World Wide Web (2009) 12:149–170
DOI 10.1007/s11280-008-0052-2

Using Local Popularity of Web Resources for Geo-Ranking of Search Engine Results

Saeid Asadi · Xiaofang Zhou · Guowei Yang

Received: 10 March 2008 / Revised: 28 July 2008 /
Accepted: 5 August 2008 / Published online: 4 September 2008

Abstract Search engines retrieve and rank Web pages which are not only relevant to a query but also important or popular for the users. This popularity has been studied by analysis of the links between Web resources. Link-based page ranking models such as PageRank and HITS assign a global weight to each page regardless of its location. This popularity measurement has shown successful on general search engines. However unlike general search engines, location-based search engines should retrieve and rank higher the pages which are more popular locally. The best results for a location-based query are those which are not only relevant to the topic but also popular with or cited by local users. Current ranking models are often less effective for these queries since they are unable to estimate the local popularity. We offer a model for calculating the local popularity of Web resources using back link locations. Our model automatically assigns correct locations to the links and content and uses them to calculate new geo-rank scores for each page. The experiments show more accurate geo-ranking of search engine results when this model is used for processing location-based queries.

Keywords location-based Web search · geo-ranking · Web graph · link analysis · geo-tagging

S. Asadi · X. Zhou (✉)
School of ITEE, The University of Queensland, GP78 South, St. Lucia, Brisbane, Australia
e-mail: zxf@itee.uq.edu.au

S. Asadi
e-mail: asadi@itee.uq.edu.au, saeid.asadi@uq.edu.au

G. Yang
Hohai University, Nanjing, China
e-mail: hhu.guowei.yang@gmail.com

1 Introduction

Search engines are the dominant tools for finding information on the World Wide Web. In October 2007, 31 billion searches have been performed only on Google alone, more than 1 billion queries each day [18]. This shows the undoubtable importance of the search engines for locating resources on the Web.

Early search engines employed the techniques and rules of traditional information retrieval for searching and ranking the results. They retrieved Web pages relevant to a query. *Relevancy* of Web page w to a query q is measured by the degree of similarity between the terms in w and terms in q. This is a straightforward measurement for ranking in most of the information retrieval systems. However, it fails on the World Wide Web because of the lack of scalability and accuracy. The gigantic size of the Web and the heterogeneous multi-media resources on it, often lead to poor quality results if only relevancy is considered for ranking. More sophisticated tools and techniques have been later employed by search engines.

Hypertext links are considered as a rich source for improving the quality of search. They can reflect the popularity or importance of Web pages. Figure 1 illustrates the link structure of the Web. The World Wide Web can be regarded as a directed graph in which Web pages are the *nodes* and hyperlinks between them are directed *edges* [15]. This graph has been studied and used for finding communities on the Web and also for improving search engine result ranking. A detailed list of Web link analysis algorithms can be found in [7].

PageRank [9] and HITS [14] are two notable link analysis algorithms for finding popular Web pages in a domain. These algorithms calculate a rank score for each page based on its hypertext links. The score is useful to distinguish important or popular pages in a domain and rank them higher than non-popular ones. Both models start with an initial set of Web pages which have been given a rank score manually. Then they try to follow links and automatically assign rank scores to other pages

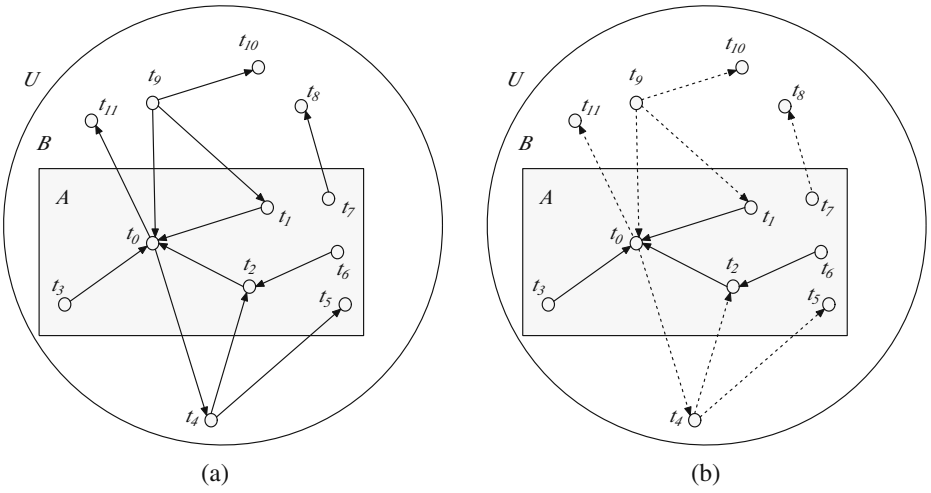

Figure 1 The link structures of the local Web graph vs. the global graph: **a** considering global nodes and links; and **b** considering only local nodes and links.

in the collection. In PageRank, the collection is virtually the entire Web while in HITS it is a smaller subset of the Web built for a specific query. PageRank and HITS are global rank scores i.e. they indicate the popularity of a Web page among all of the Web pages in a domain on the Web. For example, there are thousands of Web pages *relevant* to the query *Java tutorial*. The mentioned algorithms can find the most *popular* pages in the whole relevant set and give them a priority for retrieval and ranking.

Link-based ranking algorithms have been implemented in search engines since late 1990s. Google employed PageRank which resulted in a powerful search engine [9]. While link-based ranking algorithms show useful in general Web search, they are not successful in handling location-based queries. Different studies such as [23] and [19] address the low quality of general search engines on handling geographical queries. Beside the lack of geographic information associated with the Web resources, another issue is that in general search engines, the rank score for each page is calculated *globally* while in location-based search, the Web pages must be analyzed and evaluated *locally*.

Relevancy is the basic criterium for information retrieval in search engines in which documents with more frequent terms similar to query terms are considered more relevant and ranked higher. However, plain frequency-based weighting can lead to inaccurate ranking [25]. Unlike general queries, the best results for a location-based query are the Web resources which are not only relevant to the query topic but also related to a specific location. For example, for the query *Sydney pubs*, the best results are those ones that are *topically* relevant to the query topic i.e. *pubs* and also *geographically* relevant to Sydney. General search engines consider a location name as a keyword or search term and treat them similarly to other keywords [2]. The search engines first detect all resources in their repository which contain at least one of the query keywords and then select and rank *top-k* pages with the highest relevancy and popularity scores [1]. This model often results in poor quality results for location-based queries. For example, there might be some Web pages talking about pubs in New York which are globally famous and more recommended (cited) than the similar pages for Sydney. This conflict leads to poor results for location-based queries.

