Improving the Generation of Infoboxes from Data Silos through Machine Learning and the use of Semantic Repositories

Angel L. Garrido[1], Susana Sangiao[2], and Oscar Cardiel[2]

[1] SID Research Group
University of Zaragoza
Zaragoza, Spain
Email: **garrido@unizar.es**
[2] Computer Department
HENNEO
Zaragoza, Spain
Email: {**ssangiao,ocardiel**}**@henneo.com**

Abstract. Nowadays, both public and private organizations own large private text-based data repositories with critical information. The information stored in these data silos is usually queried through information retrieval systems based on indexes, which yield hundreds or thousands of results when interrogated using keywords. In order to improve data accessibility when searching for specific information, the use of *infoboxes* can be very useful. The generation such infoboxes is by itself a complex problem, but in this type of isolated environments, it becomes even harder as the selection of the entities and their attributes can be conditioned by local and very specific parameters. In this work, we propose a methodology to tackle this special problem, combining classical approaches with machine learning, and leveraging the resources provided by the Semantic Web. The working methodology has been applied to two well-known datasets, and also it has been tested on a real environment scenario, showing the feasibility of our approach.

Keywords: Infoboxes, Named Entity Disambiguation, Information Extraction, Machine Learning

1 Introduction

An *infobox* is a standardized table with information about an entity referred in a text-based document[3], i.e., a set of attribute / value pairs describing the entity and summarizing its most outstanding data [1,2]. Nowadays, infoboxes are usually present in popular web search engines, such as Google[4], Yahoo[5], or Bing[6]; on online encyclopedias de-

[3] They are usually used to present the information about a person, a place, a work, an organization, or an event.
[4] https://www.google.es/
[5] https://es.search.yahoo.com/
[6] https://www.bing.com

voted to general information as Wikipedia[7], or Encyclopedia of life[8]; and on websites specialized in music[9], movies[10], or news[11]. Their usefulness is clear: when a user of an information system is looking for specific and accurate data, these tables of values will prevent her/him from having to *dive* in a high number of links in order to find the information sought, which is a time-consuming and humdrum task. Moreover, the desired information could not be found or be erroneous, which is also easily spotted when displayed using this formatting.

Although the generation infoboxes can be done manually (i.e., defining the infobox for each entity), it is preferable to build them as automatically as possible. However, the automatic generation of such infoboxes and their population is not error-free because it usually requires to extract information from text-based documents, which is yet an open problem. In particular, we need to interpret natural language in order to extract the relevant information [3, 4], and to disambiguate the terms that we find within the texts [5]. Besides, the generation of infoboxes working with data silos (i.e., private data warehouses, without access from the outside, and presumably with very local information to the domain of the company or organization that maintains them) has a special difficulty, since it is very probable to find entities whose relevance is outstanding in the context of the data silo, but not out of it. As a result, many tools which rely on general sources of knowledge (e.g., the Web) to perform disambiguation tasks are not able to disambiguate correctly the entities in the silo. Finally, we have to remark the need of using public data repositories to enrich as much as possible the information stored in the private repository, making it possible to present the users with better and complete information about their searches.

In this paper, we tackle the problem of generating a set of relevant infoboxes in the context of a text-based data silo. To solve this problem, we have defined a working methodology based on leveraging both the information stored in the knowledge bases related to the private repository, and the data obtained from external sources exploiting resources of the Semantic Web. This methodology is based on performing a deep analysis of the documents of the data silo with a two-fold objective: 1) to obtain information which we can leverage to disambiguate local entities that are written similarly; 2) to apply a supervised machine learning method to identify the typology of the entities stored in it, allowing the system to choose the best way of generating the corresponding infoboxes. Our claim is that improving the identification of the entities leads to better outcomes regarding the generation of the infoboxes in such type of isolated scenario. Finally, the use of a combination of local and external knowledge bases helps to create an enhanced entity catalog from which it is more feasible to generate the infoboxes.

The main contribution of this work is to present a specific approach in order to help in the creation of infoboxes from the information contained in a private catalog of text-based documents. The methodology has been implemented through the design of a system that adopts the ideas proposed. This system has been rigorously evaluated with

[7] https://es.wikipedia.org
[8] http://eol.org/
[9] http://www.allmusic.com
[10] http://www.imdb.com
[11] http://www.bbc.co.uk

different datasets. The most outstanding has been a Spanish local dataset of regional news, coming from *Heraldo de Aragón*[12], since it represents very accurately the type of problem exposed. In this work we have taken a step forward with respect to what we have exposed in previous works. In [6] we introduced the general problem of generating infoboxes in data silos and we hypothesize about possible solutions. Later, in [7], we analyze in greater depth the problem, putting the focus on the disambiguation of the entities (Named Entity Disambiguation -NED- process) with the aim of helping to obtain a better result in the generation of infoboxes, but with the particularity of being intended for local environments. That methodology was applied on the development of NEREA (Named Entity Recognizer for spEcific Areas), a system which identifies the most relevant entities in a private set of text-based documents. In this work, we have extended our previous contributions as follows:

- The general methodology of our approach is explained at a higher level.
- We have improved the selection of templates with an automatic mechanism leveraging machine learning techniques.
- An extension of the experimental results has been made with new datasets and new annotation tools, and a more detailed feedback of the system in a real production environment has been obtained, providing a thorough validation of our approach.

This paper is structured as follows. Section 2 explains the methodology proposed both for generating the corresponding infoboxes and for disambiguating the entities. Section 3 describes how we have implemented this methodology developing and integrating different systems. The experiments and their results are shown in Section 4. Section 5 presents the state of the art related to NED tasks and infoboxes generation. Finally, Section 6 provides our conclusions and comments future work

2 Methodology

In this section, we explain the methodology we propose to generate a suitable set of infoboxes out from the information contained in a data silo of text-based documents. Before starting, is important to mention that we assume that a data silo usually is accompanied with one or several documentary information stores about the data silo itself, which can be used to categorize the documents in the data silo. We will refer to it as a set of *local knowledge bases*.

As we can see in Figure 1, the methodology is comprised of four main steps:

1. *Defining Typologies and Templates*: In this first step, the set of templates for generating infoboxes are made. These templates will be organized in a taxonomy, and reflect the classification of the entities that might be found in the texts stored within the silo.
2. *Pre-processing of the Data Silo*: A pre-processing of the documents of the data silo is required to obtain a representation of each of the documents which will help in the subsequent disambiguation process.

