

Building a Geographically Intelligent News Search Utility

Garima Sahai
Masters student, Comp. Sc.
Stanford University
CA, USA
gsahai@stanford.edu

Chiu-Ki Chan
Masters Student, Comp. Sc.
Stanford University
CA, USA
ckchan@stanford.edu

ABSTRACT

News search is very widespread nowadays. Almost all online news sources have their own search utility/use a search engine. Most of the news items have geographical context but very few news search engines exploit this. Good search for location specific news or items with geographical bias has to be done either by looking for a local newspaper or by using regional sections of an international newspaper. The concept of building geographical knowledge into the domain of news is still relatively new. In this paper, we discuss and analyze the means and use of incorporating geographical ontology knowledge into a news search engine. Our aim is to return more relevant results to a user querying for location related news. We use a map ontology provider to give us information about the location and use that to try and improve our search results. The underlying search is done using a meta-search over multiple news sites to get wider or geographically more diverse yet detailed source of news. We look at the various aspects of this approach and the tradeoffs involved and show how even a simple geographic expansion can lead to improvement in results.

Keywords

Meta-search, Map Ontology, Relevancy, Recall, Result Set, Query Expansion, Ranking, Term Frequency, Document Frequency, Boost Weight.

1. INTRODUCTION

Information retrieval based on geographic criteria is a fairly common task. Examples include travelers who wish to learn what opportunities exist in a destination,

students who are studying another part of the world, intelligence analysts preparing a report on a given area, business planners investigating an area for expansion, or government planners who wish to focus on environmental or economic impact in a specific region.

Ray R. Larson [1] discusses geographic information retrieval and spatial mapping. The paper provides some good insights into the types of geographic and spatial queries and the characteristics and problems associated with them.

Geographical information has been useful for web navigation and tried in different ways as discussed by Kevin S. McCurley [2]. He describes the implementation of a navigational tool that allows the user to browse web pages by their geographic proximity rather than their hyperlink proximity.

Buyukkokten et al [3] studied the use of several geographic keys for the purpose of assigning site-level geographic context. By analyzing "whois" records, they built a database that relates IP addresses and hostnames to approximate physical locations. By combining this information with the hyperlink structure of the web, they were able to make inferences about geography of the web at the granularity of a site. For example, they were able to confirm from their study that the New York Times has a more global reputation than the San Francisco Chronicle, based on the fact that is linked to from a more geographically diverse set of sites.

Aforementioned is some of the literature which is a very good resource for the general concept of Geographic Search. Combining geographical knowledge with news search is still a new idea and relatively unexplored. In this paper, we try and adapt some of these techniques and ideas to news search.

News is a very popular search domain and is something which is done on a repeated and regular basis. Search engines for news are important because

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

of the huge size and the frequent updates in the information sources. We found this category a good one for geographic information retrieval as typically most of the news sources and data have remarkable geographical components. For instance, news articles have a location from where they are reported, location of event occurrences, locations to which the article may be relevant etc. Also, people like to get news based on geographic entities a lot of times, say for getting the reactions in different parts of the world for a certain event, trying to see occurrences of similar events like earthquakes in vicinity of a given location, view an activity like use of the Euro currency in Paris in a geographically bigger sense i.e. say in context of France and other European countries.

The way we try and exploit this is to expand a location term hierarchically based on a already stored map ontology. So, a city is viewed as a child entity in relation to the province/state and the country it belongs to. We also play around with ranking such that the province/state is given more weight being an immediate parent than the country. Given the constraints of time, the expansion technique is pretty basic and simple, but enough to achieve our goal to explore the possibility of applying a simple geographic expansion technique to news search.

We combine data from different news sources using meta-search to have a better resource database while avoiding geographical bias of any single source. The details and tradeoffs of this are discussed in Section 2. The construction of map ontology and its use for location expansion is discussed in Section 3. Section 4 discusses the indexer and the ranking system used by the system to rank the query results.