We think that in geographic Web search, the quality of a Web page is better to be measured by its *local popularity* rather than its *global popularity*. In other words, the best results for a location-based query are those with more local back links rather than ones with geographically widely-spread back links. Similar to PageRank, the quality of the citing pages is considered here as well. The heuristics for this assumption is that local people get more chance to visit local services in the real world and as a result they can better judge the quality of each service. This judgement then will be reflected in the hyperlinks among the Web resources. The more local back links found on a Web page the more *locally* popular the page is among the local people. A geographic search engine can use this local popularity for a more accurate ranking of the results. Back to our example, the best results for the query *Sydney*

[1]The final ranking score is often more complicated and includes many parameters which varies in different search engines and confidential for the search engine. For details on rank aggregation on the World Wide Web see [5].

pubs are the pages with more back links from Sydney rather than from Brisbane or Australia.

In this paper, we introduce a model that considers the local popularity of Web resources to re-rank search engine results for a location-based query. Our model builds up a local graph for each query in a similar way to HITS algorithm and then calculates a new geo-rank score for each page in the graph based on the locations associated with the back links. The major contribution of this work is automatic association of locations to hyperlinks and quantifying the locations by precise power and spread scores which reflect the local popularity of the pages. Unlike HITS and PageRank, our model does not need an initial rank score for the root data set. As a result, the ranking scores are produced without human interfere. We also combine the locations from the content to the locations from the back links and add this as a geo-footnote to the Web pages in a structured XML format. The experiments show that our model offers an accurate result ranking model in which locally-popular Web pages are ranked higher.

The rest of this paper stands as following: In Section 2 we review the previous relevant work on Web links analysis and also on location-based Web search. In Section 3, we will talk about the hyper link structure of the Web and how this structure can improve geo-ranking of search engine results. In Section 4, the structure, tasks and algorithms used in our work are presented. Detailed evaluation methods and results are discussed in Section 5.

2 Related work

2.1 Hyperlinks and the Web graph

The hyperlink structure of the World Wide Web is one of its most significant characteristics. This structure has been studied for different purposes such as finding communities on the Web and improving search engine results. PageRank [9] and HITS [14] are two major link-based algorithms which calculate a content-independent score for Web pages.

PageRank score is calculated based on the number of back links on a Web page as well as the quality of the citing pages. The algorithm starts with a small set of Web pages scored manually with an equal score. Then it follows the hyperlinks between Web pages and calculates a score for each new page based on the number of the back links to that page as well as the quality of the citing pages. The more back links found on a Web page coming from authoritative resources, the more popular the page is. It is assumed that popular or important pages in a domain receive more citation than non-popular.

Unlike PageRank, HITS algorithm is a query-dependent algorithm that assigns an authority score a_w and a hub score h_w to each page w. Similar to PageRank, authorities are the most popular or recommended Web pages with many back links especially from the hubs. Hubs are also high quality Web pages in a domain that point to authoritative pages. A root set of Web pages is manually given the same score similar to PageRank. However, this root set is built based on the top results for a specific query. The hub and authority scores are calculated by HITS are valid only for similar queries and in the same domain while Web pages often have multi

domain contents. As a result, the hub and authority scores can not represent the general popularity of Web pages.

HITS and PageRank have become a platform for many other link-based algorithms. SALSA [17] is similar to HITS with a random walk selection of citing and cited pages. TrustRank algorithm [13] is an extension to PageRank aiming to reduce the side effects of spam and unwanted Web pages on the quality of PageRank. A detailed comparison of some link-based ranking algorithms can be found in [8]. In [12], a statistical approach is used for HITS model to find high quality pages in the base set. HITS and PageRank have also been implemented in a coarser granularity i.e. ranking of the Web site popularity [6].

The above-mentioned models all offer a global link-based score for Web resources. These models are useful for improving the ranking of the search engine results. The good reputation of Google can prove that link-based ranking models such as PageRank are successful on general Web search. However, as mentioned in the introduction, the global rank calculated by these algorithms are not useful in geographic search and they often result in a poor ranking for location-based queries. Our model instead, tries to calculate a location-aware rank score for each Web page based on the geographic information associated with the links.

2.2 Location-based Web search

Recently, location-based search has received attention from both academic and commercial groups.Yahoo Local [2] and Google Maps [3] are two prototypes of geographic search engines with a map and keyword based interface for inputting a location-based query. These search engines are not only geographically limited to few countries, they also search commercial databases such as Yellow Pages instead of the Web pages. The problem originates from the ambiguous dynamic nature of location names, various addressing styles, lack of geographic information, and multiple locations related to a Web resource.

A location-based search engine must be able to find related addresses and location names and assign them to Web pages. This geographic information can then be used to rank search engine results effectively for a location-based query. The location(s) of a Web resource can be defined in many ways: The *server location* of a Web resource can be defined as the location of the server computer that holds the Web resource [16]. This definition of location can be straightforward and available for all Web resources. However, the server location is often totally irrelevant to the intended location of a Web page. The Web site about a coffee shop in Sydney can be hosted on a server in New York for a cheaper or better service.

Content location refers to addresses, location names, and location references in page content and is often a reliable and useful way of defining location of a page. Telephones, postcodes, and latitude and longitudes can be references in a page indicating the page location. Gazetteer-based information extraction is the main approach for extraction of address and location names. As an example, Web-a-Where [1] uses this model to extract all content locations and calculate a geographic focus

[2] http://local.yahoo.com/.

[3] http://maps.google.com/.

for each page. Geographic focus is a dominant location that a page talks about it as a whole. Other studies such as [19–22, 24] used gazetteers for tagging pages with location names. However, content is not always enough for detecting the location as many Web pages do not have any addresses or location names. Also non-textual resources such as photos can not be tagged with content locations. Other sources such as links and log files have been studied for assigning new locations to Web pages.

Ding et al. [11] used the location information associated with back links to calculate a *geographical scope* for a Web page. For example, if most back links on a Web resource *w* are from Web pages related to Sydney, then the geographical scope of *w* can be Sydney. This method is useful for geo-tagging the resources without location references and also for non-textual Web resources. However, most of pages on the Web have few back links or no back links at all. Only pages with many back links have been regarded for the experiments [11]. *Target location* or the location of the visitors of a Web resource [3] has the same characteristics and advantages as geographical scope and can be applied to any Web resources which have some visitors even if they do not have any back links. However, this method also requires access to the Web site log files that are often inaccessible to search engines.

Geo-ranking is another major task of location-based search engines in which search results are ranked not only based on the degree of relevance to a topic but also based on their relevance to a reference location. A distance-based ranking model has been described in [23] that uses geometry and geographic coordinates of Web resources to rank search results based on their distance from the query reference point. This model is straightforward but it requires manually assigning a latitude and longitude to each page. Some studies such as [19] and [10] only consider the domain location as the geography of a Web resource. For example, all Web resources with *.au* in their URL can be considered geographically related to Australia. However, there are many Web sites related to Australia without *.au* in their URL.

3 Link-based geo-ranking of Web pages

As mentioned earlier, current link-based Web page ranking algorithms calculate a general rank which reflects the global popularity of the page. Our goal is to offer a ranking model which reflects the local popularity of Web resources. Similar to PageRank, we calculate a rank score for each document only by using back links. Back links can show the popularity of the resources among the whole community. The detail of our system is described in Section 4. Here we describe the theory of our geo-ranking model.

