Handling Locations in Search Engine Queries

Bruno Martins, Mário J. Silva, Sérgio Freitas and Ana Paula Afonso
Faculdade de Ciências da Universidade de Lisboa
1749-016 Lisboa, Portugal
{bmartins,mjs,sfreitas,apa}@xldb.di.fc.ul.pt

ABSTRACT

This paper proposes simple techniques for handling place references in search engine queries, an important aspect of geographical information retrieval. We address not only the detection, but also the disambiguation of place references, by matching them explicitly with concepts at an ontology. Moreover, when a query does not reference any locations, we propose to use information from documents matching the query, exploiting geographic scopes previously assigned to these documents. Evaluation experiments, using topics from CLEF campaigns and logs from real search engine queries, show the effectiveness of the proposed approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Geographic IR, Text Mining, Query Processing

1. INTRODUCTION

Search engine queries are often associated with geographical locations, either explicitly (i.e. a location reference is given as part of the query) or implicitly (i.e. the location reference is not present in the query string, but the query clearly has a local intent [17]). One of the concerns of geographical information retrieval (GIR) lies in appropriately handling such queries, bringing better targeted search results and improving user satisfaction.

Nowadays, GIR is getting increasing attention. Systems that access resources on the basis of geographic context are starting to appear, both in the academic and commercial domains [4, 7]. Accurately and effectively detecting location references in search engine queries is a crucial aspect of these systems, as they are generally based on interpreting geographical terms differently from the others. Detecting locations in queries is also important for general-propose search engines, as this information can be used to improve ranking algorithms. Queries with a local intent are best answered with "localized" pages, while queries without any geographical references are best answered with "broad" pages [5].

Text mining methods have been successfully used in GIR to detect and disambiguate geographical references in text [9], or even to infer geographic scopes for documents [1, 13]. However, this body of research has been focused on processing Web pages and full-text documents. Search engine queries are more difficult to handle, in the sense that they are very short and with implicit and subjective user intents. Moreover, the data is also noisier and more versatile in form, and we have to deal with misspellings, multilingualism and acronyms. How to automatically understand what the user intended, given a search query, without putting the burden in the user himself, remains an open text mining problem.

Key challenges in handling locations over search engine queries include their detection and disambiguation, the ranking of possible candidates, the detection of false positives (i.e not all contained location names refer to geographical locations), and the detection of implied locations by the context of the query (i.e. when the query does not explicitly contain a place reference but it is nonetheless geographical). Simple named entity recognition (NER) algorithms, based on dictionary look-ups for geographical names, may introduce high false positives for queries whose location names do not constitute place references. For example the query "Denzel Washington" contains the place name "Washington," but the query is not geographical. Queries can also be geographic without containing any explicit reference to locations at the dictionary. In these cases, place name extraction and disambiguation does not give any results, and we need to access other sources of information.

This paper proposes simple and yet effective techniques for handling place references over queries. Each query is split into a triple $< what, relation, where >$, where *what* specifies the non-geographic aspect of the information need, *where* specifies the geographic areas of interest, and *relation* specifies a spatial relationship connecting *what* and *where*. When this is not possible, i.e. the query does not contain any place references, we try using information from documents matching the query, exploiting geographic scopes previously assigned to these documents.

Disambiguating place references is one of the most important aspects. We use a search procedure that combines textual patterns with geographical names defined at an ontology, and we use heuristics to disambiguate the discovered references (e.g. more important places are preferred). Disambiguation results in having the *where* term, from the triple above, associated with the most likely corresponding concepts from the ontology. When we cannot detect any locations, we attempt to use geographical scopes previously inferred for the documents at the top search results. By doing this, we assume that the most frequent geographical scope in the results should correspond to the geographical context implicit in the query.

This research was partially supported Fundação para a Ciência e Tecnologia, under grants POSI/SRI/40193/2001 and SFRH/BD/10757/2002.

GIR'06, August 10, 2006, Seattle, Washington.

Experiments with CLEF topics [4] and sample queries from a Web search engine show that the proposed methods are accurate, and may have applications in improving search results.