An overview of the integrated search utility is given in Section 5. Section 6 talks about the experiments and user study we conducted. The results and analysis of the same are shown and discussed in Section 7. Section 8 talks about the extensions and future work possible in this area. We summarize in Section 9 with our conclusions.

All implementation was done in Java 1.4.

2. News Meta-Search

2.1 News Sources

An important decision while doing meta-search is deciding which sources to use as the input sources for

the meta-search engine. We wanted news sources which were of an international nature so that they don't introduce geographical bias.

The tradeoffs involved in this choice were:

1. A more global paper covered larger geographical region but had lesser in depth stories about each region. On the other hand, a regional or a country based paper contained more stories about the region but then was geographically biased.
2. There are a lot of news sources available online. Picking up a large number of these would definitely increase the search space but then we would inherently introduce bias by covering some parts of the world more than others. Also, it would involve more time processing each query over all these sites and though speed was not our criteria, we did want a reasonable performance.

We thus chose 3 very well known sources, BBC [4], CNN [5] and UsaToday [6], all of which had a huge and up-to-date collection of news stories and a relatively fast search utility. Besides, access to the news stories and search engines of these papers was free and easy. Though UsaToday is not nearly as international as the other two, it did give us a chance to observe the influence of a somewhat regional paper.

2.2 Querying the news Sources

After deciding the sources, we needed to figure out a way to get back relevant articles from each of them, given a query.

Since each source had its own search utility, we passed on the user query to each search engine and then parsed the HTML result pages to collect the combined results.

We had a choice between using a general HTML parser to parse the pages automatically, or do the same manually for each case. An automatic parser is relatively robust to changes in format and display and allows addition of more sources to the system in an easier fashion. On the other hand, individual manual parsing is more specific to these sources and gets us more accurate data since it handles each case specifically. For instance, a common technique used by automatic parsers is to label the largest size table

on a news page as the body of the article. While this is a good approximation, we wanted to be more specific in our case so that later our indexing based on geographical terms would be precise too. We wanted to be as exact as possible in extracting titles, summaries, links, dates and bodies of articles from each news page.

Our need for accuracy in labeling parts of the article prompted us to do manual HTML parsing for each of the sources and this was manageable due to the small size of our source set (3 news sources). Besides, all the sources are standard and established ones, so we didn't need to worry about frequent changes in their page layouts/formats.

3. Building Map Ontology

3.1 Source of Ontology

This was a difficult choice too as there are abundant sources of geographic/map information on the web. Some of the very detailed and popular ones are the Getty Thesaurus of Geographic Names [7], GEOnet Names server [8], etc. These databases are very rich and detailed but on exploring them we came across important issues in using geographic names, some of which are also noted by Ray R. Larson [1]:

1. Names are not unique: San Jose is a common city name throughout Central and South America, as well as in California. On an average, a rich database would return a lot of hits given a city name.
2. Places change size, shape and names over time.
3. Spelling variations: Local names for a region may differ from common English forms, and there may be variations in the spelling of a name over time (Peking, Beijing).

These issues involve separate research on their own. To avoid complicating matters and to get ahead with our plan to show the usefulness of search based on geographic context for news items, we chose a relatively simple but suitable database from the site www.mapplanet.com [9]. Though this is a relatively small database and does not contain information about a lot of smaller towns across the world, it is pretty good at locating important places. This reduced the hit set size for any geographic name drastically by returning only well-known or reasonably populated regions. This may not be the right thing for a real

production system, but it was just the right kind for our test study.

3.2 Ontology structure and storage.

The mapplanet database facilitated the creation of a hierarchical ontology structure we wished to use for our location expansion. The towns/cities in their database were linked to their parents i.e. the state/province and the country they belonged to. There was an alphabetical index on the city names. We simply downloaded these index pages and saved them as files per alphabet. This served as our city names database. To expand states/provinces to their country names, we preprocessed all the data in the mapplanet database to create a Hashtable mapping each state name to its parent country. For multiple states having the same names, we used a list of countries mapped to the same state name.

