GEO-NASS: A Semantic Tagging Experience from Geographical Data on the Media *

Angel L. Garrido, Maria G. Buey, Sergio Ilarri and Eduardo Mena

IIS Department, University of Zaragoza, Zaragoza, Spain.
{garrido, mgbuey, silarri, emena}@unizar.es

Abstract. From a documentary point of view, an important aspect when we are conducting a rigorous labeling is to consider the geographic locations related to each document. Although there exist tools and geographic databases, it is not easy to find an automated labeling system for multilingual texts specialized in this type of recognition and further adapted to a particular context.
This paper proposes a method that combines geographic location techniques with Natural Language Processing and statistical and semantic disambiguation tools to perform an appropriate labeling in a general way. The method can be configured and fine-tuned for a given context in order to optimize the results. The paper also details an experience of using the proposed method over a content management system in a real organization (a major Spanish newspaper). The experimental results obtained show an overall accuracy of around 80%, which shows the potential of the proposal.

Keywords: Geographic IR, gazetteer, semantic tagging; NLP; ontologies; text classification; media; news.

1 Introduction

The huge amount of digital information located in information systems in both public and private organizations is leading to a new problem: the difficulty to find accurate and relevant data among the vast amount of available data. The problem is aggravated daily because storage devices are becoming cheaper, the access to suitable document management software is becoming more common, and all these elements contribute to raise not only the total number of documents stored, but also its growth rate. In this situation, traditional indexers are not enough to provide the desired results when the search requires high precision and the volume of documents is large. This is why the interest in implementing expert labeling techniques for data catalogs increases every day, trying to to enable a situation where it is possible to find the desired information more easily.

The process of geotagging aims at extracting the places/locations where a text is framed. The complexity of this type of tagging depends on the tagging

* This research work has been supported by the CICYT project TIN2010-21387-C02-02 and DGA-FSE.

level to reach, the geography that must be covered, and the context of the text. One typical problem in this field is that we find names that are called like a location but are not. Therefore, we can identify three main problems in this field:

- *Geography - geography disambiguation*: it tries to find the exact location or geographic place that is being discussed in the text, by distinguishing among several locations with the same name.
- *Non geography - geography disambiguation*: it tries to detect correctly when the text is talking about a geographic place instead of about a different concept with the same name.
- *Selection of candidates*: when a certain location is mentioned in a text, references to other locations may also appear. However, not all the locations are always relevant, as they may not be related to the main story contained in the document. The objective is to establish a confidence threshold to decide if a tag is accepted or not.

In general, geolocation is a useful process commonly used in many information systems, but yet little known and rarely built in Content Management Systems (CMS). In a CMS it is typical to have indexing processes and search tools which can help to find quickly the documents that contain a particular word, but this kind of software has serious difficulties to face specific challenges when working with geographical issues. Therefore, integrating geolocation processes into a CMS can increase its capabilities of information management and enhance its usability from a documentary point of view.

There exist CMS that feature categorization professional tools like Drupal[1] or Athento[2], and other software products like OpenCalais[3], but it is usually not possible to customize such tools for a specific scenario. Therefore, working with those tools in very specialized multilingual environments may be impractical.

After analyzing different tools, we have not found any generic system able to automatically classify the information on a geographic basis that can be adapted to satisfy the particular specifications established by the organization that owns the documents, and of course, that takes into account all the typical difficulties for a geotagging system: to distinguish among places that share the same name, to identify a homonym of a place that is actually something else (a person, a verb, etc.) and to know how to geotag a text when it does not refer to explicit places.

Our proposal to solve this problem is to develope an automatic labeling system called GEO-NASS. The architecture of GEO-NASS includes different tools whose combination provides the desired results. These tools are a geographic database, Natural Language Processing (NLP) tools [1] and finally a disambiguation engine based on geographic, statistical and semantic techniques. The engine is designed to identify the location which is being told about in the case

[1] http://drupal.org/
[2] http://www.athento.com/
[3] http://www.opencalais.com/

of homonyms, to detect whether a word is really a location or not, and finally to establish the importance of each place/location related to the text for subsequent labeling. The system can also benefit from the context of the text to optimize its results. All of this is achieved through an open implementation which allows, on the one hand, adapting the geographical database to the context and, on the other hand, to fit the standard tagging system used in the specific organization considered. GEO-NASS has been tested with real data, specifically with about 10,000 news from one major Spanish newspaper. For comparison purposes, we considered news previously tagged manually by a professional documentation department of the mentioned newspaper. The experimental results obtained are very promising and show the interest of the proposal.