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

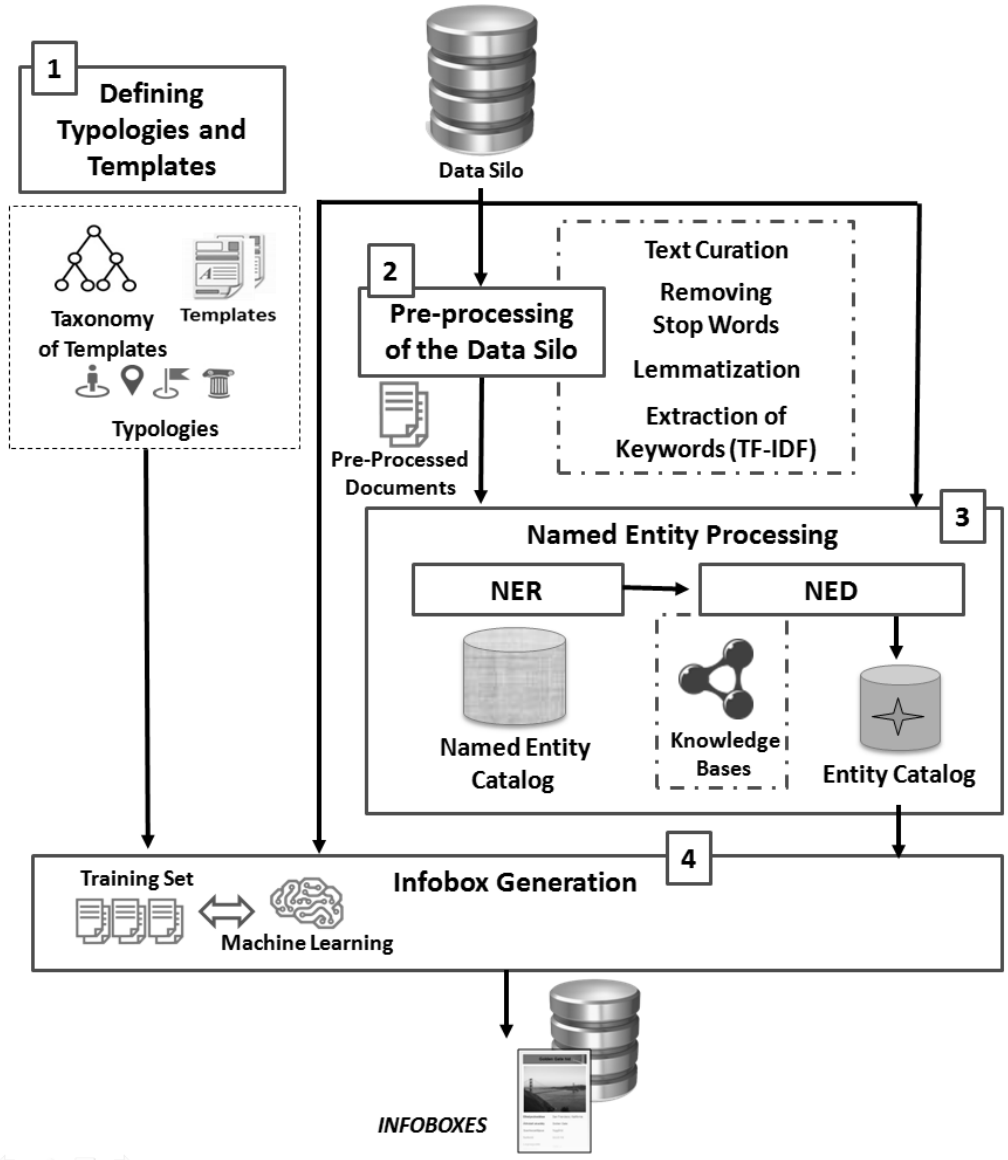

Fig. 1: Overview of the infobox generation methodology.

3. *Named Entity Processing*: The set of representative entities in the data silo are obtained by applying *Named Entity Recognition* (NER) and *Disambiguation* (NED) processes. To do so, we leverage the representation of the documents, the local knowledge bases, and the Semantic Web, with the aim of generating an Entity Cat-

alog (i.e., a repository of all the entities that appear in the data silo associated to their named entity form).
4. *Infobox Generation*: Finally, once the Entity Catalog has been obtained, and the typology of each entity has been recognized for applying the most appropriate template, the infoboxes can be filled with data.

As we will see, each of the steps can be implemented using different techniques, which would have to be selected according to the particularities of each data silo. In the rest of the section, we detail each of these steps, providing different alternatives for each of them.

2.1 Defining Typologies and Templates

The first step in our methodology is to define the general templates that will guide the infobox generation. As above mentioned, and without loss of generality, we assume that there are also one or several documentary information stores about the data silo itself. This information can vary from something very basic (e.g., a list of entities), to something more complex (e.g., a taxonomy, a thesaurus, or even an ontology). This data, which we refer to as *local knowledge bases*, should be a good starting point to perform this task regardless the final modeling approach adopted.

To accomplish the definition of the typologies and templates, we have to establish the following aspects:

- *The nature of the entities to be shown using infoboxes.* Each data silo may store different several entity types, and we have to define them in a general way. A general proposal would be to define an initial set containing templates for people, places, works, events, and general topics. However, it may be required to be extended to be adapted to the particular data silo domain. For example, if the data silo contains information about literature, it might be interesting to have an additional specific infobox for books and other one for authors.
- *A specific taxonomy of entities.* Once the general categories are defined, it may be also interesting to distinguish a hierarchy of specific types and sub-types, which can own specific properties. Following the example above, within the book domain, it may be interesting to distinguish between novels and scientific essays, since the former ones may have their main character as an interesting property, while the latter ones lack them.
- *Attributes for each category.* Each of the different typologies will be defined by having a set of own characteristics, although they can share or inherit others. In the running example, if we talked about contemporary authors, it might be interesting to have information about their social networks; however, this property would not be shared by classical authors, as it would not make sense. This leads to different infoboxes for both typologies, although they would share many other common aspects, such as name, date of birth, or nationality.

Consequently, each of the different typologies will be related to a template formed by a set of fields. Each of the fields will correspond to a property, whose domain and range must be defined as well. Furthermore, the properties can also be defined as single-

or multi-valued ones. The result of this definition step is a set of templates related to each other through a taxonomy.

This step can be done manually by documentation experts, or automatically through different approaches of greater or lesser complexity [8–10].

2.2 Pre-processing of the Data Silo

After having defined the templates, the next step is to obtain an enhanced representation of each of the documents in the data silo, more appropriate to our needs. The representation of documents is fundamental in any natural language process, and in our case it will be a fundamental tool in order to disambiguate the entities contained in it, as well as to identify its typology.

At this point, several different representations of the texts can be used. The purpose of document representation is the translation of documents into system terms, as well as providing them with a manageable structure. This representation is carried out through a set of tasks that contribute through certain simplifications and generalizations to present a logical view of the documents, which makes it possible to consult them posing different querying formalisms.

In particular, we advocate for adopting the *Bag of Words* (BOW) model, a classic approach where the model is formed by a set of simple terms (i.e., words) obtained directly from the documents, regardless of the order. This method considers simple words directly as indexing terms, assuming the correspondence between the terms and the concepts they represent. Obviously, it is not expected that the concepts of a document can be expressed explicitly, but implicitly. The simplicity of this model comes at the price of leaving aside some important elements such as the semantic dependence between terms, syntactic relations and other rules that govern natural language. However, accepting these limitations, the results are usually quite satisfactory, and is computationally inexpensive compared with other more complex methods, as for example POS-tagging, semantic analysis, or n-grams.

We also advocate for enhancing the use of BOW representation by using the following techniques:

- *Text Curation*: This process is devoted to clean the documents by using different techniques in order to, on the one hand, remove irrelevant information, and on the other hand, to correct misspelled words. This action can be very useful when the documents in the data silo have been captured using an OCR system. This process also includes the elimination of punctuation symbols, strange characters, and formulas.
- *Removing Stop Words*: Stop words are basically a set of commonly used words in any language, with no special meaning in the context. Therefore, since we have proposed to represent the documents as BOWs, its elimination will reduce the amount of possible elements, removing those which are very frequent and do not provide any semantic information. Examples of stop words can be articles, prepositions, determinants, etc. Although there are sets of standard stop words for each language, it will be necessary to adapt the set to the exact context of the data silo we are working on.

– *Lemmatization*: This process is focused on obtaining the canonical form (*lemma*), of the words from the texts with the aim of simplifying them. For example, in English, *write, writes, wrote, written, and writing* are forms of the same lexeme, with *write* as the lemma. This process is specially useful for languages with declensions and a lot of verbal forms, such as French, German, or Spanish.

Finally, with the aim of enhancing the representation of the documents, the most relevant words (*keywords*) in the text are calculated using the well-known TF-IDF algorithm [11]. These keyword sets will conform the BOW representation of each document in the data silo for some of the processes explained in the following sections.

2.3 Named Entity Processing

Before going into the details of this step, we have to make an important remark about *named entities* and *entities* in our approach. We consider that a *named entity* is the textual representation of an *entity* in a text, that is, a particular entity[13] can have associated several *named entities*.

Thus, after having pre-processing the documents in the data silo, we propose to recognize and disambiguate all the *named entities* contained in the documents. These two processes are known as *Named Entity Recognition* (*NER*), and *Named Entity Disambiguation* (*NED*). NER is a subtask of information extraction whose purpose is the search and classify of the named entities belonging a document into a set of previously defined categories. On the other hand, NED is a task devoted to determine the identity of those entities previously recognized by a NER step (i.e., associating the recognized *named entity* to a particular *entity*). The main objective of applying this step is to identify all the potential entities that appear in each document, to build a private catalog of *entities*. This catalog will be used further ahead as the closed set of entities which the infoboxes must be generated for.

