University of Alicante at NTCIR-9 GeoTime

Fernando S. Peregrino
fsperegrino@dlsi.ua.es

David Tomás
dtomas@dlsi.ua.es

Fernando Llopis
llopis@dlsi.ua.es

Department of Software and Computing Systems
University of Alicante
Carretera San Vicente del Raspeig s/n - 03690 Alicante (Spain)

ABSTRACT

In this paper we present a complete system for the treatment of the geographical dimension in the text and its application to information retrieval. This system has been evaluated in the *GeoTime* task of the *9th NTCIR* workshop, making it possible to compare the system with other current approaches to the topic. In order to participate in this task we have added to our *GIR* system the temporal dimension. The system proposed here has a modular architecture in order to add or modify features. In the development of this system we have followed a *QA*-based approach to improve the system performance.

Keywords

Geographical Information Retrieval, Geo-Tagging, Spacial Information, Temporal Information

Team Name: *GETUA*
Subtasks/Languages: *English Monolingual GeoTime*
External Resources Used: *Apache Lucene, Yahoo! Placemaker, Yahoo! Search BOSS, FreeLing, GeoNames*

1. INTRODUCTION

The *GeoTime* task included in the *NTCIR* workshop deals with Information Retrieval (*IR*), focus in an unstructured text document collection, with geographical and temporal constraints. In [2] we can see the task mentioned in more detail.

GIR is a specialization of *IR* with geographic metadata associated. *IR* systems usually see the documents as a collection or "bag of words". By contrast, *GIR* systems require semantic information, i.e. they need a place name or geographical feature associated with the document. Because of this, in *GIR* systems, it is common to separate the analysis and text indexing from the geographic indexing.

Temporal information is available in every document either explicitly, e.g., in the form of temporal expressions, or implicitly in the form of metadata. Recognizing such information and exploiting it for document retrieval and presentation purposes are important features that can significantly improve the functionality of search applications. Temporal Information Retrieval (*TIR*), analogously to the *GIR*, is a specialization of Information Retrieval with temporal metadata associated.

The objective of this work is to adopt a first approach in the geo-temporal *IR* field watching how a basic *IR* system can be improved including geo-temporal *IR* intelligence, and to know what methods used in them have a better performance. To that end, we have elaborated this paper structured as follow: In section 2, we provide a detailed description of the system, describing the system and storage architecture as well as the system operation. Subsequently, in section 3, the results of our participation in the *NTCIR-GeoTime 9* are shown. Finally, in section 4, we describe the conclusions and future work in this area.

2. SYSTEM DESCRIPTION

For the creation of this *GIR* system we have chosen to implement it in a modular way with the intention of adding new components, or improving the existing ones.

Figure 1 shows the architecture of our system with all the modules which compose it and the data flow. Solid lines represent data flow which takes place in preprocessing time. On the other hand, broken lines represent data flow which takes place in execution time, being the thicker ones those that process the query and the thinner ones those which execute it.

2.1 System Modules

In the following paragraphs every module in our system is described in detail.

2.1.1 Search Engine

This module has two functionalities: to index the whole corpus, and to retrieve a set of relevant documents for a given query.

The search engine chosen for this system was *Lucene*[1]. We have included characteristics to this search engine, such as a stemming and stopword removal. To index the corpus most techniques are based on inverse index [1]. The ranking function *Okapi BM25*[4] has been used to rank the results according to their relevance. Eventually, it has been chosen to retrieve up to 1,000 relevant documents per query.

2.1.2 Linguistic Analysis Module

This module carries out the following tasks:

- Stemming queries.
- Stopwords removal.
- Acquisition of the toponyms from the queries based on a place names data base (*GeoNames*[2]).