This paper is structured as follows. Section 2 briefly describes the state of the art. Section 3 explains the general architecture of our solution and presents the basic algorithmic details and some improvements introduced by using the context of the document and analyzing the semantics. Section 4 discusses the results of our experiments with real data. Finally, Section 5 provides our conclusions and some lines of future work.

2 Related Work

Although different techniques have been proposed to solve these problems of geotagging, we can find similar patterns in the most popular approaches [2]. They start applying a Named Entities Recognition (NER) and then they use a list of algorithms over these recognized entities. In this context, two important assumptions are considered: (1) that there is a single sense per discourse, that means that an ambiguous term is likely to mean only one of its senses when used multiple times, and (2) place names appearing in one context tend to show nearby locations. For example, if Florencia and Armenia appear in the same paragraph, then it is more likely that they indicate the two communities in Colombia rather than the larger and better known city in Italy and the country of Armenia, respectively.

Substantial effort has been applied to the most general task of Named Entity Recognition (NER) [3] concerned about in identifying proper names in a text. More recently, the specific task of determining which place is meant by a particular occurrence of a place name has been gaining attention. It requires general-world knowledge and cannot rely completely on information found in the text or even in a whole corpus. This general knowledge is provided in a gazetteer [4], which traditionally lists the names of all the places in an atlas. There are many of these free indexes that can be used for this purpose: Geonames, the NGA GEOnet Names Server, The World Gazetteer, the Alexandria Digital Gazetteer, etc.

Several disambiguation techniques have been studied for the task of geotagging. Some of them take into account the population of the location candidates as an important aspect to disambiguate places [2] or consider the context where the text is framed to establish a list of bonuses for certain regions [5]. Besides, many

of these techniques usually construct an N-term window on the right and left of the entity considered to be a geographic term, as some words can contribute with a positive or negative modifier [6], or they try to find syntactic structures like "city, country" (e.g. "Sevilla, Spain") [7]. Other techniques focus on finding common points in the location hierarchy of each candidate that appears in the text and they consider if they share the same political father (province, community, country, etc.) because this is a relevant factor when a term is being disambiguated [8]. Moreover, ontologies have been also used in disambiguation algorithms for geotagging. They are formal and explicit specifications of shared conceptualizations [9] and, in recent years, they have been applied as substitutes or complements of gazetteers [10, 11] because they can provide a rich vocabulary of classes and relations that describe a particular area, in this case a geographical scope.

In order to build our proposal, we used the techniques we have found most interesting from this section and in addition we have incorporated new techniques. This is fully explained in the following sections.

3 Architecture and Methodology

The development of this system has been carried out based on the NASS System. *NASS* is the acronym of *News Annotation Semantic System*, a software designed to obtain tags from a particular thesaurus [12] by using semantic tools and information extraction technologies [13].

In order to do its work, NASS uses statistics, NLP tools, Support Vector Machines (SVM) [14], and ontologies. The process is as follow: NASS combines NLP with statistical tools to obtain keywords and performs a filtering process of texts through SVM and a detailed labeling using ontologies [15].

In this work, we have expanded the NASS system providing it with a new option: geotagging. NASS is a tool that supports the inclusion of independent tagging modules, so we have encapsulated the new geotagging functionality into a separate unit that is incorporated into the workflow of NASS. We have called GEO-NASS to the resulting module, which we detail below. We consider that there is an appropriate gazetteer for all the documents to be labeled, and associated with certain places in the gazetteer we have several items belonging to the thesaurus with which we intend to label texts. The architecture of GEO-NASS and the NASS integration can be appreciated in Figure 1. The engine of GEO-NASS use the information provided from NASS (lemmas, keywords, and named entities), a database and a gazetteer to find possible locations, and also a set of its own context resources. With this information GEO-NASS finds the most probable locations the text is talking about, and extracts appropriate tags. Then the whole system merges the tags provided by NASS and the tags provided by GEO-NASS in one set.