Named Entity Recognition (NER)

Thus, we need to find all the named entities in the set of documents, because they correspond to these entities. Then, knowing the named entity and its typology, we will be able to orient and limit the generation of infoboxes associated with the data silo. To do so, firstly, a scan of the whole set of documents must be carried out through a NER tool, a natural language processing software able to recognize named entities (Figure 1, Step 3a). At the end of this subtask, for each document in the data silo, a list of recognized entities is obtained. This is a purely lexical catalog of *named entities*, and the correspondences of its elements to *real entities* can be very ambiguous. For example, the word *"Washington"* may refer to the city, to the state, to a sports team, or to the surname of a person. While other approaches stop at this NER step, we propose to go further and disambiguate the retrieved named entities to establish unambiguously their exact entity.

[13] In our context, an entity is an instance of a typology.

Named Entity Disambiguation (NED)

After this first step of NER, it is necessary another step devoted to disambiguation (Figure1, Step 3b). When ambiguity exists, it is necessary to choose the most appropriate entity. However, to achieve this goal, sometimes it is necessary extra information outside the context of the document. We can leverage the local knowledge bases for this, but we should introduce here a new source of resources: the Semantic Web. Nowadays, in the World Wide Web, we can easily find specific and open databases, accessible through web protocols. One of the Web data representation standards that best fits our needs for structured information is the Linked Data format. A huge amount of open knowledge bases can be freely accessed for querying generic information, and they can help us to disambiguate a named entity when the data silo does not store enough information for it.

As we can see in Figure 1, at this point we have three main sources of information: 1) the taxonomy of the types of the entities and their attributes; 2) a set of local knowledge bases with information about the entities appearing in the documents of the data silo; and 3) the text documents themselves along with their enhanced representation.

We advocate for structuring the NED task in three different steps:

1. *Calculating the Context Vector*: The input of the NED task is a named entity, and the document where it is located. The context represented by the words of the document let us to select the entity referenced by the named entity. For each named entity detected in a text, we check if it already exists in the named entity catalog. In case it does not, all the information related with the named entity is searched in the local and local knowledge sources. In each of these sources, we try 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 or external URIs, depending the type of knowledge bases. Then, with the most relevant words of the document where the named entity appears, the Bag of Words (BOW) is calculated and a context vector is generated with weights previously calculated by TF-IDF algorithms for each word. This vector must be compared with the context vector generated by the same procedure for each candidate.
2. *Searching for Entities*: The entities that have the current named entity as a named entity associated, are obtained from the Entity Catalog. For each candidate entity, the context vector of each candidate is generated. This vector will be compared to the context vector of the named entity, and the entity with the highest similarity will be selected. This is done whenever it is fulfilled that its similarity exceeds a given threshold, and then the process will finish. In case none accomplishes this condition, the process will continue the disambiguation comparing the context of the entities of the local knowledge bases that could represent the entity. If we still can not find a suitable candidate, it is time to turn to the Web. For our purposes, any external knowledge base located on the Web cannot be used. We require that it meets a number of specific characteristics: 1) its contents must be related to the data silo; 2) each entity must have a set of related named entities; and 3) An entity available in such knowledge base must have an Uniform Resource Identifier

(URI)[14] and a description of a certain extent, or failing links to HTML documents with this description.
Assuming that the selected external knowledge bases meet these criteria, the process is as follows: firstly the named entity is searched on each of the external knowledge bases. When we found a match, 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. After that, for each URI in the candidate set, the description of the entity is obtained and we extract 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, the URI with the highest similarity will be selected if it exceeds the given threshold. The next step is to check if any entity from the catalog has this URI associated, in which case that entity will be selected.

3. *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, or URI from the external knowledge base. 2) No entity has been selected: 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, the named entity will be discarded.

As it has been seen, the purpose of this process is to obtain a catalog of unique and unambiguous entities. Applying this process to all the named entities of the local repository, the *Entity Catalog* is built. The Entity Catalog generated will be used further ahead as the input to perform the *slot filling task* needed to create infoboxes, which is explained below. It is important to mention that the most relevant words of each entity, and the most outstanding documents related to an entity (i.e. those that have a higher number of words matching the aforementioned set of most relevant words), are stored together with the entity in the catalog. These resources are saved for being used both in this step, and again in the next step.

2.4 Infobox Generation

Once the entity has been detected and disambiguated, its typology must be identified. This step allows to select the correct template in order to generate an infobox with the most relevant information about the detected entity. Besides, the templates of each typology are also already defined.

To achieve this association between entities and templates we can apply different techniques. One option is linking the templates manually. In that case, an expert reads the documents that reference that entity, and he/she determines which template previously defined is the most accurate. This is the ideal process, but it has the drawback of being very expensive and time-consuming.

[14] URI is a string of characters used to identify a resource in the context of the World Wide Web.

Another option is to identify the most appropriate template using an automatic method. To do so, our approach is to try to identify the typology of an entity by studying the characteristics of the contexts of the documents where that entity appears. Assuming that we own models (i.e., documents where those typologies of entities appear, and that have been manually identified by experts for each of the topologies), we could apply supervised learning algorithms to achieve it.

Supervised learning algorithms need some attributes called features in order to generate the model. The BOW model explained in Section2.2 can be used for this purpose. The aforementioned stop words elimination and the lemmatization process help to reduce the feature space.

Finally, the generation of the infoboxes (known as *slot filling* task) can be performed in two different ways: 1) the expert users fill directly the attributes, which is a time-consuming task, only recommended for isolated cases, for example when the data silo does not have a large number of entities and they are specially local to the domain of the documents; or 2) the slots of the attributes are filled using automatic processes, depending on the classification of the entity and the typology of the attribute (number, string, date, etc.). The allocation of procedures for each infobox attribute can be a manual task made by the information extraction specialist of the computer department, or again, it could be an automatic task. Both options are out of the scope of this work.

3 Implementing the Methodology

With the aim of testing our methodology, we have developed a system that meets the requirements specified in Section 2. Leveraging our own work, we have evolved NEREA [7], a system which is able not only to recognize relevant entities from a text in a local document database, but also to disambiguate them, thanks to a subsystem denominated *POIROT*, in honor of the famous detective of Agatha Christie's books. We first overview the system as a whole, and then we present the different subsystems that implement the proposed methodology.

3.1 Overview of the system

Our system follows step by step the exposed work methodology, as it can be seen in Figure 2:

- We have developed a tool called *SOPHIE* to assist the expert users of the system on preparing the templates of the infoboxes, as described in Section 2.1.
- NEREA processes the data silo as explained in Section 2.2, in order to obtain a enhanced BOW representation of each document, without stop words, and lemmatized. NEREA is also the responsible for generating a lexical catalog of named entities out from the local database to store all the potential candidates from each source of information, as considered in Section 2.3.
- POIROT chooses the most appropriate entities given a document and a set of named entities, leveraging the information of different sources in order to provide the best result. As suggested in our proposed methodology (see Section 2.3), POIROT now