The rest of this paper is organized as follows. We first formalize the problem and describe related work to our research. Next, we describe our approach for handling place names in queries, starting with the general approach for disambiguating place references over textual strings, then presenting the method for splitting a query into a $< what, relation, where >$ triple, and finally discussing the technique for exploiting geographic scopes previously assigned to documents in the result set. Section 4 presents evaluation results. Finally, we give some conclusions and directions for future research.

2. CONCEPTS AND RELATED WORK

Search engine performance depends on the ability to capture the most likely meaning of a query as intended by the user [16]. Previous studies showed that a significant portion of the queries submitted to search engines are geographic [8, 14]. A recent enhancement to search engine technology is the addition of geographic reasoning, combining geographic information systems and information retrieval in order to build search engines that find information associated with given locations. The ability to recognize and reason about the geographical terminology, given in the text documents and user queries, is a crucial aspect of these geographical information retrieval (GIR) systems [4, 7].

Extracting and distinguishing different types of entities in text is usually referred to as Named Entity Recognition (NER). For at least a decade, this has been an important text mining task, and a key feature of the Message Understanding Conferences (MUC) [3]. NER has been successfully automated with near-human performance, but the specific problem of recognizing geographical references presents additional challenges [9]. When handling named entities with a high level of detail, ambiguity problems arise more frequently. Ambiguity in geographical references is bi-directional [15]. The same name can be used for more than one location (referent ambiguity), and the same location can have more than one name (reference ambiguity). The former has another twist, since the same name can be used for locations as well as for other class of entities, such as persons or company names (referent class ambiguity). Besides the recognition of geographical expressions, GIR also requires that the recognized expressions be classified and grounded to unique identifiers [11]. Grounding the recognized expressions (e.g. associating them to coordinates or concepts at an ontology) assures that they can be used in more advanced GIR tasks.

Previous works have addressed the tagging and grounding of locations in Web pages, as well as the assignment of geographic scopes to these documents [1, 7, 13]. This is a complementary aspect to the techniques described in this paper, since if we have the Web pages tagged with location information, a search engine can conveniently return pages with a geographical scope related to the scope of the query. The task of handling geographical references over documents is however considerably different from that of handling geographical references over queries. In our case, queries are usually short and often do not constitute proper sentences. Text mining techniques that make use of context information are difficult to apply for high accuracy.

Previous studies have also addressed the use of text mining and automated classification techniques over search engine queries [16, 10]. However, most of these works did not consider place references or geographical categories. Again, these previously proposed methods are difficult to apply to the geographic domain.

Gravano et. al. studied the classification of Web queries into two types, namely local and global [5]. They defined a query as local if its best matches on a Web search engine are likely to be local pages, such as "houses for sale." A number of classification algorithms have been evaluated using search engine queries. However, their experimental results showed that only a rather low precision and recall could be achieved. The problem addressed in this paper is also slightly different, since we are trying not only to detect local queries but also to disambiguate the local of interest.

Wang et. al. proposed to go further than detecting local queries, by also disambiguating the implicit local of interest [17]. The proposed approach works for both queries containing place references and queries not containing them, by looking for dominant geographic references over query logs and text from search results. In comparison, we propose simpler techniques based on matching names from a geographic ontology. Our approach looks for spatial relationships at the query string, and it also associates the place references to ontology concepts. In the case of queries not containing explicit place references, we use geographical scopes previously assigned to the documents, whereas Wang et. al. proposed to extract locations from the text of the top search results.

There are nowadays many geocoding, reverse-geocoding, and mapping services on the Web that can be easily integrated with other applications. Geocoding is the process of locating points on the surface of the Earth from alphanumeric addressing data. Taking a string with an address, a geocoder queries a geographical information system and returns interpolated coordinate values for the given location. Instead of computing coordinates for a given place reference, the technique described in this paper aims at assigning references to the corresponding ontology concepts. However, if each concept at the ontology contains associated coordinate information, the approach described here could also be used to build a geocoding service. Most of such existing services are commercial in nature, and there are no technical publications describing them.

A number of commercial search services have also started to support location-based searches. Google Local, for instance, initially required the user to specify a location qualifier separately from the search query. More recently, it added location look-up capabilities that extract locations from query strings. For example, in a search for "Pizza Seattle", Google Local returns "local results for pizza near Seattle, WA." However, the intrinsics of their solution are not published, and their approach also does not handle location-implicit queries. Moreover, Google Local does not take spatial relations into account.