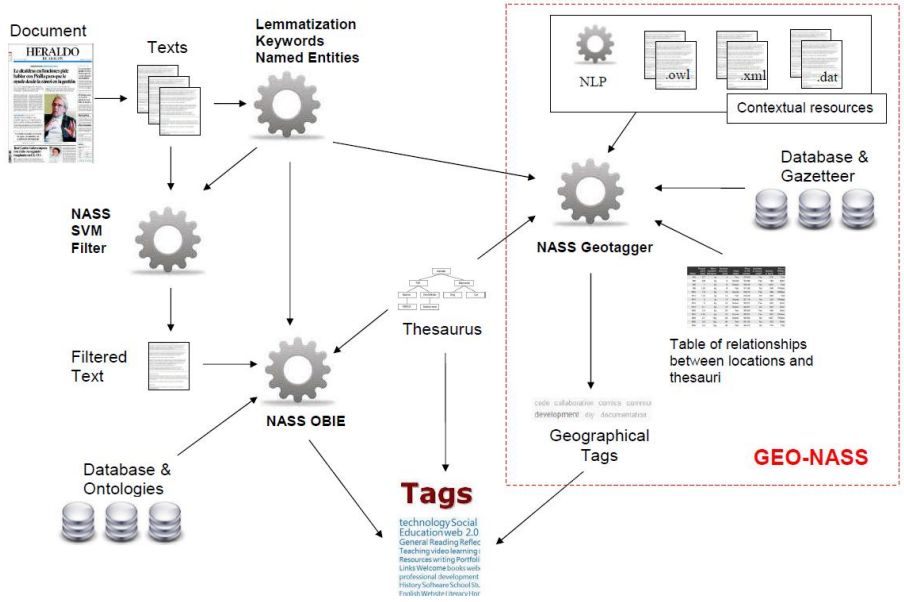

Fig. 1. General architecture of GEO-NASS and its integration with NASS

3.1 Basic GEO-NASS

As noted in Section 2, the problem of geographic information retrieval from a text has been widely studied, so to implement GEO-NASS we have taken advantage of the existing previous works.
The main functionalities considered in the initial version of GEO-NASS are:

- *Extraction of locations.* The aim is to extract from the text a list of proper nouns that correspond to a geographical location. For that purpose, our algorithm uses NASS functionalities: the text is analyzed by NASS using a Natural Language Processing software, and one of the tasks performed by this software is precisely to identify named entities [3]. Such entities may correspond to names of people, places, organizations, etc. Those named entities belonging to the gazetteer are candidates to be identified as places that must be labeled in the text. There are many gazetteers that can be used for this purpose. In our case, we have used *Geonames*[4] because it is an open source implementation, robust, comprehensive and frequently updated. We agree with other studies on the importance of a good adaptation of the gazetteer to the context [16].
- *Geographic disambiguation.* We will often find names of places with several entries in the gazetteer. For example, we can find in the gazetteer almost 20 different entries for cities and places called "Madrid". GEO-NASS assigns

[4] http://www.geonames.org/

more weight to those locations with a common parent in the same document [17]. Thus, if there exist five cities that could either be from Italy or from different countries, then it is more likely that the text concerns Italy than five different countries. Furthermore, if the locations are contained one within the other and both have the same designation, we will always choose the most specific one [2, 7].
- *Detection of the main focus.* A text may contain many locations, but probably a piece of news occurs in only one place. To establish a single geographic focus for the story, different factors can be taken into account, such as the number of occurrences of the term in the text and the list of elements in common with the other terms [2]. GEO-NASS can set the focus on a place that is not reflected in the text, when the scores for the different locations do not reach the acceptance threshold established. That means that the text is talking about several places so unimportant that the focus of the document is actually the place which includes all of them (a province, a region, or a country).