Our final ontology was the simple hierarchy:

City->state/province-> Country

3.3 Querying the ontology

Once our database was ready, querying it was straightforward. Given a city name, we would scan the corresponding alphabetical index to retrieve the list of state and country names corresponding to the city. For the state queries, we would use the hashtable to return the list of countries corresponding to that state name.

The only issue was which expansion to apply when there were multiple parents for a given location name (the case where multiple locations shared the same name)? We decided to expand using the most-important city/state first. But this involved defining relative "importance":

For multiple cities having the same name, we ranked them on the basis of their population unless it was a capital in some case. A capital was considered most important. For instance, if Paris is the capital of France, is also a city in England with population 100k, another city in say Russia with population 10k; our system would rank Paris:France before Paris:England and the last one would be Paris:Russia. There can be other better ways to rank results but using population as a measure of importance was not too bad an estimate as we shall see.

For multiple provinces with same names, we ranked them on the basis of the number of cities within those

states. Again, the criteria can be improved but it made sense to call a province important if it contained greater number of cities.

We returned the list of parents sorted by “importance” for the location query to our map ontology.

4. Indexing and Ranking

4.1 Indexer

One of the major tasks in meta-search engines is merging results retrieved from different sources in a meaningful way. Since our idea was to incorporate geographic expansion and emphasis, we needed to do index the articles based on our geographic criteria. We also required some sort of re-ranking on the retrieved set which could show the influence of our technique when compared to any standard merge. For this, we needed to index stuff on our own and then use our own set of parameters to rank them.

We used the already existing Lucene [10] text-search system for indexing purposes. Lucene was easily available and had a good API to suit our needs. We indexed the title, summary and body of the retrieved news articles. We did not store the body though to save space as Lucene provides this feature where you can choose to index some text and yet not store it. Also, Lucene supports other features like stop-word filtering. We also used Porter’s stemmer [11] to stem words to provide more meaningful search.

Since all the retrieval & indexing was done real time per query, we needed to take care of the fact that different queries being issued at the same time did not end up overwriting the same index. So, as a simple solution, we created different indexes per query to support multiple queries simultaneously. The index directory was cleaned from time to time to delete old indices which were no longer needed.

4.2 Ranking

Lucene has a ranking scheme based on term frequencies and document frequencies (tf-idf). Its scoring algorithm uses the following formula:

$$\text{score}_d = \frac{\text{sum}_t(\text{tf}_q * \text{idf}_t / \text{norm}_q * \text{tf}_d * \text{idf}_t / \text{norm}_d_t * \text{boost}_t) * \text{coord}_q_d}{\text{norm}_d_t * \text{boost}_t}$$

where:

score_d : score for document d

sum_t : sum for all terms t

tf_q : the square root of the frequency of t in the query

tf_d : the square root of the frequency of t in d

idf_t : $\log(\text{numDocs}/\text{docFreq}_t + 1) + 1.0$

numDocs : number of documents in index

docFreq_t : number of documents containing t

norm_q : $\sqrt{\text{sum}_t((\text{tf}_q * \text{idf}_t)^2)}$

norm_d_t : square root of number of tokens in d in the same field as t

boost_t : the user-specified boost for term t

coord_q_d : number of terms in both query and document / number of terms in query

The coordination factor gives an AND-like boost to documents that contain e.g., all three terms in a three word query over those that contain just two of the words.

The optional boost factor (default = 1.0) for certain terms is used to increase or decrease the ranking of the documents that contain those specific words.

We played around with these boost factors for geographic terms to add geographic relevance. The user typed Location field had the same weight as a normal query term (1.0). After location expansion using our map ontology, we boosted those additional terms (parents) in a decaying fashion: if the city name was typed by the user, we gave the state name a boost of ¼ of city and the country name a boost of ¼ of state.