In sum, there are already some studies on tagging geographical references, but Web queries pose additional challenges which have not been addressed. In this paper, we explain the proposed solutions for the identified problems.

3. HANDLING QUERIES IN GIR SYSTEMS

Most GIR queries can be parsed to $< what, relation, where >$ triple, where the *what* term is used to specify the general non-geographical aspect of the information need, the *where* term is used to specify the geographical areas of interest, and the *relation* term is used to specify a spatial relationship connecting *what* and *where*. While the *what* term can assume any form, in order to reflect any information need, the *relation* and *where* terms should be part of a controlled vocabulary. In particular, the *relation* term should refer to a well-known geographical relation that the underlying GIR system can interpret (e.g. *"near"* or *"contained at"*), and the *where* term should be disambiguated into a set of unique identifiers, corresponding to concepts at the ontology.

Different systems can use alternative schemes to take input queries from the users. Three general strategies can be identified, and GIR systems often support more than one of the following schemes:

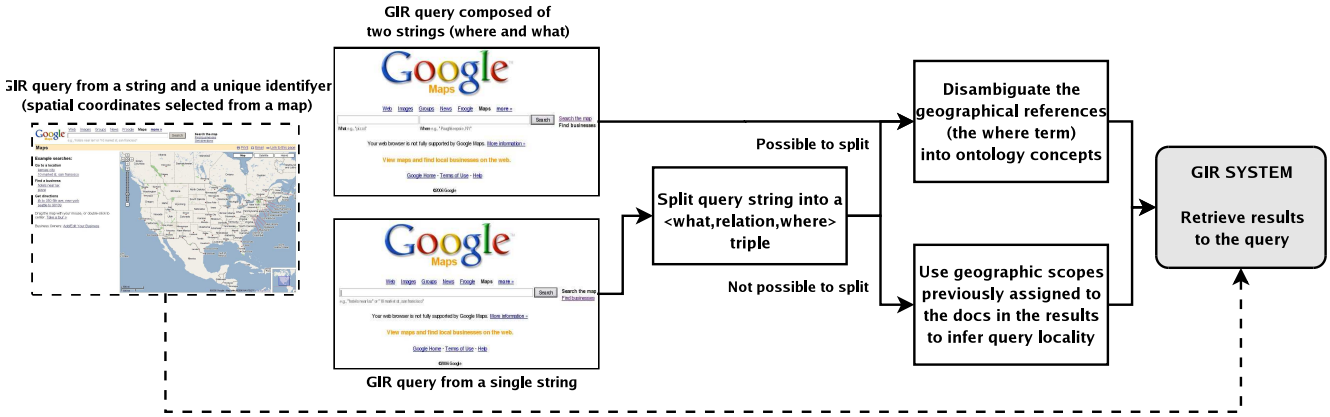

Figure 1: Strategies for processing queries in Geographical Information Retrieval systems.

1. Input to the system is a textual query string. This is the hardest case, since we need to separate the query into the three different components, and then we need to disambiguate the *where* term into a set of unique identifiers.

2. Input to the system is provided in two separate strings, one concerning the *what* term, and the other concerning the *where*. The *relation* term can be either fixed (e.g. always assume the *"near"* relation), specified together with the *where* string, or provided separately from the users from a set of possible choices. Although there is no need for separating query string into the different components, we still need to disambiguate the *where* term into a set of unique identifiers.

3. Input to the system is provided through a query string together with an unambiguous description of the geographical area of interest (e.g. a sketch in a map, spatial coordinates or a selection from a set of possible choices). No disambiguation is required, and therefore the techniques described in this paper do not have to be applied.

The first two schemes depend on place name disambiguation. Figure 1 illustrates how we propose to handle geographic queries in these first two schemes. A common component is the algorithm for disambiguating place references into corresponding ontology concepts, which is described next.

3.1 From Place Names to Ontology Concepts

A required task in handling GIR queries consists of associating a string containing a geographical reference with the set of corresponding concepts at the geographic ontology. We propose to do this according to the pseudo-code listed at Algorithm 1.