So, the basic algorithm implies performing the following steps: 1) to locate the named entities, 2) to eliminate those that are not present in the gazetteer, 3) to disambiguate them in case there are several possibilities (applying the rule of focus and choosing the most specific place if there is one inside another), and 4) to assign weights to each place to sort the results. After the process, we have a sorted list of places and weights (scores), which will be enriched with the thesaurus descriptors related to each location.

3.2 Improvements to the Basic Approach of GEO-NASS

Based on the previous basic algorithm, we have analyzed several potential improvements that benefit from information about the context and the specific language of the text. This information can be exploited without effort by a human, but requires adding several additional resources and procedures to automate it with a computer. The new algorithm is detailed below in pseudo-code:

```
PROGRAM GEO-NASS(
    INPUT: text as string;context as ListOf(Attributes);
               min_weight,  max_locations as real;
               g as Gazetteer; t as Thesaurus;
   OUTPUT: loc_list as ListOf(Location, weight, thesaurus_label);
BEGIN
   loc_list := Nothing; discard_list := Nothing;
   ne_list := Extraction_of_Named_Entites(text);
   ne_list := Remove_Stop_Words(ne_list);
   FOR EACH ne IN ne_list DO
       IF NOT(exist_in_gazetteer(g, ne)) THEN
           remove(ne, ne_list)
        ELSE
           loc_list_aux := Nothing;
```

```
            loc_list_aux := expand_locations(ne, g);
            loc_list_aux := filter_locations(loc_list_aux, g);
            loc_list.add(loc_list_aux);
        END IF
    END FOR
    FOR EACH loc IN loc_list DO
        w := calculate_weight(loc, text, context);
        IF w < min_weight THEN
             discard(loc_list, loc, w, discard_list)
          ELSE  assign_weight(loc_list, loc, w);
    END FOR
    IF loc_list = Nothing THEN
           IF discard_list <> Nothing THEN
                    loc_list := search_focus(discard_list)
               ELSE
                  loc_list := semantic_search(text, context);
      ELSE
          loc_list := filter_and_sort(loc_list, max_locations);

    loc_list := assign_thesaurus(loc_list, t, g);
    Output := loc_list;
END.
```

Notice that we not only consider the text, but also other attributes of the document. These attributes can be the tittle, the date, a summary, the author, the language in which it is written, etc. These metadata, which add context to the document, will be used in certain parts of the algorithm to optimize the results. We add also as an input the minimum weight that a place must have to be considered and the maximum number of places to obtain. Both parameters are optional, but they could help to improve the results. The first parameter avoids labeling the text with tags related to locations of minor importance in the document. The second parameter is considered relevant because it can contribute, as discussed in Section 4, to reduce the noise caused by too many locations present in the text. The high-level functions that are used in the previous algorithm are explained as follow:

1. Extraction of named entities (*Extraction_of_Named_Entites*). Among the named entities we can find names of people and organizations. One way to minimize errors is to obviate those named entities that are clearly not places, which is not a trivial task. We propose to analyze the sentence in which the named entity is embedded. Thanks to NASS, the document has already been lemmatized and the system can check the type of each word by using a morphological analyzer and its function by using a syntactic parser. Penalties or bonuses are assigned, in case certain words that are acting with a certain function are found, to try to detect with reasonable accuracy if a named entity is or not a location. For example, if the named entity is acting as subject of the sentence and the related verb is “to sing”, that means that

the named entity is probably a person and we can discard this occurrence. We also apply the syntactic structure detection method in this procedure.

2. Remove stop words (*Remove_Stop_Words*). Methods for obtaining named entities are not perfect and may cause noise. Nevertheless, as we have in the context the language as a parameter, we propose as an improvement the preparation (for each of the potential input languages) of a black list of words that, due to its high use frequency, may appear as named entities simply because they are capitalized. This simple measure speeds up the execution of the algorithm and avoids errors.
3. Expand and filter locations (*expand_locations*, *filter_locations*). As explained before in the basic algorithm, we first obtain all the places whose description matches the entity named in the text. Once expanded, the locations are filtered using the disambiguation mechanisms discussed in the previous section.
4. Calculate the weight (*calculate_weight*). In the basic algorithm described in Section 3.1, we use this formula to compute the weight of a location [2, 17].:

$$w = (wce + wp) * N$$

Where the meaning of each term in the formula is as follows:

- *wce (weight by common elements)* denotes that the weight is increased if there are other places with the same parent, thus applying the rule of "a single sense per discourse".
- *wp (weight per population)* denotes that the places with a higher population will be assigned a greater weight.
- *N (number of repetitions)* denotes the number of times the name of the place appears in the text.