Also, words in the Title and Summary were given more weight than the words in the body based on their relative importance.

5. Integrated System Methodology

Once we had the main components of the system working, we just needed a framework integrating them. The complete system had:

User Interface: This was a simple web-search interface with a query box and location boxes (for city, state and country). The user just needed to fill in whichever component of the location he knew. See <http://www.stanford.edu/~ckchan/cs276a> for the UI of the search engine.

Location Expansion: The input from the user was then fed to the location expander which queried the map ontology for the parent details of the specified location. If this resulted in multiple parents due to

duplication of location names, we just expanded based on the most important location (refer Section 3) and the rest were displayed as alternatives for the user to choose from if needed.

Meta-Search: The combination of user query and the expanded location terms was then fed to the 3 news sources & a fixed number (10-30 for smaller base set case & 40-120 for larger base set case: see Section 6) of top results were retrieved from each. For instance, for the query “Earthquake”, Location “Stanford”, we queried each news source for “Earthquake Stanford”, “Earthquake California” and “Earthquake USA” and got top 10 results for each in the smaller base set case. We chose to restrict the size of retrieved set per source for speed issues as retrieving articles was the most time intensive job (see Fig 7.4). Also, since the results are already ranked in some fashion by sources, we believe a study of the top set represents the behavior of the complete set very nearly too.

Indexing and Ranking: The retrieved results were indexed using Lucene. The corresponding URLs were followed to get the article bodies for indexing. Lucene was then queried for the user query with the expanded location and our ranking parameters (as discussed in Section 5).

Result Display: The ranked list of results as returned from Lucene was then displayed to the user. Each result displayed shows the Title, Summary, Date and the URL for the corresponding news article. Links to alternative location expansions are also displayed in case of ambiguous location names.

6. Experiments and User Study

We set up 4 test cases to analyze the working of our system and its quality. All queries were finally passed through Lucene in all 4 cases (even when we queried just a single news source) to have the same platform for comparison. The term “Small base set” means we retrieved top 10-30 results corresponding to the query from each news source. “Large base set” means we got 40-120 top results from each source.

Case 1: Here we did no location expansion, just a meta-search on all news sources with the same query as entered by the user, using a small base set.

Case 2: This case did location expansion with meta-search on all 3 sources on a small base set. (This is our proposed system and we evaluate it against all others)

Case 3: Here we did location expansion but no meta-search. All queries were directed to UsaToday only. Again, a small base set was used.

Case 4: This was a meta-search case with location expansion (case 2) but with a large base set this time.

To get data from the users for the various cases, we added a drop-down menu in our search interface for the 4 cases. Each query was run on all 4 cases. Also, to get relevancy judgment each result when displayed had a checkbox associated with it which the user could click when he found it relevant.

The users’ relevancy judgment data was collected in log files each time a separate query/case was issued which were then analyzed and results plotted as presented in the following section.

We collected data for around 170 queries from around 15 users. Most of the users were Stanford students though not related to this course. We also had to choose between giving a certain domain of query to users and inducing some bias or allowing them the freedom to enter any news query. We asked users to issue queries in the news domain which had some geographic aspect.

7. Results and Analysis

7.1 Results set size comparisons:

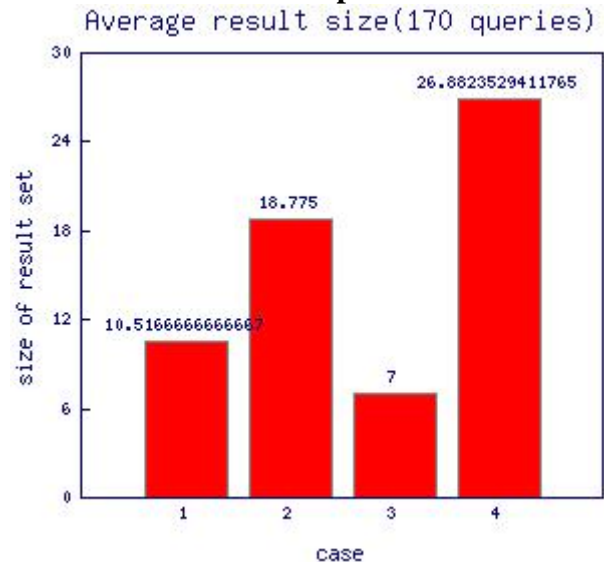


Fig 7.1.Histogram of result set size for all 4 cases.