[1] `http://lucene.apache.org/java/docs/index.html`
[2] `http://www.geonames.org/`

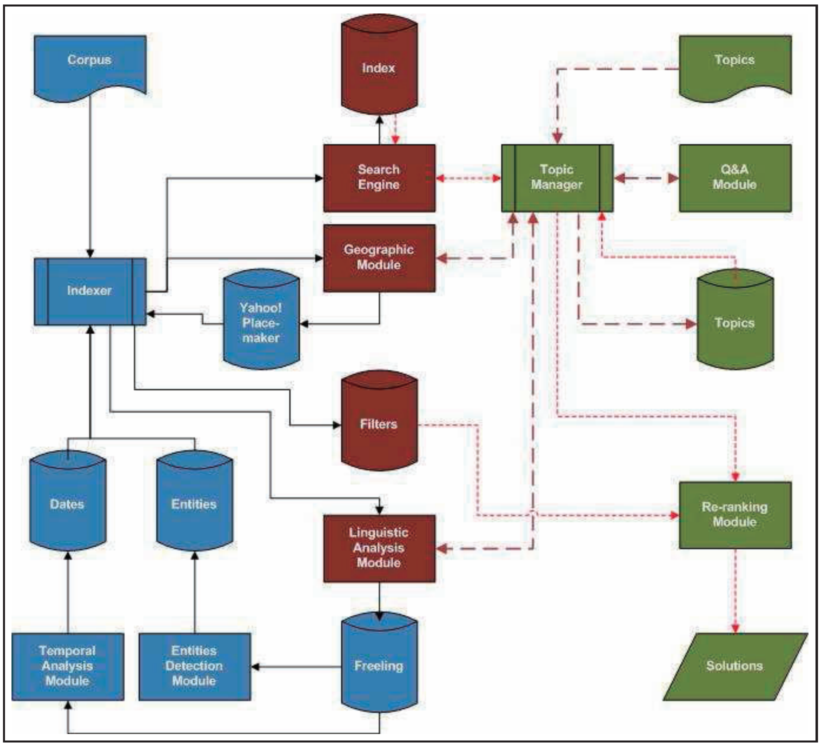

Figure 1: Diagram of the workflow in the *GIR* system implemented.

- Identification of the dates in the queries through a syntactic analyser (*FreeLing*[3]).
- To obtain the geographical and temporal queries restrictions (e.g. is the country name enough or is it necessary the town name?, do you need to know the month and year of a event or is it necessary the completed date?, etc.). To archive this goal, the narrative part of the query is analysed looking for trigger words such as country, state, city, day, exact day, etc.
- Search for other entity types (e.g. people names, companies names, etc.), by means of a syntactic analyser (*FreeLing*).
- To obtain other common terms for a future query expansion.

2.1.3 Question Answering Module

This module tries to obtain from the web geographic and temporal expressions which are relevant to the query. First, the query is sent to *Yahoo!*, collecting the first 1,000 snippets from the returned results. All dates and places from these snippets are then extracted. The total number of occurrence is computed and normalized, obtaining the 10 most relevant for each of these categories: date (completed or uncompleted), month and year, year, and place names.

2.1.4 Temporal Analysis Module

For this module we have used the linguistic analyser *FreeLing*. Specifically, we have used the Part Of Speech (*POS*) to obtain every date reference in the corpus, including the document date (the date of publication in the newspaper). Thereby, in the pre-process time, a new file is made for each document including all the dates detected in it. Subsequently, at runtime, it is tried to match the query dates with the dates of the documents returned by the search engine, giving more relevance to full dates (year, month and day) than to incomplete dates (year and month, or just year).

2.1.5 Geographic Module

We have based on *Yahoo! Placemaker*[4], an online tool, to develop this module. *Yahoo! Placemaker* is a freely available *geoparsing* web service. It helps developers make their applications location-aware by identifying places in unstructured and atomic content (e.g. feeds, web pages, news, status updates, etc.) and returning geographic metadata for geographic indexing and *markup*. The service identifies places mentioned in text, disambiguates those places, and returns unique identifiers (*WOEIDs*) for each, as well as information about how many times the place was found in the text, and where it was found in the text. With this tool we have covered subjects described in [3] for the develop of a *GIR* system, such as geo-tagging, place names disambiguation, and recognition of fuzzy geographic entities.