We improved and extendes the previous formula by considering also the list of attributes of the document, the context, and some NLP and semantic aspects:

$$w = (wce + wp) * N * wpos * wk * wat$$

Where the meaning of the new terms is as follows:

- *Weight depending on the position (wpos, with* $0 \leq wpos \leq 5$*).* We make an adjustment by the position of each location within the document. If it is located at the beginning we assign it a bonus, and if it is at the end we penalize it. The intuition is that in any text the most important keywords are usually at the beginning of the document [6].
- *Weight by keywords (wk, with* $0 \leq wk \leq 2$*).* Another function provided by NASS is the detection of keywords by using statistical methods [18], which we have also taken advantage of to improve the results of GEONASS. We have observed, through specific cases, that locations appearing near the detected keywords are more important than those that are farther away. We have developed a simple metric that consists of measuring the distance (in words) from each of the keywords to the locations. Those which are the closer receive a higher score, increasing the weight of that location.

- *Weight depending on the attributes and/or the type of place (wat, with* $0 \leq wat \leq 10$*).* Finally, we use this parameter to improve the results in a practical and smart way. If we have simple but powerful common-sense context-dependent rules, this could be a good place to implement them using a rule table. If there are difficulties finding these rules, machine learning can be used to create the configuration table. For example, I could decide that documents with the value *International* in a field *topic* give a big bonus to locations of type *country*. This would make these places appear at the top of the list of results.

After computing the weight of each individual place, we can check if it has reached the minimum acceptance threshold required, which can be set according to the requirements of the specific context considered: a high threshold will increase the hit rate and reduce the errors (misplaced labels, i.e., labels that are used but should not be applied to the document), but will also increase the number of forgotten labels (places that are applicable but that are not used for labeling). All the places below the threshold are removed from the candidate list and transferred to a *list of places discarded*, that will be used in case no relevant place is found.

5. Search the focus (*search_focus*). In the basic algorithm the objective of this function is to try to find the focus of the news when at least one place is found relevant enough. Here the system also considers the list of discarded places for cases where no place is identified as relevant.
6. Semantic Search (*semantic_search*). If the list of places discarded is also empty, we are at a stage that we had not considered yet: What happens when we find nothing relevant in the gazetteer? Indeed, it is quite possible (about 18% of the times in our experimental tests). In fact, there are texts where the location is implicit. Our proposal to solve the problem is to follow a semantic approach by using ontologies. Specifically, we use ontologies including information about important aspects related to certain locations: for example, for a city we can populate the ontology with its most important streets, monuments and outstanding buildings, neighborhoods, etc. When a text has not explicit location identified, we will use the keywords extracted from the text (obtained through statistical analysis, based for example on the frequency of terms [18]) to query the ontologies. Then, we will choose the place whose ontology gives a higher number of correct answers for those queries. Ontologies are defined in OWL [19] or RDF and are interrogated using SPARQL [20]. Existing ontologies could be reused or could be designed from scratch.
7. Filter and sort (*filter_and_sort*). This function sorts the list of candidate locations according to their weight. Besides, if a maximum number of results has been set, then the list will be truncated to keep only the elements with the higher weights.
8. Assign a thesaurus (*assign_thesaurus*). We assume the existence of a related gazetteer that will be used for labeling. This function simply labels the document using the tags obtained from the thesaurus.

In summary, the system executes an algorithm that first extracts a set of named entities present in the text and then removes those words which may cause noise, such as articles or prepositions, that may appear as named entities simply because they are capitalized. For each Named Entity found, this algorithm searches if it is included in the gazetteer; if not, it removes it from the possible locations. Then, it expands and filters the location to obtain all the places whose description matches the named entity in the text using disambiguation mechanisms. Next, for each possible location, it calculates a weight using the formula described above; if it does not pass a minimum threshold value it is removed from the candidate list and transferred to a list of places discarded. In case it has not found any relevant place then it tries to find a focus from this list. If that list is empty, then the location may be implicit, and so it tries to solve the problem following a semantic approach by using ontologies. Otherwise, it finds a list of relevant places and sorts it according to their weight. Finally, it assigns to each location its appropriate thesaurus.