In this section, we review page scoring formulas and modify them in a way to reflect the location of Web resources. The modified scoring models i.e. PageRank, Power and Spread formulas can result in a geographic page ranking. Our experiments show improvement in location-based ranking of search engine results when location-based ranking models are applied in search engine results (See Section 5).

3.1 Geographic pageRank

Brin and Page [9] introduced PageRank algorithm to calculate rank scores for Web resources based on the links among them. PageRank is a citation measurement

formula which not only counts the number of citations (links) to a Web resource, but also gives different weight to each link based on the importance of the citing page.

Back to Figure 1, let U refer to the whole collection of the Web and $t_0, t_1, t_2, ..., t_n$ be the nodes (Web pages) in this graph. The PageRank formula can offer a global rank score for each node in the graph. As an example, the PageRank score for the Web page t_0 is calculated as shown in the following formula:

$$GPR(t_0) = (1-d) + d\left(\frac{GPR(t_1)}{C(t_1)} + ... + \frac{GPR(t_n)}{C(t_n)}\right) \quad (1)$$

where $GPR(t_0)$ is the Global PageRank (GPR) of t_0; $t_1, t_2, t_3, ..., t_n$ are the pages with direct link to t_0; $C(t_i)$ is the total number of outlinks on page t_i; and finally d is a damping factor. The damping factor is a score between 0 and 1 which is manually determined to avoid unrealistic PageRank scores. For more details see [9].

We modify GPR in a way to consider the location of Web resources. Let $A \subseteq U$ be a subset of Web pages related to a specific location ℓ. We can assume that A in the Web is equivalent to ℓ in the real world. For example, one can claim that all Web pages with Sydney in their contents are geographically related to Sydney. Our intention is to calculate a location-based rank score for each page. We refer to this score as Local PageRank (LPR).

For a Web resource t_0 the LPR is a geographically modified version of the GPR formula or $LPR(t_0, A) = \{GPR(t_0, A) | A \subseteq U\}$ which can reflect the importance or popularity of the Web page t_0 within a specific location A. Web pages with a high global popularity do not necessarily have the same popularity in a specific location. As a result, the GPR formula can lead to less effective results if a location-based query is searched. Figure 1 compares the link structure of the Web when it is considered as a global graph and when it is limited to a desired geographical location. For each Web page in A, the LPR score can be calculated in different ways by using GPR scores or by building a new graph for A.

Using GPR: A naive way for calculation of LPR is using the GPR scores and revising them according to the new graph made for the pages in A. In this model, we use the GPR scores as initial LPR scores. We remove links from and to the pages in B while $B \subseteq U$ and $A \cap B = \phi$. Then we run the PageRank formula to recalculate the scores. The LPR of the page t_0 for the location A is calculated as following:

$$LPR(t_0, A) = (1-d) + d\left(\frac{GPR(t_1)}{C(t_1)} + ... + \frac{GPR(t_n)}{C(t_n)}\right) \quad (2)$$

while $t_0, t_1, t_2, ..., t_n \in A$. This model relies on the GPR. As a result, the score can not be reliable as a measure of local importance. This is because of the fact that the initial rank scores are globally calculated and do not reflect the local popularity of the Web resources.

Using Local Rank: To avoid the possible negative impact of GPR initial scores on the final rank scores, we ignore initial GPR scores and calculate LPR without referring to the GPR. In this case, a new Web graph is built for A containing all pages and links which are limited to A. All nodes in B and all link from or to the nodes in B will be eliminated (Figure 1b). In other words, the U Web graph will be

shrunk to its subgraph A. Now we can run the PageRank algorithm on the small graph of A. The LPR $LPR(t_0, A)$ can be calculated as following:

$$LPR(t_0, A) = (1 - d) + d\left(\frac{LPR(t_1, A)}{C(t_1, A)} + ... + \frac{LPR(t_n, A)}{C(t_n, A)}\right) \tag{3}$$

while $t_0, t_1, t_2, ..., t_n \in A$. Literally, this score can reflect better the local popularity of Web resources. Because only the Web pages related to the location ℓ are considered here, it is expected that more accurate rank scores be calculated.

Hybrid Rank: This method combines the GPR and LPR. GPR is already assigned to documents and LPR score can be calculated after a query is sent. The Hybrid PageRank $HPR(t_0, A)$ is calculated as following to consider both global and local rank equally:

$$HPR(t_0, A) = \frac{GPR(t_0) + LPR(t_0, A)}{2} \tag{4}$$

The hybrid rank can make a balance between the local importance and the global popularity of Web resources.

A basic difference between the original GPR and the LPR is their dependency to a query. The PageRank formula is known to be a query-independent ranking while the LPR and HPR models proposed above are query-dependent scores. Web resources can refer to many locations. As a result, LPR can be calculated literally for any location in the world. This is not useful and efficient for search engines.

3.2 Geographic power and spread

According to the shortcomings of LPR in reflection of the local importance, we use geographic power and spread measurements. Power and spread do not need initial scoring and are easier for calculation. They count the number of back links to Web pages.

Power: Power refers to the number of desired citations or links to a document compared to the total number of citations to it. For example, power can show the popularity of a Web resource in a Web community compared to the entire Web communities. Ding, Gravano and Shivakumar [11] used power as a measure to measure the popularity of Web pages in a geographic area. The power of a Web resource t_0 in a location A,is a fraction of the number of Web pages in A which cite t_0 compared to the total number of the pages in A:

$$Power(t_0, A) = \frac{Links(t_0, A)}{Pages(A)} \tag{5}$$

where $Links(t_0, A)$ is the number of pages (nodes) in the location A with a link to t_0; and $Pages(A)$ is the total number of Web pages in A. Figure 1b shows which back links are used to calculate the power of t_0 in A.

The above-mentioned formula requires counting the total number of the pages in A. This is not an effective measure for geo-ranking as there is not a logical

relationship between the number of back links on a Web page to the total number pages in a geographic area. We modify the power formula in a way to give more emphasis to geography of the back links. Assume that a link is associated with one or more locations. If we analyze the links in a collection, we can obtain some geographic scores for the pages in the collection. The power formula must reflect the power of a location A for a page t_0, compared to all locations related to t_0. Therefore, the power can be calculated as following:

$$Power(A, t_0) = \frac{BackLinks(A, t_0)}{BackLinks(U, t_0)} \tag{6}$$

According to the formula, the power of the location A for the Web page t_0 is the number of back links on t_0 from A compared to the total number of back links on t_0. As an example, if the there are 10 backlinks on t_0 and 5 of them are from Australia, then $Power(Australia, t_0) = 0.5$.

Spread: The power measurement offers only a plain score based on the number of back links. It does not consider where these locations come from. It can be assumed that a Web page has higher quality if its back links are distributed smoothly in sub-locations of A. In the previous example, the popularity of t_0 in Australia can be higher if the 5 citing pages are from 5 different cities within Australia. If they are all from Sydney, the page is probably more popular in Sydney not in entire Australia.