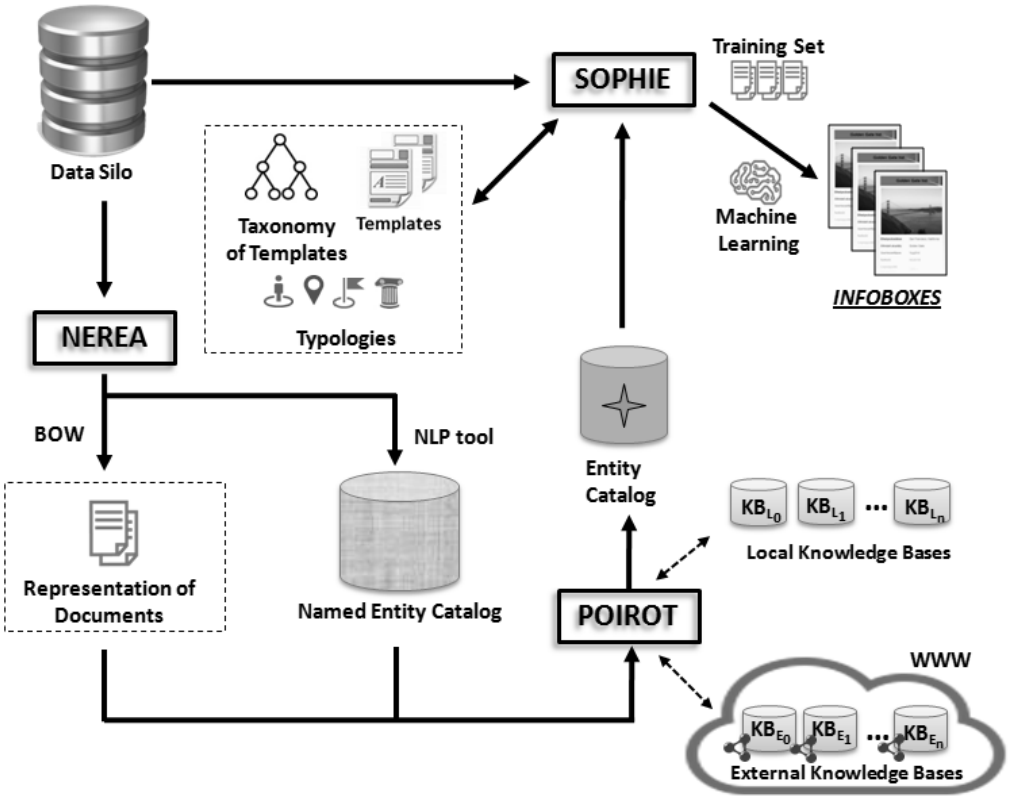

Fig. 2: Overview of the system that implements the methodology, composed by three sub-sytems: SOPHIE (defining typologies and templates managing), NEREA (pre-processing and NER), and POIROT (NED).

considers the possibility of having a set of local knowledge bases, and a set of external knowledge bases located in the web. The result of this step is an *Entity Catalog* that contains a set of disambiguated relevant entities within the local repository.

– Finally, SOPHIE is also in charge of assigning the most accurate template to each entity by leveraging machine learning techniques, as explained in the methodology in Section 2.4, before the final generation of the infoboxes.

In the rest of the section, we focus on SOPHIE and POIROT as they are the new and improved elements of our system. For space reasons, we refer the interested reader to [7] for further details on NEREA, the third subsystem in the implementation.

3.2 SOPHIE Subsystem

As the aim is to work in the specific domain of a data silo, we have designed a tool called SOPHIE, targeted to expert users, whose purpose is to create the templates for the infoboxes. These templates are important elements to support the process of extracting information from the documents, as 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. Thus, a template is defined by a name, a category, 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). These attributes are linked to specific methods for searching the information to extract given a set of knowledge bases (local or external). These methods are functional elements of information extraction, which are linked to those characteristics that are desired to obtain. Examples and further details of these kind of methods can be found in works as [12, 13].

Moreover, SOPHIE aids the user with a user-friendly GUI, that provides information from the local knowledge base when necessary. For example, if a *person* is defined in a supposed local ontology, when the user wants to create a specific template for a group of people (for example, *athletes*), this tool would prepare an automatic template based on the characteristics and the attributes of the concept of *person* in that ontology.

In our methodology, once an entity has been detected and disambiguated, its typology should be identified in order to select the correct template to show the information. In this case, SOPHIE categorizes the entity comparing the most relevant documents where that entity appears with a set of model documents for each of the typologies, and choosing the typology with most similar documents.

To do so, SOPHIE uses support vector machines (SVM) [14], a supervising learning model. The rationale behind this selection is: 1) SVMs are fast and simple, 2) they can extract optimal solutions with a small training set size, and 3) using Kernel Functions, SVMs are applicable in circumstances that other models like Bayes or Neural Networks do not behave right, such when dealing with randomly scattered data or when the distribution of the data is not well defined. In addition, our previous experience with this tool has demonstrated several times [15–17] its good performance in this type of scenarios.

As the templates are hierarchically defined (for example, a politician is a kind of person), there are several SVM models to determinate the template. SOPHIE uses binary classifiers in a tree form to determinate its typology in each level of this hierarchy. This step of the system uses the same pre-processing text as a feature for the SVM.

Therefore, SOPHIE, by using the Entity catalog created by the subsystem POIROT (see Section 3.3), helps the expert staff in preparing the templates for the infoboxes generation. A screen-shot of this software can be seen in Figure 3.

3.3 POIROT Subsystem

One of the most interesting points of the methodology proposed is the stage of entity disambiguation. In our implementation, this task is relied on POIROT subsystem.

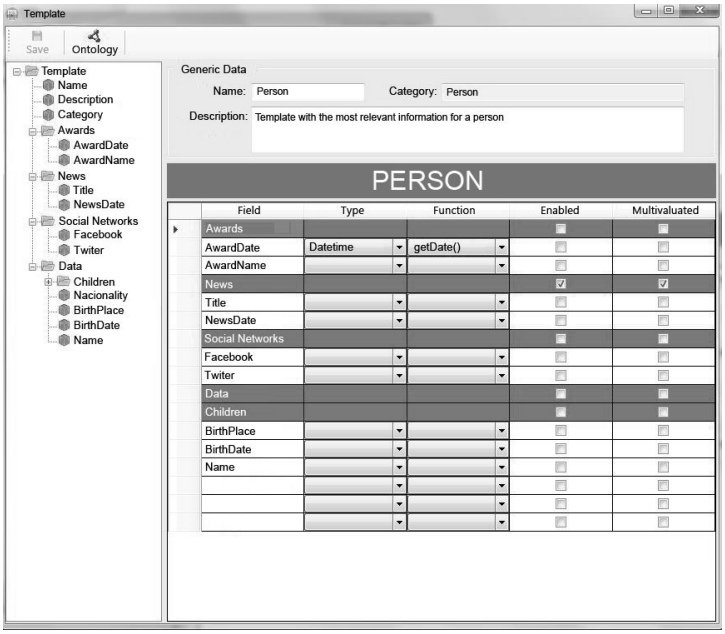

Fig. 3: Screenshot of SOPHIE, the software for definition of typologies and creation of templates. This module is also responsible for the the identification of the typologies of the discovered entities.

POIROT receives a named entity and the document where it appears as input, and queries sequentially the different available knowledge bases (namely, the Entity Catalog, the local knowledge bases, and external knowledge bases) looking for candidate entities which could be referred by such input named entity. For each of the candidates found, POIROT builds a context vector which is used to compare the candidate with the context (i.e., the representation of the input document) where the named entity appears in order to select the most probable interpretation. This comparison is performed using the cosine similarity of the contexts. The cosine similarity is a common method used to measure the similarity using the BOW model. For measuring the similarity between two BOWs the method is to create context vectors for each BOW and then calculate the cosine similarity between the vectors according to the following formula:

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

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

The complete sequence used for disambiguating is described in Algorithm 1.

Algorithm 1 Disambiguation Algorithm of a named entity, given a context and a set of resources.

function DISAMBIGUATION(NamedEntity NE, Context C, ResourceList RL) as Entity
  $V_{context} = createContextVector(C)$
  $maxSimilarityValue = 0$
  for each $candidate$ in $Candidates(RL, NE)$ **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

While the Entity Catalog and the local knowledge bases are under our control and design, external knowledge bases have to meet some criteria to be handled by POIROT (see Section 2.3). In particular, in our current implementation, we have added DBpedia [18] as an external knowledge base. DBpedia is a knowledge base that contains a vast amount of data which is obtained by extracting structured information from Wikipedia. In particular, we have used DBpedia Spotlight[15], a tool for automatically annotating mentions of DBpedia resources in text. This collection has a set of named entities related with some pre-calculated DBpedia URIs.