4 Experimental Results

This section discusses the results of experimental tests carried out to check the performance of the proposed algorithm. For testing in a real environment we have used a corpus of 9,520 news previously labeled by a professional documentation department of *Heraldo de Aragón*[5], a major Spanish media. The professional thesaurus used to label has more than 10,000 elements. The purpose is to approximate the automatic labeling of the locations to the labeling specified by that department. As the news in our data set were manually annotated by the professionals working in the documentation department of the newspaper, we can compare our automatic labeling with that performed by humans. Since the news are stored in a database, we can use the various fields of the table which are stored as context information. These fields are the list of attributes explained in Section 3.

Therefore, experiments have been performed with Spanish texts, so the input language is another context factor to consider. We had to run disambiguation algorithms in Spanish, with the additional difficulty that this implies [21–23]. We have used Freeling [24] as Spanish NLP tool, we have fixed a stop-word list, and we have prepared six ontologies populated with about a hundred words related with the most important cities in the region of the media.

We have performed four experiments. In the first one (experiment *E1*), we have used the basic algorithm presented in Section 3.1. In the second one (*E2*), we simply established a limit to the number of results. In the third one (*E3*), several improvements outlined in Section 3.2 were applied in a generic way for all the sections in the newspaper (sports, international, politics, etc.): the context window, the stop word list, the structure detection method and the weight *wpos* correction. Finally, in the fourth experiment (*E4*) the weights *wk* and *wat* were

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

introduced, we added the semantic search option to solve cases in which no location appeared explicitly in the text, and the algorithm was improved to allow a different acceptance threshold for each specific section in the newspaper.

The results of the experiments are shown in Fig. 2. We analyze the following measures, commonly used in the Information Retrieval context [25]: the precision, the recall, and the F-Measure. The first three graphs refer to these measures applied on five major sections of the newspaper, and in the last graph we display the average values for all the sections.

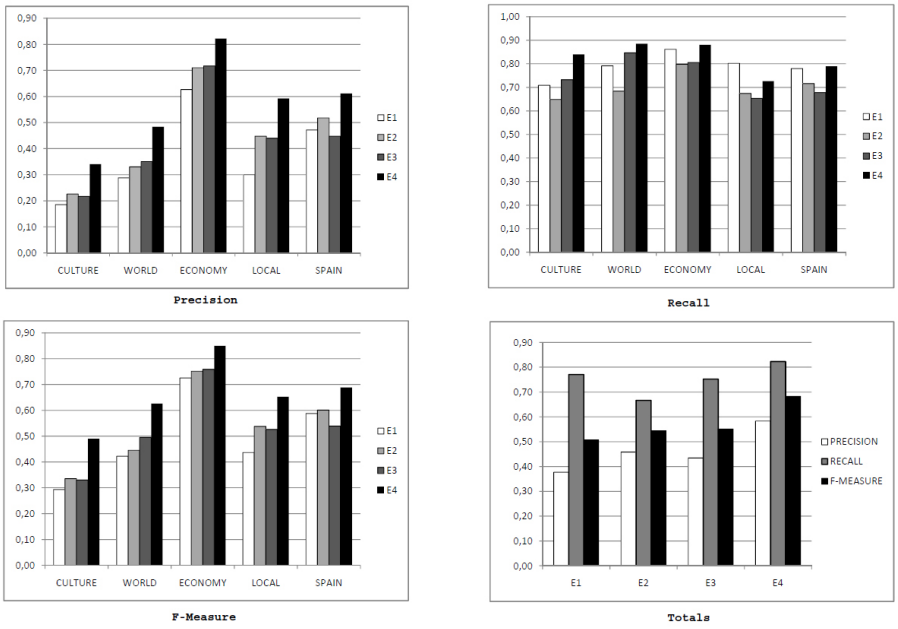

Fig. 2. Results of experiments with the news of a major Spanish media for the first two months of 2013