Ding et al. [11] offered three definitions for spread. Assume that location A has $(a_1; ...; a_n)$ child locations. Each sub-location a_i might have some pages with links to t_0. The vector $\overrightarrow{Pages} = (p_1; ...; p_n)$ is a set of pages in a_i and the vector $\overrightarrow{Links} = (l_1; ...; l_n)$ is a set of links l_n to A from the pages in a_i. A vector-space definition of spread calculates the similarity between Pages vector and Links vector i.e. $\overrightarrow{Pages} \odot \overrightarrow{Links}$ by computing the cosine of the angle between two vectors:

$$Spread_{(t_0, A)} = \frac{\sum p_i \times l_i}{\sqrt{\sum p_i^2} \cdot \sqrt{\sum l_i^2}} \tag{7}$$

The cosine of the angle between pages and links vectors will be close to 1 if the links are smoothly distributed in children of A. This means a higher spread score for t_0 in A. For more details on spread formula see [11].

The geographic power and spread formulas mentioned above have been used extensively in our system and experiments. We have developed three geo-ranking algorithms which basically use power and spread scores to re-rank search engine results for location-based queries (See Sections 4 and 5 for more details). The LPR has not been used in our system because our system is a query-dependent model. The PageRank is a pre-query algorithm and the PageRank score is query-independent. Similar to HITS algorithm, our system is query-dependent. We use this post-query analysis model because of the practical issues in collecting a large dataset similar the Web. It can be assumed that a search engine can use both LPR and the power and spread formulas offline. Practically, it would be inefficient to calculate LPR for all possible locations.

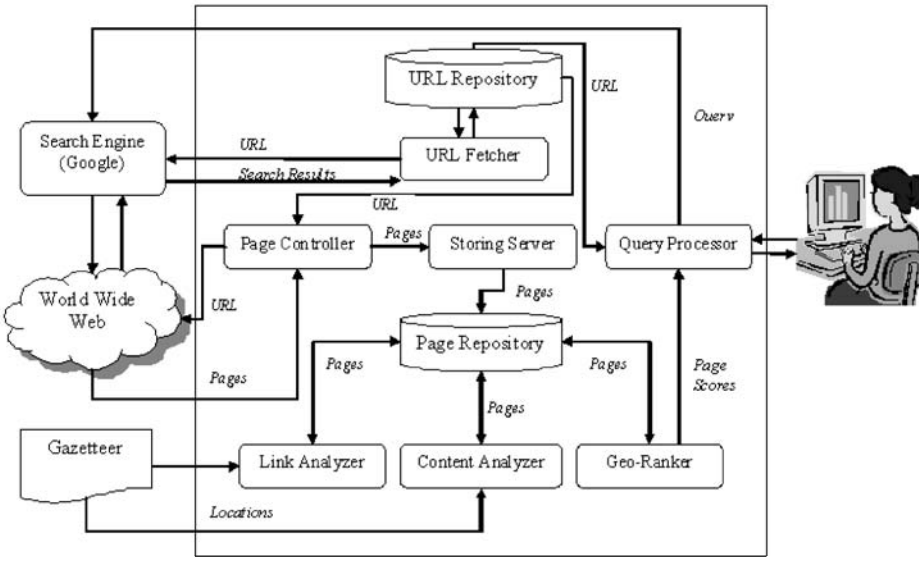

Figure 2 An overview of the system described in this paper.

4 The system architecture

Figure 2 shows the overview of our system. In general, the system first searches a location-based query on a search engine and retrieves the *top-k* results. The back links on the results are used for gathering more URLs and pages. A graph is made for all pages in the collected set. The system then finds content and back link locations and calculate power and spread scores for each location. Finally, three different algorithms are used to calculate geo-ranking scores for all pages. Here, we first talk about the major data structures in our system and then describe different tasks performed by the system.

4.1 Data structures

There are two major data structures in our system. The first one is a *gazetteer* which is the foundation for geo-tagging and link analysis. The second is *geographical footnote* which provides useful geographical information for the geo-ranking algorithms.

The Gazetteer: A gazetteer-based approach has been used in our system for extraction of addresses from the Web page content as well as geographic analysis of the links. We have used Postcodes Australia [4] as a source for obtaining location information. This database has over 100,000 entries of Australian cities, suburbs, localities etc. We built up a smaller gazetteer for our experiments in XML format using selected locations and entries from Postcode Australia. The structure of the gazetteer is given in Figure 3.

[4] Available at: http://www.postcodes-australia.com/.

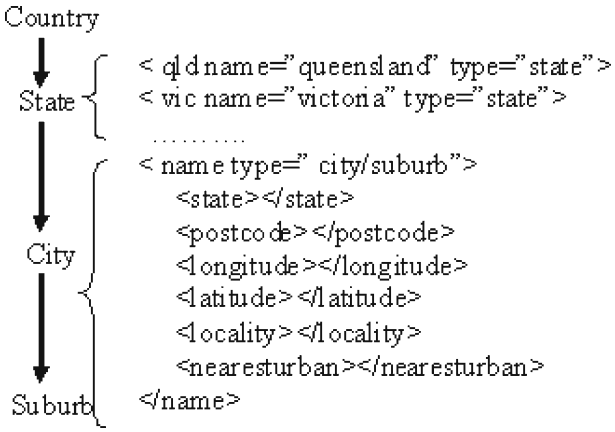

Figure 3 Hierarchical XML structure of the gazetteer.

Geo-footnotes: A geo-footnote is a structured piece of geographic information added to a Web page. This has been refereed to as *geographic footprint* in [19]. In [10] the geographic footprint of Web pages has been compared to the geographic footprint of queries for finding geographically relevant documents. In our work, geo-footnotes are XML-based geographic meta data including all content and back link locations. Figure 4 shows an example of geo-footnote. A geo-footnote consists of four parts: general information, geographical information, link information, and scores information. *General information* contains doc_id and page URL. *Geographical*

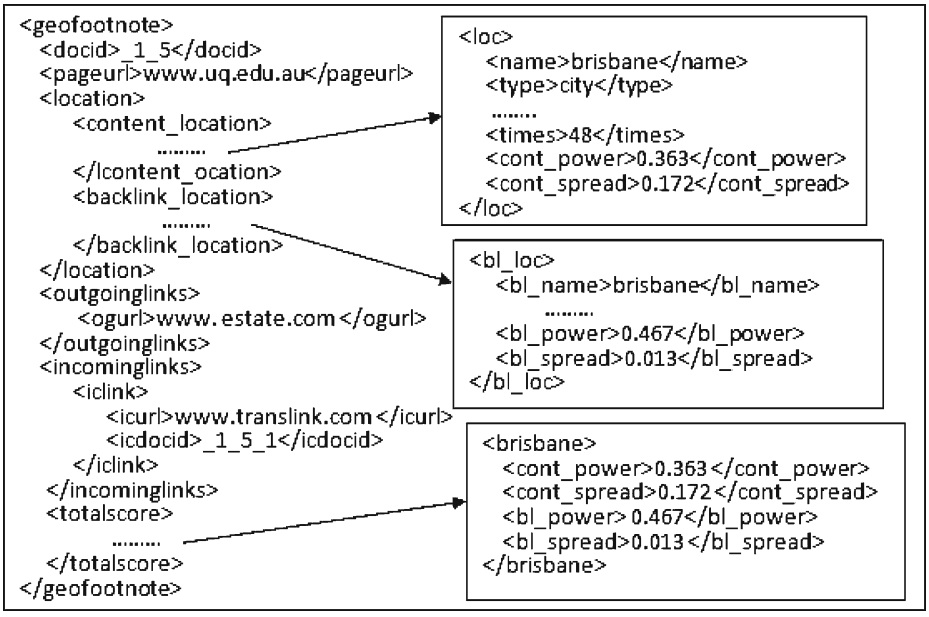

Figure 4 Data structure in the geo-footnotes.

information comprises the content locations and the back link locations with their details and scores. *Link information* contains outgoing and incoming links. And finally, the calculated geo-rank scores are stored in the *scores information* section.

