Dataset, we have proven that the results of NEREA are better than those of the previous approaches, even when contextual information is scarce.

The main contributions of this work are:

1) To study the use of local classification resources in order to improve NER processes over local databases.
2) To investigate the advantages of using these local features in order to improve the generation of infoboxes from an entity catalog.

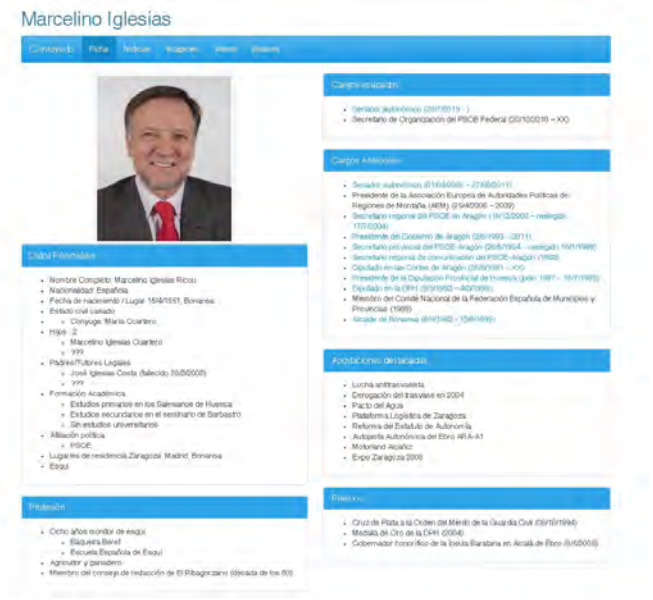

Fig. 4. An example of an infobox obtained by the NEREA system.

3) To implement and evaluate a system which exploits these characteristics with public and private datasets.
4) To analyse the results and compare the proposal with previous works.

There are several lines of development for future work. We consider that the most important one is to continue improving the NER results. For doing this we want to experiment with machine learning techniques when the system detects that the context in which the entity appears does not have enough information to successfully disambiguate it. Regarding the infoboxes generation, there is a lot of work to do in order to automatically fill infoboxes by using information extraction procedures, but thanks to NEREA, which provides a good set of entities, such work may be more accurate.

Acknowledgment

This research work has been supported by the CICYT project TIN2013-46238-C4-4-R, and DGA-FSE. Our gratitude to the Documental Department of Heraldo de Aragón, and to Andrea Aleyxendri.

References

[1] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin *et al.*, "Semtag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation," in *12th International Conference on World Wide Web*. ACM, 2003, pp. 178–186.

[2] R. C. Bunescu and M. Pasca, "Using Encyclopedic Knowledge for Named Entity Disambiguation." *EACL*, vol. 6, pp. 9–16, 2006.

[3] T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Springer, 1998.

[4] S. Cucerzan, "Large-Scale Named Entity Disambiguation based on Wikipedia Data," in *EMNLP-CoNLL*, vol. 7, 2007, pp. 708–716.

[5] V. Varma, P. Bysani, V. B. Kranthi Reddy, K. K. Santosh GSK, S. Kovelamudi, N. Kiran Kumar, and N. Maganti, "IIIT Hyderabad at TAC 2009," in *Test Analysis Conference 2009 (TAC)*, 2009.

[6] R. Mihalcea and A. Csomai, "Wikify! Linking Documents to Encyclopedic Knowledge," in *16th Conference on Information and Knowledge Management*. ACM, 2007, pp. 233–242.

[7] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran, "Evaluating Entity Linking with Wikipedia," *Artificial intelligence*, vol. 194, pp. 130–150, 2013.

[8] S. Russell, P. Norvig, and A. Intelligence, "Artificial Intelligence: A Modern Approach," *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, vol. 25, 1995.

[9] E. Riloff *et al.*, "Automatically Constructing a Dictionary for Information Extraction Tasks," in *11th National Conference on Artificial Intelligence (AAAI-93)*, 1993, pp. 811–816.

[10] R. Grishman, "Information Extraction: Techniques and challenges," in *Information extraction a multidisciplinary approach to an emerging information technology*, 1997, pp. 10–27.

[11] C. Cardie, "Empirical methods in information extraction," *AI magazine*, vol. 18, no. 4, p. 65, 1997.

[12] V. Borkar, K. Deshmukh, and S. Sarawagi, "Automatic segmentation of text into structured records," in *ACM SIGMOD Record*, vol. 30, no. 2, 2001, pp. 175–186.

[13] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation." in *27th International Conference on Machine Learning (ICML)*, vol. 17, 2000, pp. 591–598.

[14] P. Viola and M. Narasimhan, "Learning to Extract Information from Semi-structured Text using a Discriminative Context Free Grammar," in *28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, pp. 330–337.

[15] M. E. Califf and R. J. Mooney, "Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction," *The Journal of Machine Learning Research*, vol. 4, pp. 177–210, 2003.

[16] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised Named-Entity extraction from the Web: An experimental study," *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[17] F. Wu and D. S. Weld, "Autonomously Semantifying Wikipedia," in *16th Conference on Information and Knowledge Management*. ACM, 2007, pp. 41–50.

[18] D. Lange, C. Böhm, and F. Naumann, "Extracting Structured Information from Wikipedia Articles to Populate Infoboxes," in *19th International Conference on Information and Knowledge Management*. ACM, 2010, pp. 1661–1664.

[19] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," in *The Semantic Web*. Springer, 2007, pp. 722–735.

[20] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge," in *ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250.

[21] H. Mousavi, S. Gao, and C. Zaniolo, "IBminer: A Text Mining Tool for Constructing and Populating Infobox Databases and Knowledge Bases," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1330–1333, 2013.

[22] A. L. Garrido, A. Peiro, and S. Ilarri, "Hypatia: An expert system proposal for documentation departments," in *12th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2014, pp. 315–320.

[23] A. L. Garrido, P. Blázquez, M. G. Buey, and S. Ilarri, "Knowledge Obtention Combining Information Extraction Techniques with Linked Data," in *24th International Conference on World Wide Web*. ACM, 2015, pp. 643–648.

[24] A. L. Garrido, O. Gómez, S. Ilarri, and E. Mena, "An experience developing a semantic annotation system in a media group," in *International Conference on Application of Natural Language to Information Systems*. Springer, 2012, pp. 333–338.

[25] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[26] F. Nooralahzadeh, C. Lopez, E. Cabrio, F. Gandon, and F. Segond, "Adapting Semantic Spreading Activation to Entity Linking in Text," in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2016, pp. 74–90.