Thus, POIROT checks if the named entity is in the DBpedia Spotlight *pair count* collection. This collection contains pair-wise occurrence counts that keep track of anchor texts in DBpedia articles, storing the different named entities that appear linked to a DBpedia article, and the count of the number of times this link occurs. Then, our system adds the URIs assigned to the named entity in this collection to the set of candidate entities. Besides, POIROT exploits the information contained in the DBpedia disambiguation page to further expand the candidate set, i.e., 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 the external knowledge base where the entity has got a textual description and extracts a BOW from it, which is used to generate context vectors for each candidate.

The URI with the highest similarity will be selected if it exceeds a given threshold. In this step, exceptionally, if no URI has exceeded the similarity threshold, POIROT chooses the URI that has the greatest count in the DBpedia Spotlight pair count collection (i.e., when the context is not sufficient to choose a particular interpretation, we rely on the information provided by the DBpedia Spotlight pair count collection). The next

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

step is to check if any entity from the catalog has this URI associated, in which case that entity will be selected.

For each identifier of the set of candidates of the knowledge bases, 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 knowledge base identifier of the text with the highest similarity will be selected, as long as its similarity value exceeds the similarity threshold.

4 Experimental Evaluation

In this section, we describe the datasets and the experimental tests that we have performed to check our methodology through the system described in Section 3.

To our knowledge, there is not any standard dataset for testing the problem we have faced. Hence, we have evaluated the problem by dividing it into two different stages, but taking into account the influence of the first stage over the second one:

1. As our methodology heavily relies on correctly choosing and detecting the entities in the data silo documents, we have evaluated firstly the NED performance of our system. This task is regarded in the literature as *entity linking*.
2. For the infobox generation, we evaluated the ability to complete a template with data from a set of natural language based documents. This is commonly known as a *slot filling task*.

For testing the NED subsystem, we have used *GERBIL 1.2.4*[16], a general entity annotation system [19], and two different datasets. The evaluation of the slot filling task led us to search for a use case where a data silo was involved. We found the news domain very appropriate. Media companies usually own data silos of news intended for the particular use of their journalists. These data silo usually are complemented with local knowledge bases prepared by the documentation department. The generation of infoboxes from these databases is a very useful tool, especially if these infoboxes can be enriched with information from the Web. In particular, we have worked in collaboration with HENNEO[17], a major Spanish media company, which has given us access to its historical archive of news to perform our experimental tests.

In the rest of the section, we firstly present the experiments performed for evaluating the NED task describe and the datasets used. Then, we move onto the complete evaluation of the system in a real environment, interpreting the results of the different experiments, and we provide some conclusions about them.

4.1 Entity linking evaluation

Firstly, as its results are crucial for our approach, we have evaluated the quality of the *entity linking* that our system performs. In general, the input for this task is a text

[16] http://gerbil.aksw.org/gerbil/
[17] http://henneo.com/

document with a set of entities which should be matched in a given database. As stated before, this disambiguation task has been evaluated comparing to *GERBIL*. This tool is very useful for our purposes because it provides an unbiased and fully automated evaluation platform in order to compare our system to other systems regarding the task of entity disambiguation.

Datasets We have tested the latest version of NEREA and POIROT over two different datasets:

1. *Reuters-N3-128 dataset*[18]: This English corpus is based on the well known Reuters-21578 corpus which contains economic news articles. In particular, the authors [20] chose 128 articles containing at least one named entity.
2. *OKE 2015 dataset*[19]: It 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 for people, locations, organizations, and roles.

In summary, both datasets are a set of texts with additional information about the entities that are included in them along with its pointer to DBpedia.

Experimental Results First, we have to analyze some of the parameters of our system, i.e., the context vector size and the similarity threshold. Regarding the context vector size used to disambiguate the entities, we have observed that there is not a fixed size for optimal results, as it strongly depends on the amount of relevant information available in the sources (i.e., the amount of relevant words that we can leverage to build a relevant BOW representation).

We studied the combined influence of the length of the context vector and the similarity threshold, and we saw that, for these datasets, the optimal value for the context vector was between 50 and 120 words. We have fixed its value to 50 words in order to have not too much context information in the experiments. Once fixed, in Figure 4, the change in the F-measure (F1%) depending on the similarity threshold can be observed for the OKE 2015 dataset. The best system performance is obtained with a threshold value of 0.59. Due to weak contextual information present in that dataset, this figure is very similar to the one obtained with context vector sizes from 50 to 120. However, in the case of the Reuters-N3-128 dataset, the best results are achieved with a threshold value of 0.11 (see Figure 5).

This difference in the similarity threshold is caused by the nature of the documents of the data set. In the OKE 2015 dataset, we can find 301 entities which are all of them linked to a DBpedia entity. However, in the Reuters-N3-128 dataset, there are 880 entities, 650 of which are linked to a DBpedia entity (i.e., 230 entities are not linked to its DBpedia resource). Besides, the texts of OKE 2015 dataset are rather shorter than the news in Reuters-N3-128 dataset, because they are formed by only one sentence.

[18] https://github.com/AKSW/n3-collection/blob/master/Reuters-128.ttl
[19] https://github.com/anuzzolese/oke-challenge

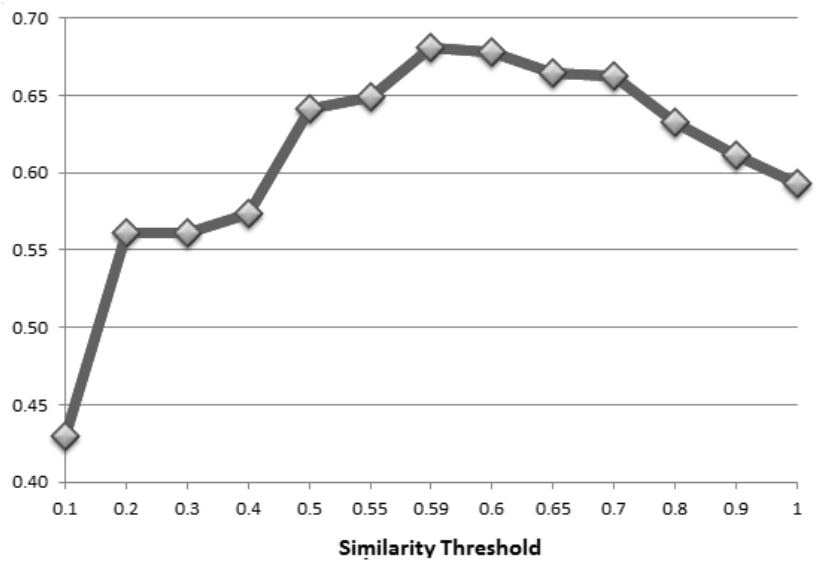

Fig. 4: Influence of the similarity threshold in the F1 values using the OKE 2015 dataset (context vectors length set to 50) [7]. The peak value of the F1 is on 0.59.

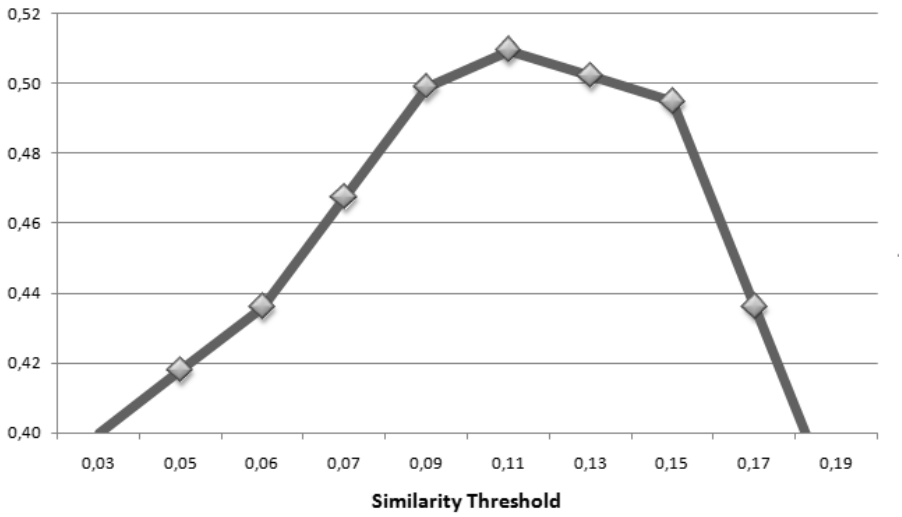

Fig. 5: Influence of the similarity threshold in the F1 values using the Reuters-N3-128 dataset (context vectors length set to 50). The peak value of the F1 is on 0.11.