4.2 System tasks

In this section, we give an in-depth explanation of how different major tasks are handled in our system. The system has four major tasks: querying, content geo-tagging, link analysis, and geo-ranking.

Querying: The Query Processor receives a query from a user and sends this query to a search engine. We have used Google search engines in our experiments. A query includes a *topic* and a reference *location*. The *top-k* search engine results are received by URL Fetcher and considered as the *root set* for making a graph. URL fetcher extracts the URLs in the root set and sends them to the search engine again to get the list of back links to each *top-k* page. This URL extraction and search is repeated until a desired number of URLs are achieved. In our experiments, we have collected more than 1200 pages in average for each query (See Table 1). This number is often enough to make a local Web graph.

All of the URLs from the root set and their direct or indirect back links are saved in the URL Repository in XML files. Then Page Collector reads these XML files and downloads corresponding Web pages. The downloaded Web pages get processed on the fly by Storing Server and each receives a doc_id. A geo-footnote is added then to the end of the page's HTML file and additional information including doc_id, PageRank and URL are added to this footnote. The Web pages then will be stored in the Page Repository.

Content-Based Geo-Tagging: Assigning web pages with proper locations extracted from their contents is a requirement for any geographic search engine.

Table 1 20 location-based queries used for the experiments.

No	Query	Pages	DBL	TBL
1	Coffee shop Brisbane	1160	1.20	10.80
2	Coffee shop Toowong	895	0.97	6.37
3	Coffee shop Queensland	1436	1.32	8.92
4	Coffee shop Australia	1709	1.16	11.09
5	Coffee shop Sydney	1386	1.23	11.10
6	Gold Coast Cheap hotels	1380	1.60	8.95
7	Indian food, Gold Coast	990	1.57	8.43
8	Flight to Brisbane	1022	1.94	9.44
9	Brisbane Sydney Flight	1007	1.90	8.33
10	Sydney pubs	1061	1.87	8.35
11	Toowong backpackers	712	0.81	7.95
12	Brisbane backpackers	1169	0.98	8.36
13	Queensland backpackers	1754	1.21	10.30
14	Sydney backpackers	1040	1.33	10.58
15	Australia backpackers	1466	1.05	9.36
	Average	1212.5	1.34	9.22

The process of content geo-tagging starts by finding all possible location names in every Web page's content using a gazetteer-based information extraction approach. The focus of this paper is not on the content geo-tagging. As a result, we simply limit this task to a gazetteer look up. The location, its gazetteer-derived information and its weight in the page are added to the geo-footnote. The content geo-tagging algorithm is as follows:

Content-Based Geo-Tagging Algorithm.
Input: w - a Web page in the Page Repository
Output: w' - Content-based geo-tagged w

```
        WordSet = ProcessSet = null;
1.Open w
2.  TotalWeight = 0;
3.  WordSet = ExtractAllWordFromPage(w);
5.  addGeneralInformation(w); //URL and doc-ID
7.  for each term t in WordSet
8.    if(!ExistInProcessSet(t) && ExistInGazetteer(t))
9.      ℓ = t; add ℓ to footnote
10.       addWeight(Tf/Itf(ℓ)); ProcessSet.add(ℓ);
12.       TotalWeight(ℓ) = TotalWeight + Tf/Itf(ℓ);
13.   else continue
15. for each location ℓ_f in footnote
16.                          SetContPower(Weight(ℓ_f)/∑
TotalWeight(ℓ_1...ℓ_n));
17.   for each location ℓ_p in ProcessSet
17.   if(ℓ_p.equals(ℓ_f))
18.     continue;
19.   else
20.     if(ℓ_p.IsSubLocationOf(l_f))
20.     Spread(ℓ_f)=Spread(ℓ_f)+log(Weight(ℓ_f,w)/TotalWeight);
20.   SetContSpread(-Spread(ℓ_f,w)/
      Log(NumberOfSubLoc(l_f)));
20.  addLinks(w);//incoming and outgoing links
20.  ProcessSet.clear();
20. Close w; rename w to w'
```

The algorithm analyzes every page in the Page Repository. First, all words in a page are extracted and compared with gazetteer. If a word has not been processed before and it exists in the gazetteer, it will be added to the geo-footnote as a location. Then *content power* and *content spread* of each location will be calculated and added into footnote.

Link-Based Geo-Tagging: In the process of link analysis, we try to gather geographical information from the citing Web pages. More precisely, the locations found in a citing page will be added to the footnote of the cited pages if their power and spread exceeds a threshold. This part is done after content geo-tagging and after few iterations it will stop when all geo-footnotes in the data set are being tagged with back link locations. *Back link power* and *back link spread* are two significant criteria

in this step. Later, these parameters will be used to calculate a back link geo-rank score. The algorithm for back link geo-tagging is as follows:

Back Link Geo-Tagging Algorithm.
Input: *w'* - Content-based geo-tagged Web page
Output: *w"* - Geo-tagged page with weighted back links

```
            ProcessLoc = null;
1.Open w'
2.  BackLinkList = getBackLinkList(p);
3.  //add backlink-power
5.  for each backlink bl in BackLinkList
7.    LocList = getLocationList(bl);
8.    for each loc l in LocList
9.    if(ExistInProcessLoc(l))
10.       getBackLinkFootnote(p, l).setBacklinkPower(
            getBackLinkPower(p, l) + getContPower(bl,
l));
13.    else
15.       addBackLinkFootnote(p, l);
16.       setBackLinkPower(getContPower(bl, l));
17.       ProcessLoc.add(l);
17.  CalculateSpreadForBackLinkloc(p);
18.  ProcessLoc.clear();
20. Close w; rename w' to w"
```

In this algorithm, the locations in the geo-footnote of the citing pages to A are analyzed and then transferred to geo-footnote of A. The back link power and spread for each location in A is a normalized sum of the location's content power and back link scores in all of the citing pages.

For both content locations and back link locations, the power and spread scores are calculated using the power formula 3.2 and spread formula 3.2.