Fig 1.shows the result set size comparison for the 4 test cases. As expected, case 4 shows the maximum size which is obvious for larger base set. But, it is interesting to note that even though the base set was

almost 4 times in case 4 than in case 2, the result set is not even double in size. This just shows increasing the data set size may increase the result size only to a certain extent.

Case 3 which used only 1 news source instead of 3, understandably shows almost 1/3 the size of case 2 which used all 3 sources.

Between case 1 and case 2, it is easy to see that case 2 has a larger result set as it fetches more articles due to location expansion.

7.2 Precision Comparisons:

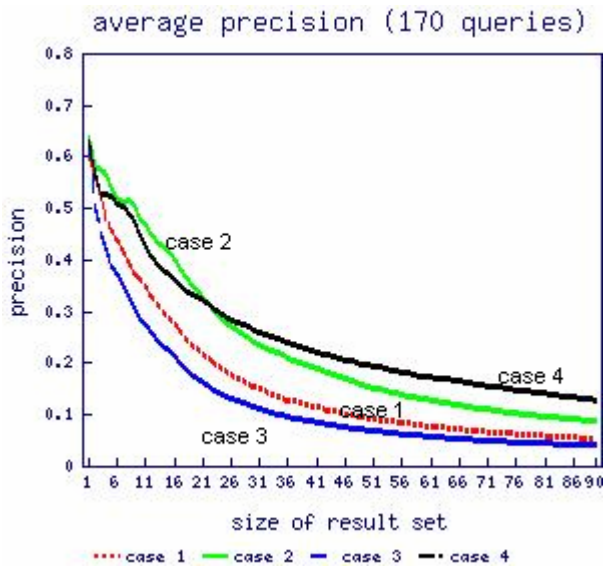


Fig 7.2 Precision versus result set size for all cases

Fig 7.2 plots the average precision against the result set size for each of the 4 cases. For calculating average precision values, we extrapolated the precision curves to get values at points beyond the actual result set size for smaller result set queries. For instance, if a particular query resulted in 5 results of which 3 were relevant, its precision for a result set size of say 10 will be 3/10.

As we can see, the results are pretty encouraging. Our proposed system, case 2 shows the best performance up to a result set size of around 22. This clearly shows the improvement in relevance due to the location expansion when compared to case 1 where all other parameters are the same but there is no location expansion. In fact, for any result set size, case 2 always dominates case 1 on an average basis.

Case 3 shows the least average precision throughout justifying our decision for using meta-search instead of a single source search. As we can observe, case 1 performs better than case 3 throughout even though it has no location expansion. This emphasizes the importance of using multiple news sources when trying to take the real advantage of geographic expansion.

Case 4 (larger base set) outperforms case 2 for result set sizes exceeding 22 and this is not really unexpected. Due to a larger base set of retrieved results, there is a bigger search space to extract good articles from. But, since each news search already ranks the results according to its own relevance criteria, most of the relevant articles are present in the top set of results. In fact the initial better performance of case 2 shows that increasing the base set actually increased the noise in the retrieved set too. The size of the base set can be tuned according to the user need. If user is only interested in the top few results, increasing the base set size might not always be a good choice.