In the particular case of our system, we use *Yahoo! Placemaker* to obtain the place names, to disambiguate them, and to get the upper and lower geographic administrative entities, e.g. if in a query it is asked for volcanoes in Africa, and there is an article about an erupting volcano in Cameroon, we will deduce that Cameroon is in Africa through *Yahoo! Placemaker*.

Yahoo! Placemaker returns a *XML* document for each

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

[4] http://developer.yahoo.com/geo/placemaker/

piece of text we send to it. Finally, the module stores all the pertinent geographic information of the article in a new *XML* document.

This module has another function when we have to analyse queries. Specifically, it collect the place names from the queries *XML* documents and it transforms them to the *Yahoo!* unique identifier (*WOEID*) to disambiguate them.

2.1.6 Entity Detection Module

This module is responsible for storing a new document for each document. In the mentioned new document will be every identified entity by *FreeLing* from the original article. It will be useful to make a final *XML* filter query document.

2.1.7 Re-ranking Module

This module works in runtime, i.e. at the moment of doing the search for a concrete query. The goal of this module is to intersect the *XML* query documents and the *XML* corpus documents which have been returned as a solution by the search engine. The re-ranking is given by two weighting functions. In both it can be observed the importance of *Lucene* weight, but they differ in:

- Weighting function 'ω_A'. It allows to evaluate the QA module relevance as well as the descriptive and narrative parts of the query:

$$\omega_A = \lambda \cdot L + (1-\lambda) \cdot [\alpha \cdot (\beta \cdot \omega_{desc} + (1-\beta) \cdot \omega_{narr}) + (1-\alpha) \cdot \omega_{QA}] \quad (1)$$

  Where:

  - λ = *Lucene* weight.
  - L = *Lucene* normalized value. The score returned by *Lucene* normalized between 0 and 1.
  - α = Weight given to the descriptive and narrative part of the query.
  - β = Weight given to the descriptive part of the query.
  - ω_{desc} = Normalized value of the descriptive part of the query. This value is obtained adding the following parts:
    * Geographical: A counter is increased for each geographical entity which is in the descriptive part of the topic and is found in the document that is being evaluated. The final score of this part is obtained dividing the total number of geographical entities in the topic into this counter.
    * Temporal: If any of the obtained dates from the descriptive part of the topic match with the document date, then the score of this part will be 1. Otherwise, a counter is increased for each temporal expression which is in the descriptive part of the topic and is found in the document that is being evaluated. The final score of this part is obtained dividing the total number of temporal expressions of the topic.
    * Entities: A counter is increased for each non-geographical entity which is in the descriptive part of the topic and is found in the document that is being evaluated. The final score of this part is obtained dividing the total number of non-geographical entities of the topic into this counter.

    If there are not any of these three parts the score returned will be 0, otherwise, the final score sum will be divided into the number of parts that there were, in order to get a normalized value.
  - ω_{narr} = Normalized value of the narrative part of the query. This value is accomplished in analogous way to the descriptive part.
  - ω_{QA} = Normalized value of the *QA* part of the query. This value is obtained adding the following parts:
    * Dates: If the document date or any other date from the document text match some of the dates that there are in the *dates* section of the *XML* topic document (see figure 3), then, among all dates which match, that one which has the maximum weight returns its value and it is multiplied by 0.6. Otherwise, zero is returned.
    * Month and year: This score is calculated in a similar way to the previous one, but with the *dates_month* section and it is multiplied by 0.35.
    * Year: This score is calculated in a similar way to the previous one, but with the *dates_year* section and it is multiplied by 0.05.
    * Location: This score is calculated in a similar way to the *Dates* score.

    If there are not any of these four parts, the score returned will be 0. Otherwise, there will be three possible cases:

    1. There are not any dates but there are some locations: the returned score will be that one returned by the location section.
    2. There are not any locations but there are some dates: the returned score will be the sum of all three dates section scores.
    3. There are both dates and locations: the returned score will be the sum of all four sections scores divided into two in order to obtain a normalized score.