The algorithm considers the cases where a second (or even more than one) location is given to qualify a first (e.g. "Paris, France"). It makes recursive calls to match each location, and relies on hierarchical part-of relations to detect if two locations share a common hierarchy path. One of the provided locations should be more general and the other more specific, in the sense that there must exist a part-of relationship among the associated concepts at the ontology (either direct or transitive). The most specific location is a sub-region of the most general, and the algorithm returns the most specific one (i.e. for "Paris, France" the algorithm returns the ontology concept associated with Paris, the capital city of France).

We also consider the cases where a geographical type expression is used to qualify a given name (e.g. "city of Lisbon" or "state of New York"). For instance the name "Lisbon" can correspond to many different concepts at a geographical ontology, and type

Algorithm 1 Matching a place name with ontology concepts

Require: O = A geographic ontology
Require: GN = A string with the geographic name to be matched
L = An empty list
$INDEX$ = The position in GN for the first occurrence of a comma, semi-colon or bracket character
if $INDEX$ is defined **then**
  GN_1 = The substring of GN from position 0 to $INDEX$
  GN_2 = The substring of GN from $INDEX+1$ to $length(GN)$
  $L_1 = Algorithm1(O, GN_1)$
  $L_2 = Algorithm1(O, GN_2)$
  for each C_1 in L_1 **do**
    for each C_2 in L_2 **do**
      if C_1 is an ancestor of C_2 at O **then**
        L = The list L after adding element C_2
      else if C_1 is a descendant of C_2 at O **then**
        L = The list L after adding element C_1
      end if
    end for
  end for
else
  GN = The string GN after removing case and diacritics
  if GN contains a geographic type qualifier **then**
    T = The substring of GN containing the type qualifier
    GN = The substring of GN with the type qualifier removed
    L = The list of concepts from O with name GN and type T
  else
    L = The list of concepts from O with name GN
  end if
end if
return The list L

qualifiers can provide useful information for disambiguation. The considered type qualifiers should also described at the ontologies (e.g. each geographic concept should be associated to a type that is also defined at the ontology, such as country, district or city).

Ideally, the geographical reference provided by the user should be disambiguated into a single ontology concept. However, this is not always possible, since the user may not provide all the required information (i.e. a type expression or a second qualifying location). The output is therefore a list with the possible concepts being referred to by the user. In a final step, we propose to sort this list, so that if a single concept is required as output, we can use the one that is ranked higher. The sorting procedure reflects the likelihood of each concept being indeed the one referred to. We propose to rank concepts according to the following heuristics:

1. The geographical type expression associated with the ontology concept. For the same name, a country is more likely to be referenced than a city, and in turn a city more likely to be referenced than a street.

2. Number of ancestors at the ontology. Top places at the ontology tend to be more general, and are therefore more likely to be referenced in search engine queries.

3. Population count. Highly populated places are better known, and therefore more likely to be referenced in queries.

4. Population counts from direct ancestors at the ontology. Subregions of highly populated places are better known, and also more likely to be referenced in search engine queries.

5. Occurrence frequency over Web documents (e.g. Google counts) for the geographical names. Places names that occur more frequently over Web documents are also more likely to be referenced in search engine queries.

6. Number of descendants at the ontology. Places that have more sub-regions tend to be more general, and are therefore more likely to be mentioned in search engine queries.

7. String size for the geographical names. Short names are more likely to be mentioned in search engine queries.

Algorithm 1, plus the ranking procedure, can already handle GIR queries where the *where* term is given separately from the *what* and *relation* terms. However, if the query is given in a single string, we require the identification of the associated $< what, relation, where >$ triple, before disambiguating the *where* term into the corresponding ontology concepts. This is described in the following Section.

3.2 Handling Single Query Strings

Algorithm 2 provides the mechanism for separating a query string into a $< what, relation, where >$ triple. It uses Algorithm 1 to find the *where* term, disambiguating it into a set of ontology concepts.

The algorithm starts by tokenizing the query string into individual words, also taking care of removing case and diacritics. We have a simple tokenizer that uses the space character as a word delimiter, but we could also have a tokenization approach similar to the proposal of Wang et. al. which relies on Web occurrence statistics to avoid breaking collocations [17]. In the future, we plan on testing if this different tokenization scheme can improve results.