7.3 Sample Graph for a good case

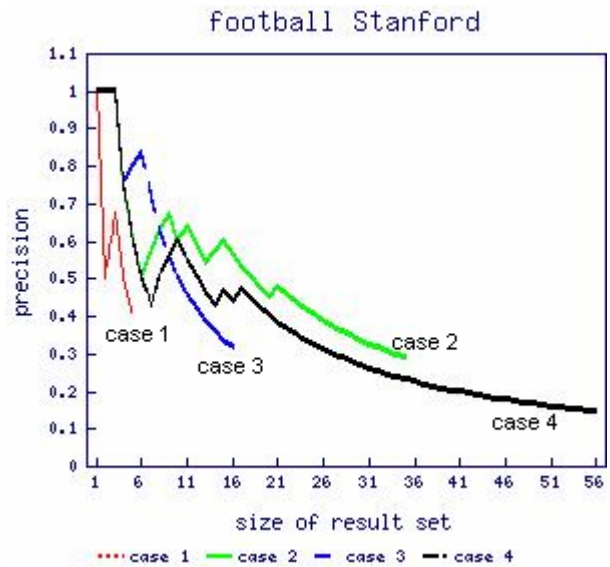


Fig 7.3 Precision curve for Query = football, city = Stanford.

Fig 7.3 is one of the cases which clearly shows that location expansion can lead to much improved performance in certain cases. Here, since the query just included the city name, we could use the full

benefit of expansion by expanding Stanford to California and USA. Also, since 2 of our news sources are US based, they offered a rich collection of results. In fact that is why we see case 3 (UsaToday only) outperforms the other case for a short while. After a small number of results, case 2 stabilizes to the best precision. Note that case 1 is the worst performer for all result sizes.

7.4 Timing Graph

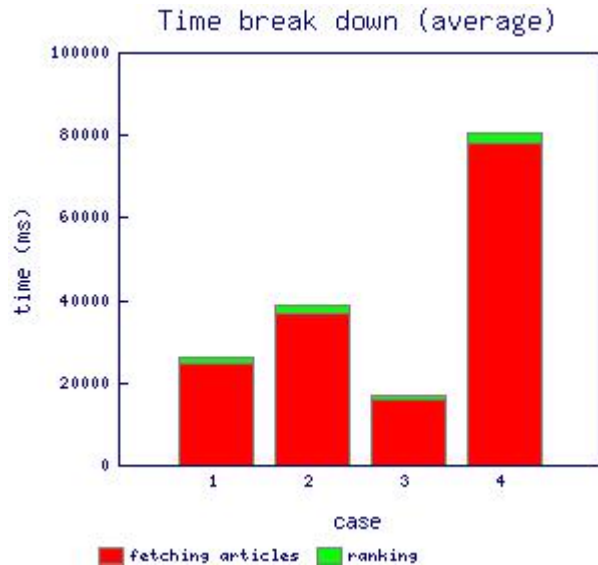


Fig 7.4 Time distribution for various steps in query processing

Our performance criteria in this project was relevance but the slow speed of our query processing made us look into the time distribution as well to figure out the bottleneck. As seen in fig 7.4, over 95% time is spent in fetching articles over the network during meta-search. The time for ranking is relatively negligible and the time for other tasks like location expansion, result display was not even significant enough to be visible on this graph scale. News domain as such is inherently computationally expensive due to frequent update requirements but various techniques like caching, parallel crawling etc. can be applied to reduce time costs.

8. Extensions

This is a low-scale, simplified implementation of an idea to show that geographical expansion makes sense when applied to news and indeed the results are pretty promising. There are many ways in which this

could be extended. Also, there is a lot of scope for improvement.

For instance, Ray R. Larson [1] proposes a coordinate system for geographical information retrieval instead of using names because of the ambiguity linked with location name. So, it is possible to treat a query location in terms of its coordinates rather than as a name. Expansion based on coordinates can be much more accurate, as then we can easily identify the vicinity of a location and closeness can be based on distance based on co-ordinates. But, translating user location to a required coordinate may be an issue.