Geo-Ranking Algorithms: After tagging pages with geographical information, three different algorithms are used to evaluate each location's popularity in a page and give each location a geo-rank: CGR or Content-based Geo-Ranking; BGR or Back Link based Geo-Ranking; and HGR or Hybrid Geo-Ranking.

(a) In CGR, only *content power* and *content spread* are considered to give a score for each location. For a location ℓ in content geo-footnote of Web page w, $CGR_{(\ell,w)}$ will be calculated as:

$$CGR_{(\ell,w)} = ContentPower_{(\ell,w)} + ContentSpread_{(\ell,w)}$$

Term frequency is the basic score for information retrieval tools where the similarity between a query and a document is measured according to the similarity of their terms. Search engines consider this measure by making inverted indexes of the collected Web pages. However, they do not differentiate location names and, as a result, treat them as other terms in a page. In CGR, location names are deliberately extracted from the text and weighted according to their occurrence and distribution compared to other locations in the same Web page. CGR is the bottom line algorithm in this work as it only considers the

traditional content-based extracted locations. BGR algorithm instead, works based on the links and Web graph.

(b) BGR focuses on the locations associated with the back links. Every location in the goe-footnote of the citing pages can be added to the footnote of the cited page. For a location ℓ in the back link geo-footnote of Web page w, $CGR_{(\ell,w)}$ will be calculated as:

$$BGR_{(\ell,w)} = BackLinkPower_{(\ell,w)} + BackLinkSpread_{(\ell,w)}$$

A page can still be considered for a query even if the query's reference point is not directly found in the page. For example, for the query "*coffee shops in Brisbane*", BGR can find a Web page without *Brisbane* explicitly mentioned in the content if Brisbane has been found in the back links to this page.

(c) HGR is the last algorithm which combines the geography of the content with the geography of the back links. HGR can be used after all relevant content and back link locations are extracted from the Web pages. It adds up the power and spread scores of each location in the footnote regardless of its source.
For a location ℓ in the geo-footnote of Web page w, $HGR_{(\ell,w)}$ will be calculated as:

$$BGR_{(\ell,w)} = CGR_{(\ell,w)} + BGR_{(\ell,w)}$$

4.3 Limitations

Location Ambiguities: Location ambiguity is a common problem for geographic indexing search tools. There are many geo/geo ambiguities such as similar city or suburb names. For example, Singapore can refer to a city and to a country. Geo/non-geo ambiguities often refer to the similarity between the name of places and people. Washington, Paris and Adelaide are name of cities. However, there are famous people with similar names to these cities. In our previous work [4], we offered a context-aware approach to extract and disambiguate location names. This model has been used in the current paper to disambiguate location names from non-location names. For geo/geo ambiguities, we have relied on the same method as well as the gazetteer data. For example, if West End is found in a page, then according to the gazetteer data and also the surrounding information in the page it will be decided which entry in the gazetteer is the most relevant entry to this location.

Slow Process: The analysis of links and building geographic footnotes can be time consuming. In our experiments, most of the time has been spent for gathering the pages and analyzing the links. For a search engine, this process can be much faster as they already collect and index the pages.

Insufficient Links: Link analysis algorithm can only be applied to Web pages with inlinks or outlinks. Inlinks (backlinks) are essential for HITS, PageRank, power and spread formulas. We expand the number of links to a page by including indirect back-links. For example, if t_0 is cited by t_1, we include the backlinks to t_1 as indirect links to t_0. For those pages in *top-k* search engine results with no backlinks, we include the backlinks to the parent domain. For example, if www.ourbrisbane.com/Toowong does not have any backlinks, we search for backlinks to www.ourbrisbane.com. In this way, we can increase the number of backlinks for each page (See Table 1).

The experimental results for evaluation of the above-mentioned geo-ranking system is presented in next section.

5 Evaluation

In this section, the settings and results for evaluation of our system are presented. The experiments try to illustrate how accurately our system can estimate the local popularity of Web pages and reflect them in the location-based re-ranking of search engine results.

5.1 Measurements

Our experiments are limited to the accuracy of the geo-ranking and geo-tagging models described in the previous sections. Performance evaluation is out of this paper's scope. Accuracy is often measured by recall and precision. For search engine ranking, recall is not a reliable measure because of the unknown size of the relevant resources on the Web. Precision can be a better measure since the documents can be judged by users. In our work, precision has been described as following:

$$GeoRanking - Precision = \frac{RelevantPages_{(top-k)}}{Pages_{(top-k)}}$$

We use *top-k* search engine results as the root. We also only retrieve and rank top-k results after all pages in the repository are tagged and their scores are calculated. In most of our experiments, *k=100*. Precision is measured for three different criteria: topic, location, and topic and location. A single Web page retrieved in the *top-k* results by any of our geo-ranking algorithms will be checked to see if it is *topically* and *geographically* correct.

Another task of this research is geo-tagging the pages with content and back link locations. This is a post-query process and as a result, it is possible to measure the recall and precision of this task in the collected set. The recall and precision of geo-tagging are measured as following:

$$\begin{aligned} GeoTagging - Recall &= \frac{CorrectLocations\_detected}{CorrectLocations} \\ GeoTagging - Precision &= \frac{CorrectLocations\_detected}{Locations\_detected} \end{aligned} \tag{8}$$

5.2 Experimental setup and data set

To collect the experimental data set, 15 location-based queries were searched on Google and the results were manually analyzed to assign each page with proper locations (extracted from the content) as well as to count and control the hyperlinks. Only *top-100* results were collected and analyzed for each query. The queries were limited to Australia and its sub-locations. Table 1 shows queries 1 to 15. On average, 1212.5 pages were collected for each query. A local graph was made for each query in which collected Web pages were considered *nodes* and links between them *edges*.

Table 2 Distribution of geo-footnotes.

Web Page Geo-Footnote	Distribution
Non-empty content footnotes	79.3%
Non-empty back link footnotes	96.2%
Non-empty footnotes	96.6%

The table shows that Web pages in the root set (*top-100* Google results) often do not have many direct back links (DBR), 1.34 in average for each page. Since our model relies on back links, back links to the main domain were used for each page in the root set if no back link was found on it. After downloading the whole set and making the local Web graph, the average total number of back links (TBL) for each page increases to 9.22. This is enough for many pages to be analyzed for further link-based experiments.

The collected pages were first geo-tagged with the content locations and with back link locations. The power and spread scores were then calculated and assigned to each location in the geo-footnotes. The next step was calculating geo-rank scores for the query reference location and it's related locations. For example, if the reference location was Sydney, then all suburbs of Sydney were considered as well. Three geo-ranking models described in Section 3 i.e. Content-Based Geo-Ranking (CGR), Back Link Geo-Ranking (BGR), and Hybrid Geo-Ranking (HGR) were used to calculate the final ranking scores for the pages. Finally, top-k results with highest geo-ranking scores are selected and ranked separately for CGR, BGR and HGR.

5.3 Results

Table 2 shows that 96.6% of the Web pages in the collected set were successfully tagged with at least one location. Almost all of these pages have at least one location in their back link footnotes. The content footnote was non-empty for less than 80% of the pages. This Shows that back links can be a rich source of location information.