Next, the algorithm tests different possible splits of the query, building the *what*, *relation* and *where* terms through concatenations of the individual tokens. The *relation* term is matched against a list of possible values (e.g. *"near," "at," "around,"* or *"south of"*), corresponding to the operators that are supported by the GIR system. Note that is also the responsibility of the underlying GIR system to interpret the actual meaning of the different spatial relations. Algorithm 1 is used to check whether a *where* term constitutes a geographical reference or not. We also check if the last word in the *what* term belongs to a list of exceptions, containing for instance first names of people in different languages. This ensures that a query like "Denzel Washington" is appropriately handled.

If the algorithm succeeds in finding valid *relation* and *where* terms, then the corresponding triple is returned. Otherwise, we return a triple with the *what* term equaling the query string, and the *relation* and *where* terms set as empty. If the entire query string constitutes a geographical reference, we return a triple with the *what* term set to empty, the *where* term equaling the query string, and the *relation* term set the "DEFINITION" (i.e. these queries should be answered with information about the given place references). The algorithm also handles query strings where more than one geographical reference is provided, using "and" or an equivalent preposition, together with a recursive call to Algorithm 2. A query like "Diamond trade in Angola and South Africa" is

Algorithm 2 Get $< what, relation, where >$ from a query string

Require: O = A geographical ontology
Require: Q = A non-empty string with the query
Q = The string Q after removing case and diacritics
$TOKENS[0..N]$ = An array of strings with the individual words of Q
N = The size of the $TOKENS$ array
for $INDEX = 0$ to N **do**
  if $INDEX = 0$ **then**
    $WHAT$ = Concatenation of $TOKENS[0..INDEX-1]$
    $LASTWHAT = TOKENS[INDEX-1]$
  else
    $WHAT$ = An empty string
    $LASTWHAT$ = An empty string
  end if
  $WHERE$ = Concatenation of $TOKENS[INDEX..N]$
  $RELATION$ = An empty string
  for $INDEX_2 = INDEX$ to $N-1$ **do**
    $RELATION_2$ = Concatenation of $TOKENS[INDEX..INDEX_2]$
    if $RELATION_2$ is a valid geographical relation **then**
      $WHERE$ = Concatenation of $S[INDEX_2+1..N]$
      $RELATION = RELATION_2$;
    end if
  end for
  if $RELATION$ = empty AND $LASTWHAT$ in an exception **then**
    $TESTGEO = FALSE$
  else
    $TESTGEO = TRUE$
  end if
  if $TESTGEO$ AND $Algorithm1(WHERE)$ <> EMPTY **then**
    if $WHERE$ ends with "AND SURROUNDINGS" **then**
      $RELATION$ = The string "NEAR";
      $WHERE$ = The substring of $WHERE$ with "AND SURROUNDINGS" removed
    end if
    if $WHAT$ ends with "AND" or similar) **then**
      $< WHAT, RELATION, WHERE_2 >= Algorithm2(WHAT)$
      $WHERE$ = Concatenation of $WHERE$ with $WHERE_2$
    end if
    if $RELATION$ = An empty string **then**
      if $WHAT$ = An empty string **then**
        $RELATION$ = The string "DEFINITION"
      else
        $RELATION$ = The string "CONTAINED-AT"
      end if
    end if
  else
    $WHAT$ = The string Q
    $WHERE$ = An empty string
    $RELATION$ = An empty string
  end if
end for
return $< WHAT, RELATION, WHERE >$

therefore appropriately handled. Finally, if the geographical reference in the query is complemented with an expression similar to *"and its surroundings,"* the spatial relation (which is assumed to be "CONTAINED-AT" if none is provided) is changed to "NEAR".

3.3 From Search Results to Query Locality

The procedures given so far are appropriate for handling queries where a place reference is explicitly mentioned. However, the fact that a query can be associated with a geographical context may not be directly observable in the query itself, but rather from the results returned. For instance, queries like "recommended hotels for SIGIR 2006" or "SeaFair 2006 lodging" can be seen to refer to the city of Seattle. Although they do not contain an explicit place reference, we expect results to be about hotels in Seattle.