Based on the results, we can make the following statements:

- The progression of the indicators from experiment 1 to 4 are pretty good: an 82% of recall, has been achieved, and the precision is improved from a modest 38% to an acceptable 58%, reaching an F-measure value of near 70%, which is acceptable to start using GEO-NASS in a real environment.
- In the first version of the algorithm, the number of labels used to annotate each document was not limited, yielding an average of 3.85 tags per document. As the average number of geotags per document used by the real documentation department was 1.89, that was one main reason for the low precision in experiment *E1*. So, the algorithm was generating excessive noise because it was labeling with all the places reaching the minimum threshold.

- In the second experiment we limited the number of results to three. After this there was an improvement in the recall, but obviously a setback for the precision, leading to the conclusion that the problem was on the scoring used to decide which places would be the first three, so we had to improve the adjustment of the weights.
- The results of the third experiment show an improvement in the recall but some decrease in the precision, which means that we have managed to improve the order of the results but we continue to generate many invalid tags.
- In the last experiment, the use of semantics to label items without explicit locations, combined with the *wat* adjustment, resulted in a substantial difference in the results (10%-15%). Adjusting the acceptance threshold per section contributed to a major reduction in the noise caused by excessive labeling.

Finally, if we look at the overall results for all the sections, we could say that the results of the precision could be considered to be not very good, but by analyzing samples of the results with the documentation department it was found that a significant proportion (around 50%) of the labels considered erroneous actually could be considered correct because their omission was due to redundancy or oversights of the person who tagged the news. In other words, if the documentation department had found those labels placed in advance then they would not have removed them (instead, they would have considered them as appropriate labels). Therefore, we can say that *the precision in practice is much better* than what is reflected in Fig. 2, reaching up to near the 80%.

5 Conclusions and Future Work

In this paper, we have presented a method to solve the problem of geotagging documents using specific vocabulary. The algorithm combines techniques well-known in the field of geographic labeling with other NLP tools and statistic methods. We also added the use of ontologies to solve certain problems that have been identified as potential loopholes in labeling standard procedures. Our main contributions are:

- Testing geotagging algorithms in a real environment, comparing experimental results with the actual labeling of a professional documentation department.
- Improving the results achieved by well known techniques using a new algorithm supported by NLP tools, ontologies and statistic methods.
- Measuring quantitatively the contribution of each geotagging technique to the final result.

The proposal has been tested in the real environment of a major Spanish newspaper. So, this paper also helps to increase the small number of experimental auto-tagging studies available for Spanish, which always implies a greater difficulty than English because of its great ambiguity and verb complexity [21–23].

In any case we believe that the algorithm used is independent of the language, question that we plan to test in a short-term.

In this work, we were able to take advantage of the fact that we have thousands of texts labeled by a professional team of archivists, which has allowed us to verify accurately the results of our tests and compare them with the actual labeling. The fact that it has been tested with professional archivists reduces the subjectivity that a manual labeling could have involved.

In conclusion we can say that, when using classic geotagging procedures over documents in real environments, the algorithms require certain adjustments related to language and context to provide acceptable results in a production environment, such as the case of applying geotagging over news in a media. Our proposed algorithm, incorporating semantic tools, NLP procedures and statistical functions, has responded adequately in such an environment, achieving 80% of precision and recall in practice. So, we are confident that it could also be used in other similar environments with guarantee of success.

As future work, we would like to automate the adaptation of the algorithms to real contexts, for which we think that the incorporation of machine learning techniques could be very helpful. Furthermore, we believe that it could be very interesting to expand the scope of the labeling, adapting these techniques for tagging specific places such as neighborhoods, streets, squares or buildings within a particular identified location.