In fact, our BOW representation of each document in the OKE 2015 dataset is practically the whole document, offering a set of very suitable words. On the other hand,

with the Reuters dataset, it is better to choose the proposed entities even if the degree of similarity is low, because the vocabulary, being more extensive, is more dispersed.

We compared the results of our system in this task to the results achieved by other systems using GERBIL as framework. We have selected the experiment type *ADKB*[20]. The results are shown in Table 1, in terms of micro F-measure (mF1%), micro precision (mP%), and micro recall (mR%)

Table 1: Results of ADKB task reported by GERBIL using several annotators on the Reuters-N3-128 and the OKE 2015 datasets.

Annotator	Reuters-N3-128 dataset			OKE 2015 dataset		
	mF1%	mP%	mR%	mF1%	mP%	mR%
AIDA	46.59	65.57	36.14	53.73	67.97	44.43
Babelfly	45.29	63.39	35.23	56.21	68.32	47.74
DBpedia Spotlight	33.09	56.44	23.41	30.75	45.06	23.34
Dexter	36.05	75.91	23.64	46.17	**87.76**	31.33
Entityclassifier.eu NER	37.04	44.77	31.59	32.10	58.33	22.14
FRED	40.64	56.97	31.59	44.94	57.82	36.75
NERD-ML	41.10	62.70	30.57	59.15	79.80	46.99
xLISA	38.65	81.39	25.34	60.23	81.12	47.89
WAT	50.36	**82.43**	36.25	55.26	70.40	45.48
NEREA & POIROT	**50.95**	48.51	**55.50**	**68.00**	79.40	**59.50**

We can see how our approach, while not achieve the best results in terms of precision (achieved by WAT [21] and Dexter [22] systems for Reuters and OKE 2015 datasets respectively), it improves the micro F1-measure and the recall on both datasets, clearly outperforming other annotators when dealing with the OKE 2015 dataset.

4.2 Evaluation with a real data silo

Apart from evaluating the NED process by itself, we have evaluated the whole system that implements our proposed methodology thanks to the collaboration of HENNEO[21], a major Spanish media company. In particular, we have worked with the documentary archive of the newspaper *Heraldo de Aragón*[22], a clear sample of a data silo devoted exclusively to be used by the company workers. The news archive contains information published from the year 1895 to the present. This archive contains more than a million of news, and it has several local knowledge bases with information about the data silo itself: a thesaurus with a taxonomy of the entities, a geographic database [23], and an ontology that details the relations between entities.

[20] The task ADKB in GERBIL consists of, given a text, recognizing the named entities contained in that text, and linking them to a knowledge base.

[21] http://henneo.com/

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

The main themes of this data silo are local politics and local sports. This is a problem to perform disambiguation tasks, because in many cases the characters and locations that appear in the news do not exist on the Web, or they do not have a web page published on Wikipedia. As a result, our system does no have external sources which to extract information from. Hence, it is very important for our system to use its local knowledge bases 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 their 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 cases, they will not be referenced in Wikipedia or 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.

The dataset we used consists of 5,000 news from years 2014 and 2015 written in Spanish. We have forced the presence of particularly ambiguous entities, that is the case of several famous people surnamed "Iglesias", including singers, politicians, and businessmen. The documentation department had annotated previously these news with the correct entities.

NED evaluation with NEREA and POIROT We have evaluated the disambiguation task, corresponding to perform the first three steps of the methodology. Firstly, we used NEREA system for pre-processing the data silo and obtaining the named entities catalog, with the help of a natural language tool called Freeling[23]. Then, we performed a first test (*Experiment 1* in Table 2) carrying out the disambiguation process *without* using the local knowledge bases. We did it using a downgraded version of POIROT (only taking into account the DBpedia Spotlight pair counts collection). Afterwards, we carried out another test (*Experiment 2* in Table 2) with the complete version of the system, leveraging the local knowledge bases of the data silo. The results can be appreciated in Table 2, which shows micro F-measure (mF1%), micro precision (mP%), and micro recall (mR%), using context vectors of length 50 and 0.11 as similarity threshold due to the size of the documents (learned from our previous experiments with the Reuters dataset, see Section 4.1). As expected, the accuracy of the disambiguation process is better when using the local knowledge bases.

Table 2: Named Entity Disambiguation task evaluation on the Heraldo dataset depending on the use of local knowledge bases (LKB).

NEREA	Heraldo dataset		
	mF1%	mP%	mR%
Experiment 1 (without LKB)	55.2	45.3	49.7
Experiment 2 (with LKB)	**83.5**	62.4	71.4

[23] http://nlp.cs.upc.edu/freeling/

Infobox generation evaluation with SOPHIE Finally, we have evaluated the infobox generation, which corresponds to the final step of our methodology. To do so, we started using a controlled set of templates dedicated to obtain information from people: politicians, athletes, musicians, and actors. We selected a set of 30 famous people appearing in the Heraldo dataset. Out of this 30 people, 20 of them are internationally well-known personalities, with information easily accessible in knowledge bases (e.g., DBpedia). The other 10 characters are locally known only in the area of Aragón, the Spanish region whose news covers the newspaper *Heraldo de Aragón*. Thus, it is more complicated to find information about them on the Web. Each of the templates has a set of 15 fields for different properties, including, for example, the surname, date of birth, or nationality of the character. Five of these fields are specific for each typology of template. SOPHIE is a tool for searching information both in the data silo and in the Web from a template. As stated in Section 3.2, the templates include the specific methods for searching and extracting of information from the local knowledge bases and from the Semantic Web. The current upper limit of system performance would be achieved if each character would be assigned its most appropriate template manually. We have carried out manually the process, and a 93% of accuracy [24] in the results would be obtained (100% is not achieved due to the difficulty of extracting the information accurately from external sources).

Regarding the automatic classification of templates, we considered the four aforementioned typologies of templates. For the classification, we used Support vector Machines (SVM) Multiclass with a radial basis function. We trained the classifier with 14,400 news, including 2,400 news related to these four type of templates. The objective is, as explained in Section 3.2, to find the set of model documents more related the entity, for selecting the most suitable template. The results after a 5-fold validation gives an accuracy of 84.65%.

We evaluated the success rate when filling the fields of the 30 infoboxes from the information embedded in the Heraldo dataset for different settings of our system. To show the influence of the different steps of our approach, we have devised different working settings where different capabilities of the system are progressively activated, until the whole process is working.

In each of these work settings, we vary the use of local and external knowledge bases, as well as the classification tool usage. In particular:

1. *Work setting 1:* The Entity Catalog is generated without using the local knowledge bases, and the automatic classification tool for obtaining the most accurate template is not operating. The system only uses a generic template for filling people's information. So, in each template, 33% of the fields are always empty.
2. *Work setting 2:* The Entity Catalog is generated combining the external knowledge bases with local knowledge bases. As in the previous working setting, the system uses only generic templates.
3. *Work setting 3:* The Entity Catalog is generated without using the local knowledge bases, but here the automatic classification tool for obtaining the most accurate template is active.

[24] The accuracy in this context is based on considering the number of attributes correctly obtained ("filled slots") on the total existing.

4. *Work setting 4:* This working setting corresponds to our complete system. The Entity Catalog is generated combining the external knowledge bases with local knowledge bases, and the automatic classification tool for obtaining the most accurate template is active.

Table 3: Accuracy of the infobox generation task on the real data silo of *Heraldo de Aragón* using our approach with different work settings.