Table 3 Precision of 3 ranking algorithms for the 20 location-based queries.

No	CGR	BGR	HGR
1	0.61	0.73	0.73
2	0.43	0.57	0.60
3	0.58	0.64	0.69
4	0.56	0.76	0.67
5	0.70	0.75	0.54
6	0.64	0.82	0.77
7	0.49	0.71	0.70
8	0.55	0.60	0.57
9	0.51	0.58	0.55
10	0.60	0.81	0.74
11	0.52	0.73	0.61
12	0.58	0.63	0.64
13	0.53	0.71	0.76
14	0.61	0.55	0.57
15	0.48	0.64	0.66
Average	0.56	0.68	0.65

Table 4 Precision of the proposed ranking models compared to Google.

	Topic	Location	Topic & location
Google	0.42	0.76	0.59
CGR	0.41	0.70	0.56
BGR	0.44	0.89	0.68
HGR	0.40	0.89	0.65

Similar to any other ranking systems, *accuracy* is the main measure in our work. We use *precision* as a standard measure to compare different algorithms to each other and also to Google ranking. The precision of our model for ranking *top-100* results for each query is shown in Table 3.

Table 4 summarizes the accuracy of the three ranking models used in our system compared to Google. The results show that almost BGR is slightly more precise in ranking the results based on *topic*. The table also indicates that Google and our models detect the correct *location* of Web pages much more precisely than the *topic*. On average, the precision of topic detection is 0.42 while this is 0.81 for detecting locations. This big gap may result from the fact that search engines consider location

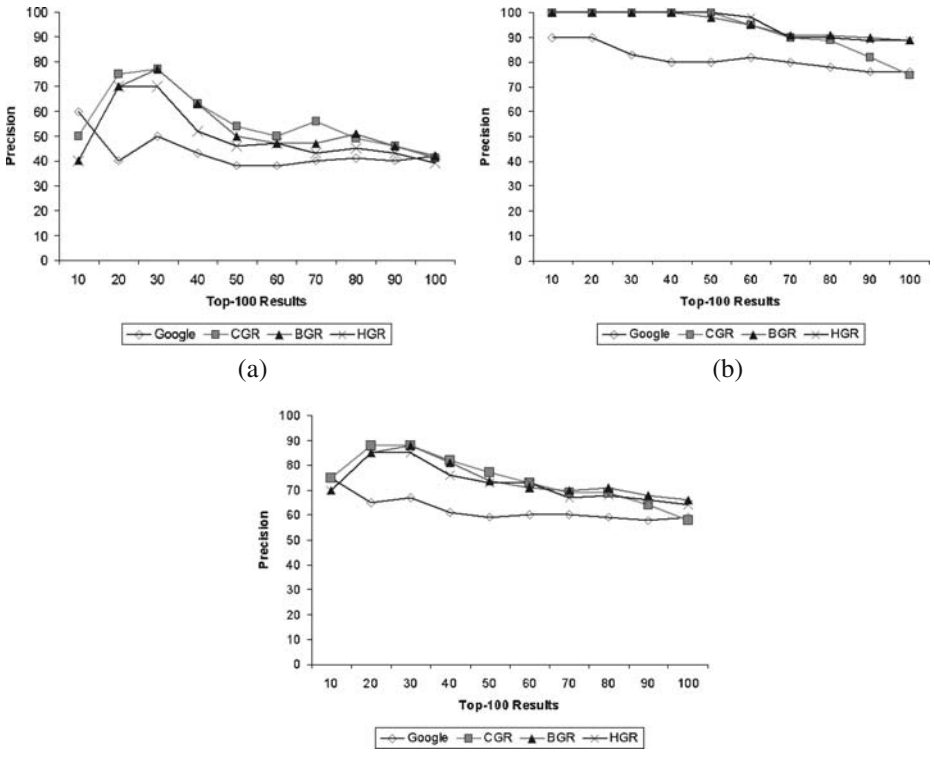

Figure 5 Average precision of CGR, BGR, HGR compared to Google in accurate detection of: **a** Topic; **b** Location; and **c** Topic and Location.

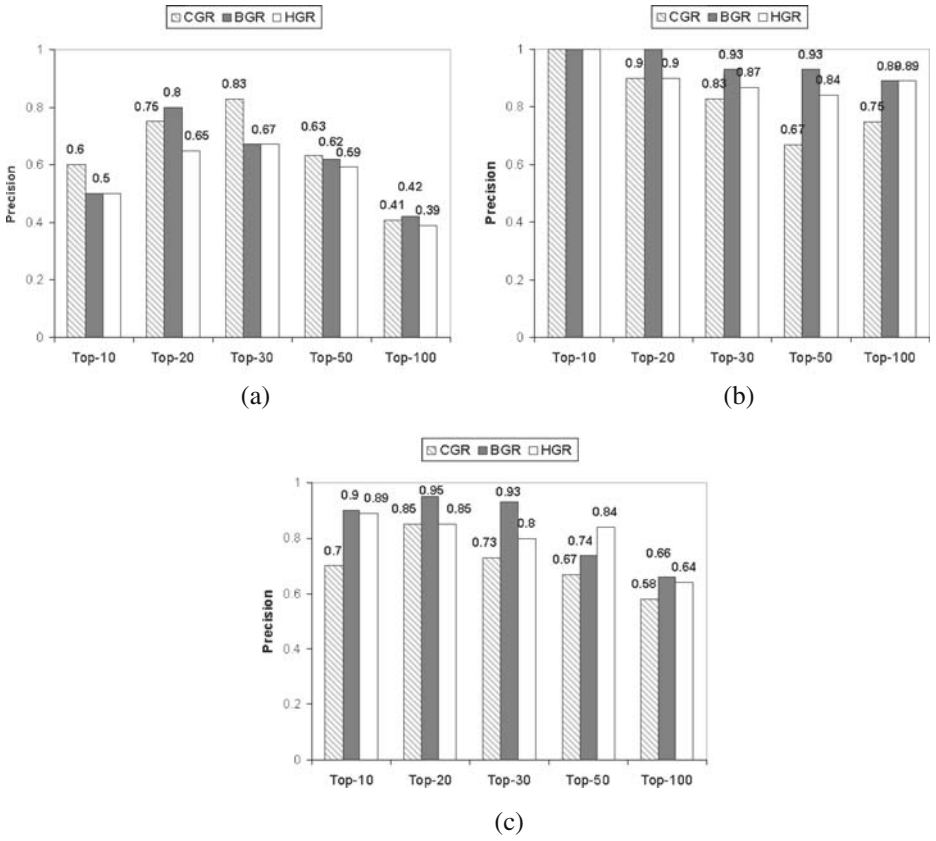

Figure 6 Effect of root set size on final geo-ranking of: **a** Topic; **b** Location; and **c** Topic and Location.

names as keywords. A proper name e.g. *Sydney* in a text can be more deterministic than a general keyword such as *pubs*.