Another idea, as discussed by Harith Alani et al. [12] is considering different map ontologies and semantic relationships besides hierarchical expansion. This includes member-of, overlap, associated with, in vicinity of, sharing boundary type of relationships which can be taken as different ways to define a closeness measure between geographic locations.

An issue to be explored is the granularity of location expansion; if the expansion is done at a very fine granularity say towns being expanded to streets and nearby towns, it may result in too many local results where the user might be interested in a broader perspective. Similarly, a coarse granularity, like used in our study (city to state to country) may add too much noise to the expansion where the user might be interested in a specific localized news-item. This is a trade-off which should be evaluated and thought about as per user needs.

Another area to be investigated is personalization of query results. Based on the user location, the expansion can be varied to improve relevancy of the news articles. GIS based systems or DNS decoding can help trace the current user location to improve search. Also, a history of user preferred areas/regions when looking for news can help break ambiguity when faced with multiple locations of same name.

It is also a good idea to explore the number and type of news sources to be incorporate din the meta-search to increase our resource. It may be nice to have many regional sources which give more in-depth coverage of various geographic regions.

Some other ideas involve improving the result display. Instead of just a ranked list, it may be a good idea to display a map with dots representing the distribution of articles from different locations/region

on the map. The number of articles from a particular location can be depicted by the radius of the dot etc.

9. CONCLUSIONS

Our test study was on a relatively small scale but it does show a good start in the right direction. In essence, the experiments do support our main belief that even a simple geographic expansion on news meta-search can lead to improved relevance performance.

We believe more research in this area can result in a lot of improvement in the domain of news search. And, as discussed in introduction (Section 1), the idea of geographic information retrieval is not entirely new and there already exists a plethora of well explored techniques to apply to this domain and work on.

10. ACKNOWLEDGMENTS

We would like to thank our CS276A teachers, Prof. Chris Manning, Prof. Hinrich Schütze and Prof. Prabhakar Raghavan, for all the valuable input and feedback during this project. Our thanks to the CS276A TA, Taher Haveliwala, for his time and ideas.

We would like to extend our gratitude to the Mapplanet development team for being very responsive to our queries and letting us use their database.

Last but definitely not the least, we would like to thank all the users who tried out our search utility and gave us valuable data and feedback. Their input made it possible for us to evaluate our system.

11. REFERENCES

[1] Larson, R. (1995). Geographic Information Retrieval and Spatial Browsing. School of Library and Information Studies, University of

California, Berkeley.

http://sherlock.berkeley.edu/geo_ir/PART1.html

- [2] McCurley, K.S. *Geospatial mapping and navigation on the web*. in WWW10, 2001 <http://www10.org/cdrom/papers/278/>
- [3] Orkut Buyukkokten, Junghoo Cho, Hector Garcia-Molina, Luis Gravano, Narayanan Shivakumar, "Exploiting geographical location information of web pages." In Proceedings of Workshop on Web Databases (WebDB'99) held in conjunction with ACM SIGMOD'99, June 1999.
- [4] BBC online news. <http://news.bbc.co.uk>
- [5] CNN online news. <http://www.cnn.com>
- [6] USATODAY online news. <http://www.usatoday.com>
- [7] Getty Thesaurus of Geographic Names. http://shiva.pub.getty.edu/tgn_browser
- [8] National Imagery and Mapping Agency GeoNet names server. <http://164.214.2.59/gns/html/>
- [9] Search engine for important places across the globe. <http://www.mapplanet.com>
- [10] Jakarta Lucene Text search engine in Java. <http://jakarta.apache.org/lucene/docs/index.html>
- [11] Porter, M.F., 1980, An algorithm for suffix stripping, Program, 14(3):130-137. <http://www.tartarus.org/~martin/PorterStemmer/index.html>
- [12] Alani, Harith and Jones, Christopher and Tudhope, Douglas (2000) Ontology-Driven Geographical Information Retrieval. <http://www.giscience.org/GIScience2000/papers/147-Alani.pdf>