- Weighting function 'ω_B'. It allows to evaluate the geographic, temporal, and entities relevance:

$$\omega_B = \lambda \cdot L + (1-\lambda) \cdot [\alpha \cdot (\beta \cdot \omega_{geo} + (1-\beta) \cdot \omega_{temp}) + (1-\alpha) \cdot \omega_{ent}] \quad (2)$$

  Where λ and L have the same mean as in the previous weighting function, and the rest of symbols mean:

  - α = Weight given to the geographic and temporal part of the query regarding the entities one.
  - β = Weight given to the geographic part of the query respect to the temporal one.
  - ω_{geo} = Normalized value of the geographic part. This part gets its score from:

1. Descriptive part of the topic. Similarly to the geographical part of the topic into the ω_{desc} and the ω_{narr} parts in the weighting function 'ω_A', a counter is increased for each geographical entity which is in the descriptive part of the topic and is found in the document that is being evaluated. The final score of this part is obtained dividing the total number of geographical entities in the topic into this counter. If there is not any geographical entity in the descriptive part of the topic, or they do not match with any in the document which is being evaluated, the score of this section will be zero.
2. Narrative part of the topic. Analogous to the previous one.
3. *QA* (*locations* in figure 3) part of the topic. The same that in the *Location* part, inside the ω_{QA} part in the weighting function 'ω_A',

- ω_{temp} = Normalized value of the temporal part. Analogous to the ω_{geo} but with temporal expressions instead geographical ones.
- ω_{ent} = Normalized value of the entities part. Akin to the two previous scores but regardless of the *QA* part.

2.2 Storage Scheme

In order to expedite and streamline the process when executing queries, it have been kept two sets of *XML* documents *XML*: filter documents for each corpus articles and analysis documents for each query.

2.2.1 Corpus

As shown in Figure 2, this *XML* documents are divided into three parts, geographical, temporal, and entities.

Geographical.
This part stores the geographic relevant aspect from the original article. This geographical information is extracted with *Yahoo! Placemaker*. Specifically, for each place name in the text the following data is stored: type (it indicates whether the place name speaks about the general document scope, an ancestor, a specific place name found in the text, etc.), *WOEID*, kind of toponym (country, city, imprecise region, etc.), and place name.

Temporal.
In this set all document dates are stored. At least, there must be one entry, the article date, and after this it will be the rest of dates found in the text.

Entities.
This section has all non-geographical entities named in the original article.

2.2.2 Queries

As shown in Figure 3, it has been made the following sections for each query:

- Search terms: all search term without stopwords.
- Stemmed search terms: stemmed search terms section.
- Filters:
  - Descriptive part: dates, place names, and entities found in the descriptive part of the query.
  - Narrative part: Analogous to the previous one and, in addition, it has the geographical and temporal constrains.
  - Query expanding. It has expanded entries of the most representative terms of the query to a possible future query expansion.
  - *QA*: It has the following extracted data from *Yahoo!*: dates, year and month dates, year dates, and toponyms. It has normalized the 10 more representative values for all four piece of date aforementioned.

```
▼<query id="GeoTime-0040">
   <search>Concorde crash</search>
   <search_lemma>concorde crash</search_lemma>
  ▼<filters>
    ▼<description>
      ▼<entities>
         <item>concorde</item>
       </entities>
     </description>
    ▼<narrative>
      ▼<entities>
         <item>concorde</item>
       </entities>
      ▼<commons>
         <item>crash</item>
         <item>airliner</item>
       </commons>
     </narrative>
    ▼<yahoo>
      ▼<dates>
         <item weight="1.0">[??:??/??/2000:??.??:??]</item>
         <item weight="0.8043478">[??:25/7/2000:??.??:??]</item>
         <item weight="0.6847826">[??:??/7/2000:??.??:??]</item>
         <item weight="0.1521739">[??:??/??/2003:??.??:??]</item>
         <item weight="0.07608695">[??:??/??/1976:??.??:??]</item>
         <item weight="0.07608695">[??:??/??/1969:??.??:??]</item>
         <item weight="0.06521739">[??:11/9/2001:??.??:??]</item>
         <item weight="0.06521739">[??:2/2/2010:??.??:??]</item>
         <item weight="0.04347826">[??:??/6/2000:??.??:??]</item>
         <item weight="0.04347826">[??:10/4/2003:??.??:??]</item>
       </dates>
      ▼<dates_year>
         <item weight="1.0">[??:??/??/2000:??.??:??]</item>
         <item weight="0.098425195">[??:??/??/2003:??.??:??]</item>
         <item weight="0.08661418">[??:??/??/2010:??.??:??]</item>
         <item weight="0.05905512">[??:??/??/2001:??.??:??]</item>
         <item weight="0.05511811">[??:??/??/1969:??.??:??]</item>
         <item weight="0.03937008">[??:??/??/1976:??.??:??]</item>
         <item weight="0.023622047">[??:??/??/2008:??.??:??]</item>
         <item weight="0.01968504">[??:??/??/2011:??.??:??]</item>
         <item weight="0.007874016">[??:??/??/1985:??.??:??]</item>
         <item weight="0.007874016">[??:??/??/1979:??.??:??]</item>
       </dates_year>
      ▼<dates_month>
         <item weight="1.0">[??:??/7/2000:??.??:??]</item>
         <item weight="0.06849315">[??:??/2/2010:??.??:??]</item>
         <item weight="0.05479452">[??:??/8/2000:??.??:??]</item>
         <item weight="0.047945205">[??:??/3/1969:??.??:??]</item>
         <item weight="0.047945205">[??:??/10/2003:??.??:??]</item>
         <item weight="0.047945205">[??:??/7/2001:??.??:??]</item>
         <item weight="0.04109589">[??:??/9/2001:??.??:??]</item>
         <item weight="0.034246575">[??:??/12/2010:??.??:??]</item>
         <item weight="0.02739726">[??:??/6/2011:??.??:??]</item>
         <item weight="0.02739726">[??:??/4/2003:??.??:??]</item>
       </dates_month>
      ▼<locations>
         <item weight="1.0">615702</item>
         <item weight="0.49312714">23424819</item>
         <item weight="0.3676976"/>
         <item weight="0.1580756"/>
         <item weight="0.10584193">23424977</item>
         <item weight="0.0790378">44418</item>
         <item weight="0.06872852"/>
         <item weight="0.04467354">2384019</item>
         <item weight="0.030927835">24865675</item>
         <item weight="0.02749141">2459115</item>
       </locations>
     </yahoo>
   </filters>
 </query>
```

Figure 3: *XML* topic document sample

2.3 System Operation

The system operation is divided into three phases: preprocessing and indexing the corpus, processing queries, and running queries.

```
▼<document id="XIE20000726.0196">
  ▼<geo>
    ▼<entity>
        <documentType>ancestor</documentType>
        <woeid>12597155</woeid>
        <type>Department</type>
        <name>Paris</name>
      </entity>
    ▼<entity>
        <documentType>administrativeScope</documentType>
        <woeid>615702</woeid>
        <type>Town</type>
        <name>Paris, Ile-de-France, FR</name>
      </entity>
    ▼<entity>
        <documentType>ancestor</documentType>
        <woeid>7153319</woeid>
        <type>State</type>
        <name>Île-de-France</name>
      </entity>
    ▼<entity>
        <documentType>ancestor</documentType>
        <woeid>23424819</woeid>
        <type>Country</type>
        <name>France</name>
      </entity>
    </geo>
  ▼<temp>
      <dateDoc>[??:26/07/2000:??.??:??]</dateDoc>
      <date>[??:26/07/2000:??.??:??]</date>
      <date>[??:??/??/1969:??.??:??]</date>
    </temp>
  ▼<names>
      <name>Crash crash</name>
    ▼<name>
        French_Minister_of_Transport_Jean-Claude_Gayssot french_minister_of_transport_jean-claude_gayssot
      </name>
      <name>Concorde concorde</name>
    </names>
  </document>
```


Figure 2: *XML* filter corpus document sample.

2.3.1 *Corpus Pre-process*

In this phase the stemmed corpus is parsed with *Lucene*, *Yahoo! Placemaker* obtains the geographic entities, and with *FreeLing* the temporal expressions and the rest of named entities are obtained. With all this information a new *XML* file is made for each corpus article (see Figure 2). These *XML* document will be useful to know the article relevance respect to the query in the query runtime phase.