SOPHIE	Configuration			Results			
	EKB	LKB	ML	Entities	Templates	Slots	Acc%
Work setting 1	X			16	0	148	0.33
Work setting 2	X	X		25	0	232	0.52
Work setting 3	X		X	16	13	209	0.46
Work setting 4	X	X	X	25	21	330	**0.73**

Table 3 shows the number of correct entities, detected templates, slots filled, and accuracy (Acc%) of SOPHIE retrieving the field values of the 30 characters. The columns *External Knowledgebases* (EKB), *Local Knowledgebases* (LKB), and *Machine Learning Tool* for selecting the template (ML) mark whether the particular element is active in such configuration. Regarding the results, the column *Entities* shows the number of correct entities returned by the NED system, the column *Templates* shows the number of correct templates selected by the ML tool, and the column *Slots* indicate the number of correct slots filled in the information extraction process. As explained before, the best result with an optimal assignment of templates to each entity is 93% of correct slots filled in the infoboxes, considering a total amount of 450 slots.

As regards the disambiguation task, if we only consider the local entities, the performance of our system is outstanding. This is mainly due to the fact that if we work with entities for which there is no information on external knowledge base, the only source of information is our local knowledge. In our real environment, this is the situation of one out of every eight entities in our catalog. Hence, we can say that in 12.5% of cases the systems based on web resources for disambiguating tasks will fail. Thanks to our ability to integrate local knowledge bases we can palliate that problem.

If we compare the system performance between the different datasets, we note that the performance of the system with the data silo dataset is considerably better, because the contextual information available in the Heraldo dataset is much higher than in the Reuters-N3-128, and OKE 2015 evaluation datasets. Furthermore, in the OKE 2015 dataset, the size of the documents is very small with a mean length of just 10 words. That amount of information is usually insufficient for performing an accurate context comparison. When the length of the text exceeds 100 words, the accuracy of our approach when disambiguating an entity is always above 70%. However, we have to remark the difficulty regarding the disambiguation task, which, sometimes, even for humans, it is not even clear which entity is referenced in a text. There can be manifold reasons for this: not enough amount of contextual information, high similarity between two entities, etc. For example, if the context is only the sentence *"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", both famous singers, father and son respectively.

Regarding the generation of infoboxes in the data silo, the results of Table 3 show the great influence of disambiguating and generating a good entity catalog in the quality of the infoboxes (20% more of accuracy), and the good outcomes when combined with the automatic selection of templates using machine learning, reaching a 73% of accuracy. These good results suggest that a system implemented following the methodology described in this work achieves much better results, so they endorse our working thesis.

5 Related Work

In this section, we describe some works related with our proposed methodology. As far as we know, there are no studies devoted to study the generation of infoboxes from a data silo; thus, we focus on analyzing proposals that are relevant to some of the intermediate steps proposed in this work.

5.1 Named Entity Disambiguation

The term *record linkage* [24] appeared in the scientific literature back in 1946. Its mission is to find records that refer to the same entity in different data sources, such as databases, books, etc. The main difference between record linkage and NED tasks is that NED tasks rely on a knowledge-base (e.g., Wikipedia) with a list of gold-standard entities, with textual information describing each one. In the 1970s the problem of word sense disambiguation was faced with approaches based on language understanding [25], but the generalization of results was very hard. Wide lexical resources were released during the 1980s [26] for helping knowledge extraction algorithms, and the use of statistical methods was the typical approach of the 1990s [27]. Finally, approaches that used external knowledge sources containing information about entities (e.g., Wikipedia) appeared in 2003 [28]. Later, in 2006, titles, hyperlinks, and disambiguation pages are used to generate candidate entities [29]. Afterwards, in 2009, a system [30] with a simple text similarity approach for disambiguation was implemented, but the authors gave special attention to the step of generating candidates, using Bayes and K-Nearest Neighbors classifiers.

NEREA and POIROT use a clustering approach, a well-known technique based on grouping similar objects on groups called *clusters*. This approach has been widely used in the past and nowadays [31–33]. Specifically, if we focus on the disambiguation method, it can be classified as a context clustering unsupervised approach [5], because of their representation of occurrences through context vectors, and the use of cosine similarity. The methodology points to the same direction proposed by an inspiring study published in 2013 [34], where several NED systems were compared, and the conclusion was that the stage of search and generation of candidate entities seems to have more impact on the final accuracy than the disambiguation stage.

Regarding recent systems, we can cite AIDA [35], a framework for entity detection and disambiguation. This system map named entities with entities from a given natural-language text or a Web table registered in a knowledge base. Like our proposal, AIDA

exploits the similarity between the context of the mention and its candidates. AIDA also uses the coherence between two entities, saying that it is proportional to the number of incoming links that are shared between their Wikipedia articles. X-LiSA [36]is another relevant system for cross-lingual semantic annotation. X-LiSA supports both service-oriented and user-oriented interfaces for annotating text documents and web pages in different languages using resources from Wikipedia and Linked Open Data (LOD), by leveraging the semantic similarity between entities. Both systems use a graph-based model for disambiguation. Our context clustering model based on BOW is simpler but as we can see in the experiments it is more accurate, and it is also more scalable and faster. In a context devoted to the recognition of entities in a silo data, we consider that an important feature is to be able handle local catalogs of entities for helping the NED system, extending it by exploiting with other Web resources. In this way, the local information is complemented and the NED performance is clearly improved, and it provides great flexibility when applied in different environments.

5.2 Slot Filling and Infoboxes

Information Extraction [37] consists of automatically extracting structured information from natural language documents. The generation of infoboxes from a data silo with text-based documents can be framed in this discipline. Since the appearance of the infoboxes, different methodologies have emerged to automatically fill them, leading a task known as *slot filling*. In the 1990s, the first systems dedicated to this task were mainly based on rules [38]. However, these kind of techniques suffered from a very low performance and, besides, they were very time-consuming as they required to code the extraction rules manually. Then, statistical learning techniques [39] and grammatical construction techniques [40] appeared, and even they were merged [41], bringing together the advantages of both methodologies.

If we focus on works related with the generation of infoboxes from large text-based data repositories like Wikipedia, we find some relevant samples, like *KYLIN* [2], which uses the infoboxes as training data, and then accurately extract triples from Wikipedias natural-language text, combining the use of Wordnet [42] with Markov Logic Networks. Another prominent system is iPopulator [43], which populates infoboxes of Wikipedia articles by extracting attribute values from the articles text using machine learning. Finally, IBminer [44] is a system to derive structured information from Wikipedia using natural language processing. However, none of these systems take into account the importance of disambiguation, they simply do not face the problem because they do not consider it. So, their approaches lack of tools to start from scratch when faced a data silo without any infobox that they can use as training data. Regarding works related with selection of templates, Sultana et al. in [8] show a system able to select the most appropriate template, based on machine learning over existing infoboxes. Again, our methodology is able to start from a earlier step, when there are no infoboxes still created, and we only have a data silo and its associated knowledge bases.

Hence, we have not found special studies that have the goal and broad scope of our approach, proposing a whole work methodology which cover the generation and population of infoboxes from an isolated data silo, and studying the influence of the steps that can have greater impact in its outcome, like the disambiguation of the named

entities or the automatic selection of templates. The utility of these kind of systems is clear for enhancing libraries, private archives, and other data silos, as we proposed in our initial works [45].

6 Conclusions and Future Work

The use of infoboxes for presenting information about different entities is useful as they comprise the most important information about them, allowing the user to check both the existence of the data searched, as well as its validity in a first sight. However, building and populating them is a costly task that must be automatized as much as possible. While there are some previous approaches devoted to such task, its difficulty increases even further when the target of our infobox creation are unstructured texts of a given data silo, which information is prone to be completely local (i.e., there are no general knowledge bases on the Web that are likely to have information about them).