In the cases where a query does not contain place references, we start by assuming that the top results from a search engine represent the most popular and correct context and usage for the query. We

Topic	What	Relation	Where	TGN concepts	ML concepts
Vegetable Exporters of Europe	Vegetable Exporters	CONTAINED-AT	Europe	1	1
Trade Unions in Europe	Trade Unions	CONTAINED-AT	Europe	1	1
Roman cities in the UK and Germany	Roman cities	CONTAINED-AT	UK and Germany	6	2
Cathedrals in Europe	Cathedrals	CONTAINED-AT	Europe	1	1
Car bombings near Madrid	Car bombings	NEAR	Madrid	14	2
Volcanos around Quito	Volcanos	NEAR	Quito	4	1
Cities within 100km of Frankfurt	Cities	NEAR	Frankfurt	3	1
Russian troops in south(ern) Caucasus	Russian troops in south(ern)	CONTAINED-AT	Caucasus	2	1
Cities near active volcanoes	(This topic could not be appropriately handled – the "relation" and "where" terms are returned empty)				
Japanese rice imports	(This topic could not be appropriately handled – the "relation" and "where" terms are returned empty)				

Table 1: Example topics from the GeoCLEF evaluation campaigns and the corresponding $< what, relation, where >$ triples.

then propose to use the distributional characteristics of geographical scopes previously assigned to the documents corresponding to these top results. In a previous work, we presented a text mining approach for assigning documents with corresponding geographical scopes, defined at an ontology, that worked as an offline preprocessing stage in a GIR system [13]. This pre-processing step is a fundamental stage of GIR, and it is reasonable to assume that this kind of information would be available on any system. Similarly to Wang et. al., we could also attempt to process the results on-line, in order to detect place references in the documents [17]. However, a GIR system already requires the offline stage.

For the top *N* documents given at the results, we check the geographic scopes that were assigned to them. If a significant portion of the results are assigned to the same scope, than the query can be seen to be related to the corresponding geographic concept. This assumption could even be relaxed, for instance by checking if the documents belong to scopes that are hierarchically related.

4. EVALUATION EXPERIMENTS

We used three different ontologies in evaluation experiments, namely the Getty thesaurus of geographic names (TGN) [6] and two specific resources developed at our group, here referred to as the PT and ML ontologies [2]. TGN and ML include global geographical information in multiple languages (although TGN is considerably larger), while the PT ontology focuses on the Portuguese territory with a high detail. Place types are also different across ontologies, as for instance PT includes street names and postal addresses, whereas ML only goes to the level of cities. The reader should refer to [2, 6] for a complete description of these resources.

Our initial experiments used Portuguese and English topics from the GeoCLEF 2005 and 2006 evaluation campaigns. Topics in GeoCLEF correspond to query strings that can be used as input to a GIR system [4]. ImageCLEF 2006 also included topics specifying place references, and participants were encouraged to run their GIR systems on them. Our experiments also considered this dataset. For each topic, we measured if Algorithm 2 was able to find the corresponding $< what, relation, where >$ triple. The ontologies used in this experiment were the TGN and ML, as topics were given in multiple languages and covered the whole globe.

Dataset	Number of Queries	Correct Triples		Time per Query	
		ML	TGN	ML	TGN
GeoCLEF05 EN	25	19	20		
GeoCLEF05 PT	25	20	18	288.1	334.5
GeoCLEF06 EN	32	28	19	msec	msec
GeoCLEF06 PT	25	23	11		
ImgCLEF06 EN	24	16	18		

Table 2: Summary of results over CLEF topics.

Table 1 illustrates some of the topics, and Table 2 summarizes the obtained results. The tables show that the proposed technique adequately handles most of these queries. A manual inspection of the ontology concepts that were returned for each case also revealed that the *where* term was being correctly disambiguated. Note that the TGN ontology indeed added some ambiguity, as for instance names like "Madrid" can correspond to many different places around the globe. It should also be noted that some of the considered topics are very hard for an automated system to handle. Some of them were ambiguous (e.g. in "Japanese rice imports," the query can be said to refer either rice imports in Japan or imports of Japanese rice), and others contained no direct geographical references (e.g. cities near active volcanoes). Besides these very hard cases, we also missed some topics due to their usage of place adjectives and specific regions that are not defined at the ontologies (e.g. environmental concerns around the Scottish Trossachs).