There is an average of 21 footprints per documents which could make a geographical index unapproachable or, at least, waste too much time [5]. This is the reason why we tried not to use a geographical index, since our system works fast enough.

2.3.2 *Query Process*

In this second phase, the queries are sent to the linguistic analysis module and to the *QA* module. Afterwards, our system sends every geographic reference got from the two previous modules to the geographic module in order to transform them to the *Yahoo!* unique identifier (*WOEID*). Finally, all these data are stored making a new *XML* file for each query (see Figure 3).

2.3.3 *Query Runtime*

In this third and last phase, the system gets all the *XML* query files, and together with the *XML* relevant filter files obtained from the search engine for each query are sent to the re-ranking module to do its work. Once it is finished, the result will be saved in a solution file.

2.4 Run Description

In the *NTCIR-GeoTime 9* it was allowed to send up to 5 different runs. Each of these runs had to have an end identifier which shows whether the narrative part of the query is being used (the run identifier ends with *DN*) or not (the run identifier ends with *D*). On the other hand, it was possible both to run the queries automatically and examining them by hand to extract characteristics. Our system is completed automatic with the description of all five runs as follows:

1. GETUA-EN-EN-05-D. It is our baseline, just using *Lucene* with *BM25* ranking function, lemmatized and a list of 571 stopwords (see Equation 1 or 2 with $\lambda = 1$).

2. GETUA-EN-EN-04-D. 10% of *GETUA-EN-EN-05-D* (the baseline) and the other 90% doing searches with the descriptive part of the query (see Equation 1 with $\lambda = 0.1$, $\alpha = 1$, $\beta = 1$).

3. GETUA-EN-EN-03-DN. 10% of the *GETUA-EN-EN-05-D* and the other 90% distributed as follows: 10% doing named entities searches and the remaining 90% doing geographical searches without regarding to temporal restrictions (see Equation 2 with $\lambda = 0.1$, $\alpha = 0.9$, $\beta = 1$).

4. GETUA-EN-EN-02-DN. 10% of the *GETUA-EN-EN-05-D* and the other 90% distributed as follows: 10% doing named entities searches and the remaining 90% doing temporal searches without regarding to geographical restrictions (see Equation 2 with $\lambda = 0.1$, $\alpha = 0.9$, $\beta = 0$).

5. GETUA-EN-EN-01-D. 10% of *GETUA-EN-EN-05-D* and the other 90% distributed as follows: 20% doing searches only with the descriptive part of the query and the remaining 80% applying techniques of *QA* (see Equation 1 with $\lambda = 0.1$, $\alpha = 0.2$, $\beta = 1$).

3. RESULTS

In this section, first, it is going to be shown the general system results. After that, the *QA* module results will be particularized.

3.1 General System Results

In both Figure 4 and 5 are showed the results of the search engine module (see Section 2.1.1) with regard to *NTCIR-9 GeoTime* partially and full relevant results, respectively. Over 55% of accuracy is obtained.

In Table 1 can be seen the result of our 5 runs. We can realize the different between the run 5 (only the search engine) and 1 (the complete system). An improvement is obtained, in part, by the linguistic analysis module (see Section 2.1.2) which is responsible of getting all temporal and entity data. On the other hand, watching runs 3 and 4, we will see that geographic module (see Section 2.1.5) could have more relevance than the previous one mentioned. Looking into the result of run 4 and 1, we can also assume that the *QA* module gets an outstanding enrichment. Finally, the re-ranking module (see Section 2.1.7) performance can be noticed watching the results between the run 5 (only the search engine) and the run 1 (re-ranking weight function).

RunName	MAP	Q	nDCG@10	nDCG@100	nDCG@1000
GETUA-EN-EN-01-D	0.2026	0.2159	0.2903	0.3081	0.4063
GETUA-EN-EN-02-DN	0.1610	0.1774	0.1712	0.2524	0.3629
GETUA-EN-EN-03-DN	0.1872	0.2021	0.2763	0.2786	0.3899
GETUA-EN-EN-04-D	0.1547	0.1730	0.2204	0.2409	0.3623
GETUA-EN-EN-05-D	0.1536	0.1740	0.3238	0.2567	0.3858

Table 1: Our runs average results

Figure 6 shows our system average performance for each individual run. There, we can appreciate how a basic search engine can be improved adding spatial and temporal dimension aware.