In this work, we have tackled the generation of infoboxes in this compelling scenario, where the flexibility to handle and integrate different information resources (i.e., local and external knowledge bases) is shown to be of capital importance. The main contributions of this work are as follows:

- We have proposed a methodology based on the detection and disambiguation of the entities in the data silo in order to make it feasible to both select the appropriate infoboxes, and to populate them.
- We have validated our proposal by implementing such methodology developing and integrating several systems dedicated to different tasks, which has led to a complete pipeline for dealing with data silos. In this integration, in particular, we have improved NEREA and POIROT to handle and integrate local knowledge bases with external ones.
- We have included the capability of selecting automatically the most suitable template for generating the infobox leveraging machine learning techniques within SOPHIE subsystem. This allows to further improve the generation of the infoboxes.
- Last but not least, we have tested thoroughly the main different steps of our proposal using well-known and publicly available datasets, as well as in the real scenario of a news media company, dealing with their local data silo.

There are several lines of development for future work, but for us the main one is to continue advancing in the complete automation of the different steps where the skilled personnel presence is still required. In particular, we plan to work on the automatic design of the templates out from the local knowledge bases, and on the automation of the development of the methods of extraction of information.

Acknowledgments

This research work has been supported by the CICYT project TIN2013-46238-C4-4-R, TIN2016-78011-C4-3-R (AEI/FEDER, UE), and DGA/FEDER. We want to thank HENNEO for their collaboration in different stages of the project, to Estela Garrido, and specially to Dr. Carlos Bobed for his collaboration in the writing of this work, his advice, and his priceless help.

References

1. F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from wikipedia and wordnet," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 3, pp. 203–217, 2008.
2. F. Wu, R. Hoffmann, and D. S. Weld, "Information Extraction from Wikipedia: Moving down the long tail," in *14th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 731–739, ACM, 2008.
3. D. Downey, O. Etzioni, S. Soderland, and D. S. Weld, "Learning text patterns for web information extraction and assessment," in *Workshop on Adaptive Text Extraction and Mining (ATEM)*, pp. 50–55, 2004.
4. W. W. Cohen, H. Kautz, and D. McAllester, "Hardening soft information sources," in *6th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 255–259, ACM, 2000.
5. R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.
6. 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 (WWW)*, pp. 643–648, ACM, 2015.
7. A. L. Garrido, S. Ilarri, S. Sangiao, A. Gañán, A. Bean, and Ó. Cardiel, "NEREA: Named entity recognition and disambiguation exploiting local document repositories," in *28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1035–1042, IEEE, 2016.
8. A. Sultana, Q. M. Hasan, A. K. Biswas, S. Das, H. Rahman, C. Ding, and C. Li, "Infobox suggestion for Wikipedia Entities," in *21st International Conference on Information and Knowledge Management (CIKM)*, pp. 2307–2310, ACM, 2012.
9. R. Yus, V. Mulwad, T. Finin, and E. Mena, "Infoboxer: using statistical and semantic knowledge to help create Wikipedia infoboxes," in *13th International Semantic Web Conference (ISWC)*, vol. 1272, pp. 405–408, CEUR-WS. org, 2014.
10. N. F. Rajani and R. J. Mooney, "Combining supervised and unsupervised ensembles for knowledge base population," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
11. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.
12. M. G. Buey, A. L. Garrido, C. Bobed, and S. Ilarri, "The AIS project: Boosting Information Extraction from Legal Documents by using Ontologies," in *8th International Conference on Agents and Artificial Intelligence (ICAART)*, pp. 438–445, SCITEPRESS, 2016.
13. A. L. Garrido, M. G. Buey, G. Muñoz, and J.-L. Casado-Rubio, "Information Extraction on Weather Forecasts with Semantic Technologies," in *International Conference on Applications of Natural Language to Information Systems (NLDB)*, pp. 140–151, Springer International Publishing, 2016.
14. T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Springer, 1998.
15. A. L. Garrido, O. Gomez, S. Ilarri, and E. Mena, "NASS: News Annotation Semantic System," in *23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 904–905, IEEE, 2011.
16. A. L. Garrido, O. Gómez, S. Ilarri, and E. Mena, "An experience developing a semantic annotation system in a media group," in *17th International Conference on Applications of Natural Language Processing to Information Systems (NLDB)*, pp. 333–338, Springer Berlin/Heidelberg, 2012.

17. A. L. Garrido, M. G. Buey, S. Escudero, A. Peiro, S. Ilarri, and E. Mena, "The GENIE project-a semantic pipeline for automatic document categorisation.," in *10th Conference on Web Information Systems and Technologies (WEBIST)*, pp. 161–171, 2014.
18. 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*, pp. 722–735, Springer, 2007.
19. R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, *et al.*, "GERBIL: General Entity Annotator Benchmarking Framework," in *24th International Conference on World Wide Web (WWW)*, pp. 1133–1143, ACM, 2015.
20. M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N^3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format.," in *9th Conference on Language Resources and Evaluation (LREC)*, pp. 3529–3533, 2014.
21. F. Piccinno and P. Ferragina, "From TagME to WAT: a new entity annotator," in *1st International Workshop on Entity recognition & Disambiguation*, pp. 55–62, ACM, 2014.
22. D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani, "Dexter: an open source framework for entity linking," in *6th International Workshop on Exploiting Semantic Annotations in Information Retrieval*, pp. 17–20, ACM, 2013.
23. A. L. Garrido, M. G. Buey, S. Ilarri, and E. Mena, "GEO-NASS: A semantic tagging experience from geographical data on the media," in *17th East European Conference on Advances in Databases and Information Systems (ADBIS)*, pp. 56–69, Springer, 2013.
24. H. L. Dunn, "Record linkage*," *American Journal of Public Health and the Nations Health*, vol. 36, no. 12, pp. 1412–1416, 1946.
25. Y. Wilks, "Preference Semantics," *Formal Semantics of Natural Language*, pp. 329–348, 1975.
26. Y. Wilks and B. M. Slator, "Towards semantic structures from dictionary entries," in *2nd Annual Rocky Mountain Conference on AI*, pp. 85–96, 1989.
27. J. Veronis and N. M. Ide, "Word sense disambiguation with very large neural networks extracted from machine readable dictionaries," in *13th Conference on Computational Linguistics (ACL)*, pp. 389–394, Association for Computational Linguistics, 1990.
28. 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 (WWW)*, pp. 178–186, ACM, 2003.
29. R. C. Bunescu and M. Pasca, "Using Encyclopedic Knowledge for Named Entity Disambiguation.," *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, vol. 6, pp. 9–16, 2006.
30. 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.
31. V. Filkov and S. Skiena, "Integrating microarray data by consensus clustering," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 04, pp. 863–880, 2004.
32. M.-A. Rizoiu, J. Velcin, and S. Lallich, "How to use temporal-driven constrained clustering to detect typical evolutions," *International Journal on Artificial Intelligence Tools*, vol. 23, no. 04, p. 1460013, 2014.
33. T. Theodorou, I. Mporas, A. Lazaridis, and N. Fakotakis, "Data-driven audio feature space clustering for automatic sound recognition in radio broadcast news," *International Journal on Artificial Intelligence Tools*, vol. 26, no. 02, p. 1750005, 2017.
34. 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.

35. M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum, "Aida: An online tool for accurate disambiguation of named entities in text and tables," *Proceedings of the VLDB Endowment*, vol. 4, no. 12, pp. 1450–1453, 2011.
36. L. Zhang and A. Rettinger, "X-LiSA: cross-lingual semantic annotation," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1693–1696, 2014.
37. S. Russell, P. Norvig, and A. Intelligence, "Artificial Intelligence: A Modern Approach," *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, vol. 25, 1995.
38. E. Riloff *et al.*, "Automatically Constructing a Dictionary for Information Extraction Tasks," in *11th National Conference on Artificial Intelligence (AAAI)*, pp. 811–816, 1993.
39. P. Viola and M. Narasimhan, "Learning to Extract Information from Semi-structured Text using a Discriminative Context Free Grammar," in *28th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 330–337, 2005.
40. V. Borkar, K. Deshmukh, and S. Sarawagi, "Automatic segmentation of text into structured records," in *ACM SIGMOD Record*, vol. 30, pp. 175–186, 2001.
41. 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.
42. G. A. Miller, "Wordnet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
43. 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 (CIKM)*, pp. 1661–1664, ACM, 2010.
44. 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.
45. 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)*, pp. 315–320, IEEE, 2014.