In a second experiment, we used a sample of around 100,000 real search engine queries. The objective was to see if a significant number of these queries were geographical in nature, also checking if the algorithm did not produce many mistakes by classifying a query as geographical when that was not the case. The Portuguese ontology was used in this experiment, and queries were taken from the logs of a Portuguese Web search engine available at www.tumba.pt. Table 3 summarizes the obtained results. Many queries were indeed geographical (around 3.4%, although previous studies reported values above 14% [8]). A manual inspection showed that the algorithm did not produce many false positives, and the geographical queries were indeed correctly split into correct $< what, relation, where >$ triple. The few mistakes we encountered were related to place names that were more frequently used in other contexts (e.g. in "Teófilo Braga" we have the problem that "Braga" is a Portuguese district, and "Teófilo Braga" was a well known Portuguese writer and politician). The addition of more names to the exception list can provide a workaround for most of these cases.

	Value
Num. Queries	110,916
Num. Queries without Geographical References	107,159 (96.6%)
Num. Queries with Geographical References	3,757 (3.4%)

Table 3: Results from an experiment with search engine logs.

We also tested the procedure for detecting queries that are implicitly geographical with a small sample of queries from the logs. For instance, for the query "Estádio do Dragão" (e.g. home stadium for a soccer team from Porto), the correct geographical context can be discovered from the analysis of the results (more than 75% of the top 20 results are assigned with the scope "Porto"). For future work, we plan on using a larger collection of queries to evaluate this aspect. Besides queries from the search engine logs, we also plan on using the names of well-known buildings, monuments and other landmarks, as they have a strong geographical connotation.

Finally, we also made a comparative experiment with 2 popular geocoders, Maporama and Microsoft's Mappoint. The objective was to compare Algorithm 1 with other approaches, in terms of being able to correctly disambiguate a string with a place reference.

Civil Parishes from Lisbon	Maporama	Mappoint	Ours
Coded refs. (out of 53)	9 (16.9%)	30 (56,6%)	15 (28.3%)
Avg. Time per ref. (msec)	506.23	1235.87	143.43
Civil Parishes from Porto	Maporama	Mappoint	Ours
Coded refs. (out of 15)	0 (0%)	2 (13,3%)	5 (33.3%)
Avg. Time per ref. (msec)	514.45	991.88	132.14

Table 4: Results from a comparison with geocoding services.

The Portuguese ontology was used in this experiment, taking as input the names of civil parishes from the Portuguese municipalities of Lisbon and Porto, and checking if the systems were able to disambiguate the full name (e.g. "Campo Grande, Lisboa" or "Foz do Douro, Porto") into the correct geocode. We specifically measured whether our approach was better at unambiguously returning geocodes given the place reference (i.e. return the single correct code), and providing results rapidly. Table 4 shows the obtained results, and the accuracy of our method seems comparable to the commercial geocoders. Note that for Maporama and Mappoint, the times given at Table 4 include fetching results from the Web, but we have no direct way of accessing the geocoding algorithms (in both cases, fetching static content from the Web servers takes around 125 milliseconds). Although our approach cannot unambiguously return the correct geocode in most cases (only 20 out of a total of 68 cases), it nonetheless returns results that a human user can disambiguate (e.g. for "Madalena, Lisboa" we return both a street and a civil parish), as opposed to the other systems that often did not produce results. Moreover, if we consider the top geocode according to the ranking procedure described in Section 3.1, or if we use a type qualifier in the name (e.g. "civil parish of Campo Grande, Lisboa"), our algorithm always returns the correct geocode.