In Figure 7 it can be seen the Q measure system performance for every run on each individual topic. It should be noticed the individual performance in each query that our system gets. It can be seen some topics where the system gets a really good score and some other where the result must be improved.

3.2 QA Module Results

In order to evaluate the *QA* module importance, the 'ω_A' weight function has been used (see section 2.1.7). For this purpose, the value that got the best performance to variable β has been allocated and the variable α values have been modified between the range 0 - 1 with an increase of 0.1. The result can be seen in figure 8. As it can be seen in the figure 8, for the range of values between 0.2 and 0.9, results remain very close, which seems to say that the module *QA* is important although not crucial in the outcome.

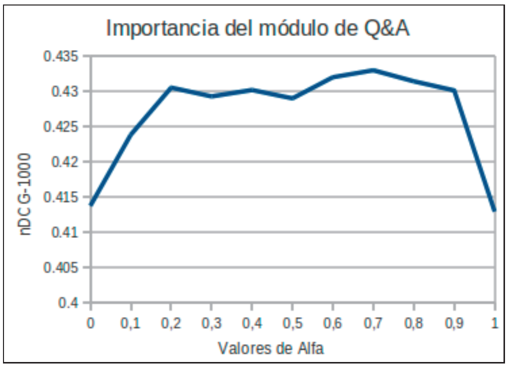

Figure 8: *QA* module importance

4. CONCLUSIONS

At this first approach to geo-temporal *IR* systems, we have started from a *IR* system and we have added geographical intelligence. In addition, we have used a naive implementation to tackle the temporal dimension. In spite of this, we can draw the conclusion showed below.

We have obtained the improvement from the run 1 to the 5 just using the descriptive part of the topic, since the narrative one introduced much noise. In the future it could be developed a module which will be able to extract and filter the information better from the narrative part.

The huge different performance between every topic could be due to the frequency of the topic terms in the corpus. As we mentioned previously, our system starts with the *Lucene* retrieval, which used the ranking function *Okapi BM25*. This function gives greater value to those terms that occur in fewer documents, the terms appear more frequently in a document, and the proportion of occurrences of a term in a document given the length of the latter [4]. If we analyse the topics individually, we will notice our system gets good results in those ones that have unusually terms. Consequently, we must process separately the topics with infrequent terms and the rest of them in order to get better performance.

As we have mentioned before, the *QA* module gets an outstanding enrichment, therefore, we are exploring different *QA* techniques to use in the future.

On the other hand, the temporal module has had significantly less importance in the evaluation than the geographical one, due to it, it has been developed in a simple way. We are working in upgrading our temporal module to procure more favourable results at the *NTCIR-10 GeoTime*, actually, our research group has developed a completed temporal *IR* system (*TIPSem*[5]) which will be added to our system.

Focusing in the geographical module, currently we have two active fronts. On the one hand, we are exploiting more metadata from *Yahoo! Placemaker*, such as the general geographical scope of the document. On the other hand, we intend to fully develop the geographic module to be independent of applications which are subject to restrictions of third parties.

5. ACKNOWLEDGMENTS

This research has been partially funded by the Spanish Government under project TEXTMESS 2.0 (TIN2009-13391-C04-01), and by the University of Alicante under project GRE10-33.