References

1. A. F. Smeaton, *Using NLP or NLP Resources for Information Retrieval Tasks. Natural Language Information Retrieval.* Kluwer Academic Publishers, 1999.
2. E. Amitay, N. Har'El, R. Sivan, and A. Soffer, "Web-a-where: geotagging web content," in *27th International Conference on Research and Development in Information Retrieval (SIGIR'04)*, pp. 273–280, ACM, 2004.
3. S. Sekine and E. Ranchhod, *Named Entities: Recognition, Classification and Use.* John Benjamins, 2009.
4. L. Hill, "Core elements of digital gazetteers: placenames, categories, and footprints," in *Research and Advanced Technology for Digital Libraries*, pp. 280–290, Springer, 2000.
5. G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman, "Determining the spatial reader scopes of news sources using local lexicons," in *18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 43–52, ACM, 2010.
6. E. Rauch, M. Bukatin, and K. Baker, "A confidence-based framework for disambiguating geographic terms," in *HLT-NAACL 2003 Workshop on Analysis of Geographic References, vol. 1*, pp. 50–54, Association for Computational Linguistics, 2003.
7. H. Li, R. K. Srihari, C. Niu, and W. Li, "Location normalization for information extraction," in *19th International Conference on Computational Linguistics-Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.
8. B. Pouliquen, M. Kimler, R. Steinberger, C. Ignat, T. Oellinger, K. Blackler, F. Fuart, W. Zaghouani, A. Widiger, A.-C. Forslund, and C. Best, "Geocoding

multilingual texts: Recognition, disambiguation and visualisation," *The Computing Research Repository (CoRR) abs/cs/0609065*, 2006.
9. T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
10. K. Janowicz and C. Keßler, "The role of ontology in improving gazetteer interaction," *International Journal of Geographical Information Science*, vol. 22, no. 10, pp. 1129–1157, 2008.
11. I. M. R. Machado, R. O. de Alencar, R. de Oliveira Campos Jr, and C. A. Davis Jr, "An ontological gazetteer and its application for place name disambiguation in text," *Journal of the Brazilian Computer Society*, vol. 17, no. 4, pp. 267–279, 2011.
12. A. Gilchrist, "Thesauri, taxonomies and ontologies - an etymological note," *Journal of Documentation*, vol. 59, no. 1, pp. 7–18, 2003.
13. A. Garrido, O. Gómez, S. Ilarri, and E. Mena, "Nass: News Annotation Semantic System," in *23rd International Conference on Tools with Artificial Intelligence*, pp. 904–905, IEEE, 2011.
14. T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *10th European Conference on Machine Learning*, pp. 137–142, Springer, 1998.
15. A. Garrido, O. Gómez, S. Ilarri, and E. Mena, "An experience developing a semantic annotation system in a media group," *17th International Conference on Applications of Natural Language Processing to Information Systems*, pp. 333–338, Springer, 2012.
16. M. D. Lieberman, H. Samet, and J. Sankaranarayanan, "Geotagging with local lexicons to build indexes for textually-specified spatial data," in *2010 IEEE 26th International Conference on Data Engineering*, pp. 201–212, IEEE, 2010.
17. W. A. Gale, K. W. Church, and D. Yarowsky, "One sense per discourse," in *Workshop on Speech and Natural Language (HLT'91)*, pp. 233–237, Association for Computational Linguistics, 1992.
18. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *Information Processing and Management*, vol. 24, pp. 513–523, Pergamon Press, Inc., 1988.
19. D. L. McGuinness and F. van Harmelen, *OWL Web Ontology Language Overview. W3C Recommendation. Available at:* `http://www.w3.org/TR/owl-features/`, 2004.
20. E. Prudhommeaux, *SPARQL Query Language for RDF. W3C Working Draft. Available at:* `http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/`, 2006.
21. R. Carrasco and A. Gelbukh, "Evaluation of TnT Tagger for Spanish," in *4th Mexican International Conference on Computer Science*, pp. 18–25, IEEE, 2003.
22. M. Vallez and R. Pedraza-Jimenez, "Natural language processing in textual information retrieval and related topics, 'Hipertext.net' no. 5, 2007.
23. G. Aguado de Cea, J. Puch, and J. Ramos, "Tagging spanish texts: The problem of 'se'," in *Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pp. 2321–2324, 2008.
24. X. Carreras, I. Chao, L. Padró, and M. Padró, "Freeling: An open-source suite of language analyzers," in *4th International Conference on Language Resources and Evaluation*, pp. 239–242, European Language Resources Association, 2004.
25. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.