5. CONCLUSIONS

This paper presented simple approaches for handling place references in search engine queries. This is a hard text mining problem, as queries are often ambiguous or underspecify information needs. However, our initial experiments indicate that for many queries, the referenced places can be determined effectively. Unlike the techniques proposed by Wang et. al. [17], we mainly focused on recognizing spatial relations and associating place names to ontology concepts. The proposed techniques were employed in the prototype system that we used for participating in GeoCLEF 2006. In queries where a geographical reference is not explicitly mentioned, we propose to use the results for the query, exploiting geographic scopes previously assigned to these documents. In the future, we plan on doing a careful evaluation of this last approach. Another idea that we would like to test involves the integration of a spelling correction mechanism [12] into Algorithm 1, so that incorrectly spelled place references can be matched to ontology concepts.

The proposed techniques for handling geographic queries can have many applications in improving GIR systems or even general purpose search engines. After place references are appropriately disambiguated into ontology concepts, a GIR system can use them to retrieve relevant results, through the use of appropriate index structures (e.g. indexing the spatial coordinates associated with ontology concepts) and provided that the documents are also assigned to scopes corresponding to ontology concepts. A different GIR strategy can involve query expansion, by taking the *where* terms from the query and using the ontology to add names from neighboring locations. In a general purpose search engine, and if a local query is detected, we can forward users to a GIR system, which should be better suited for properly handling the query. The regular Google search interface already does this, by presenting a link to Google Local when it detects a geographical query.

6. REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging Web content. In *Proceedings of SIGIR-04, the 27th Conference on research and development in information retrieval*, 2004.

[2] M. Chaves, M. J. Silva, and B. Martins. A Geographic Knowledge Base for Semantic Web Applications. In *Proceedings of SBBD-05, the 20th Brazilian Symposium on Databases*, 2005.

[3] N. A. Chinchor. Overview of MUC-7/MET-2. In *Proceedings of MUC-7, the 7th Message Understanding Conference*, 1998.

[4] F. Gey, R. Larson, M. Sanderson, H. Joho, and P. Clough. GeoCLEF: the CLEF 2005 cross-language geographic information retrieval track. In *Working Notes for the CLEF 2005 Workshop*, 2005.

[5] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. Categorizing Web queries according to geographical locality. In *Proceedings of CIKM-03, the 12th Conference on Information and knowledge management*, 2003.

[6] P. Harpring. Proper words in proper places: The thesaurus of geographic names. *MDA Information*, 3, 1997.

[7] C. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. van Kreveld, and R. Weibel. Spatial information retrieval and geographical ontologies: An overview of the SPIRIT project. In *Proceedings of SIGIR-02, the 25th Conference on Research and Development in Information Retrieval*, 2002.

[8] J. Kohler. Analyzing search engine queries for the use of geographic terms, 2003. (MSc Thesis).

[9] A. Kornai and B. Sundheim, editors. *Proceedings of the NAACL-HLT Workshop on the Analysis of Geographic References*, 2003.

[10] Y. Li, Z. Zheng, and H. Dai. KDD CUP-2005 report: Facing a great challenge. *SIGKDD Explorations*, 7, 2006.

[11] D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, D. Maynard, and H. Cunningham. Experiments with geographic knowledge for information extraction. In *Proceedings of the NAACL-HLT Workshop on the Analysis of Geographic References*, 2003.

[12] B. Martins and M. J. Silva. Spelling correction for search engine queries. In *Proceedings of EsTAL-04, España for Natural Language Processing*, 2004.

[13] B. Martins and M. J. Silva. A graph-ranking algorithm for geo-referencing documents. In *Proceedings of ICDM-05, the 5th IEEE International Conference on Data Mining*, 2005.

[14] L. Souza, C. J. Davis, K. Borges, T. Delboni, and A. Laender. The role of gazetteers in geographic knowledge discovery on the web. In *Proceedings of LA-Web-05, the 3rd Latin American Web Congress*, 2005.

[15] E. Tjong, K. Sang, and F. D. Meulder. Introduction to the CoNLL-2003 shared task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003, the 7th Conference on Natural Language Learning*, 2003.

[16] D. Vogel, S. Bickel, P. Haider, R. Schimpfky, P. Siemen, S. Bridges, and T. Scheffer. Classifying search engine queries using the Web as background knowledge. *SIGKDD Explorations Newsletter*, 7(2):117–122, 2005.

[17] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. Detecting dominant locations from search queries. In *Proceedings of SIGIR-05, the 28th Conference on Research and development in information retrieval*, 2005.