[5] http://gplsi.dlsi.ua.es/demos/TIMEE/

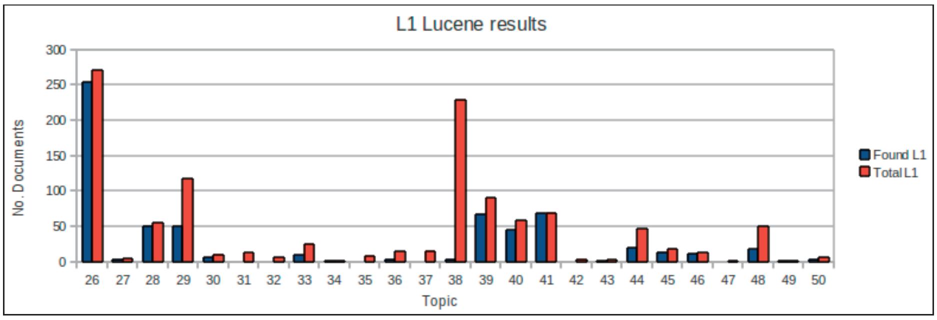

Figure 4: *Lucene* accuracy respect to the *NTCIR-9 GeoTime* partially relevant results

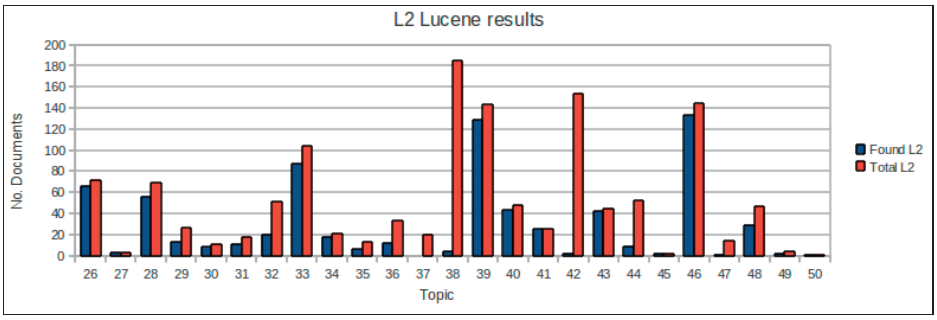

Figure 5: *Lucene* accuracy respect to the *NTCIR-9 GeoTime* full relevant results

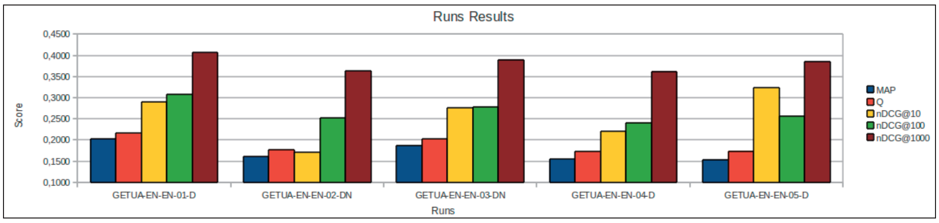

Figure 6: Average performance for each individual run

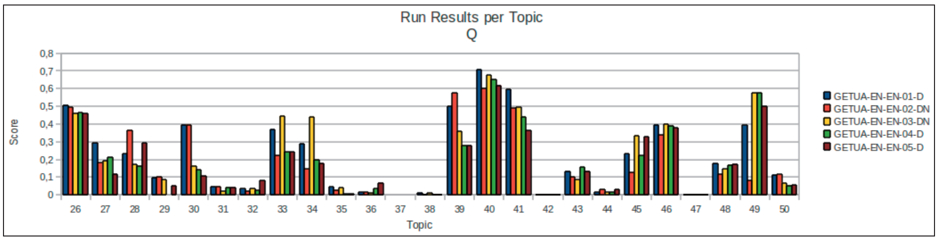

Figure 7: *Q* measure performance of every run in every topic

6. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, volume 463. Addison Wesley, 1999.

[2] F. Gey, R. Larson, J. Machado, and M. Yoshioka. *NTCIR9-GeoTime Overview - Evaluating Geographic and Temporal Search: Round 2, In this proceedings.* 2011.

[3] C. B. Jones and R. S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219–228, 2008.

[4] S. E. Robertson, S. Walker, and M. Hancock-Beaulieu. *Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive*, pages 199–210. NIST, 1998.

[5] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson. Spatio-textual indexing for geographical search on the web. *Advances in Spatial and Temporal Databases 9th International Symposium SSTD 2005*, 3633:218–235, 2005.