According to Table 4, BGR achieves a higher precision for ranking the results based on both topic and location. The overall precision for BGR is 0.68 while it is 0.59 for Google in our experiments. GCR does not show a high accuracy which means using only the content locations can not result in an accurate geo-ranking. However, when content is accompanied with back links (HGR), the overall precision will be higher.

Figure 5 shows the average precision of the ranking models in different sets of results. As mentioned before, we have setup different *top-k* of results to compare the changes in precision. From the figure it can be concluded that the highest precision can be achieved in *top-30* results if only *topic* is considered. This will be *top-60* if only *location* is considered. Again a comparison of the sub-figures shows higher average precision of our algorithms compared to Google ranking.

In previous experiments, we have used *top-100* results of Google as the root set. Next step is using different sizes of the root set to find the best *top-k* input for

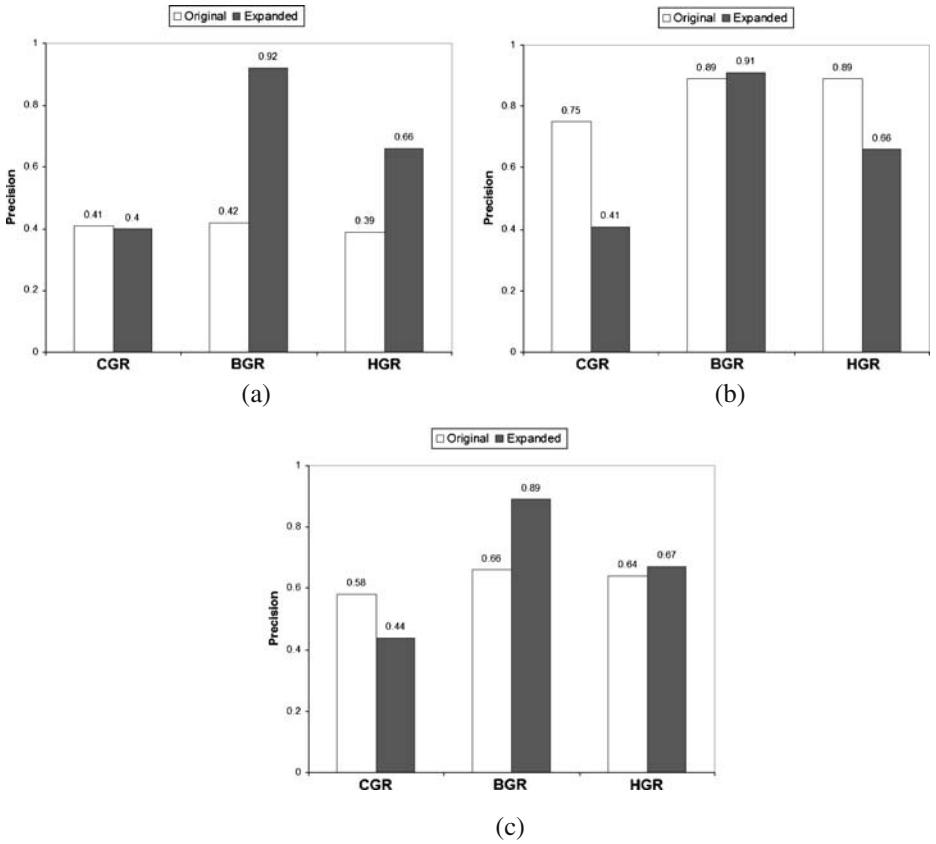

Figure 7 Effect of location expansion on the precision of detecting and ranking Web pages based on: **a** Topic; **b** Location; and **c** Topic and Location.

achieving the best results. We have examined our system with different root sets: *top-10*, *top-20*, *top-30* and *top-50* Google results. Figure 6 shows the effect of the root set size on the final geo-ranking. The best results for accurately detecting and ranking of the Web pages based on their *topic* is acquired when the *top-30* Google results are used as the root set (Figure 6a). For accurate detection and ranking of the *locations*, top-10 root set will result in the highest precision (Figure 6b). Using the top 30 Google results as the root set leads to the best precision in general (Figure 6c).

We have also examined the effect of location expansion on the final results. Normally, a *location* is a set of one or more keywords in a query. Search engines use

Table 5 Accuracy of detecting locations in content and back links.

Level	Recall	Precision
Country	0.99	0.92
State	0.95	0.88
City	0.84	0.79
Suburb	0.78	0.68
Total	0.89	0.82

an arbitrary logic to retrieve the pages which contain the location name according to the indexed content terms. We know that location often can have an implicit definition. As a result, considering all possibilities of equivalent places for a reference location can result in better retrieval and ranking. This is a query expansion in which the reference location of a query is expanded according to the knowledge from a gazetteer. As an example, if the query is *"Brisbane coffee shops"* then a Web page talking about coffee shops in Toowong will also be included in the results since Toowong is a suburb in Brisbane.

Figure 7 compares the precision of geo-ranking when the reference location is expanded. According to the figure, expansion of the reference location has a negative effect on ranking precision if only content locations are used (CGR). However, this expansion will improve the accuracy of ranking if back links are included (BGR). Combining the geographic information from the content and back links (HGR) will result in a slightly more accurate overall geo-ranking (Figure 7c).

While our focus in this paper is on geo-ranking, the accuracy of location extraction from page content is measured. According to Table 5, our system extracts countries and states with a high recall and precision. The precision reduces for cities and suburbs because of synonymy, more locations and more ambiguities. Altogether, the recall and precision for location extraction is approximately 90% and 80% respectively in our system.

6 Conclusion

Location-based search is becoming more popular as more local services and facilities are becoming available through the World Wide Web. Unlike general search which looks for globally more important Web pages, geographic search engines must give a higher rank to those pages which are locally more important. In this paper, we offered some modifications to PageRank, power and spread formulas and made use of HITS algorithm to make a local graph for search engine results and calculate new geo-scores for the Web pages. The system builds a geo-footnote for each page and assigns content and back link locations to the footnote. The experiments show that back links can reflect the local popularity of Web pages well. Precise geo-ranking is acquired when back link locations are considered. Our geo-ranking algorithms show more accurate topically and geographic ranking of search engine results. A future work would be efficient query processing for faster calculation of the geo-scores and presentation of the geo-ranked results for a location-based query. It is also considered to compare the effectiveness of the local popularity measures described in this paper with the traditional global popularity measures such as PageRank and HITS.

References

1. Amitay, E., Har'El, N., Sivan, R., Soffer, A.: Web-a-where: geotagging web content. In: SIGIR, pp. 273–280 (2004)
2. Asadi, S., Chang, C.-Y., Zhou, X., Diederich, J.: Searching the world wide web for local services and facilities: a review on the patterns of location-based queries. In: WAIM, pp. 91–101 (2005)

3. Asadi, S., Xu, J., Shi, Y., Diederich, J., Zhou, X.: Calculation of target locations for web resources. In: WISE, pp. 277–288 (2006)
4. Asadi, S., Yang, G., Zhou, X., Shi, Y., Zhai, B., Jiang, W.W.-R.: Pattern-based extraction of addresses from web page content. In: APWeb, pp. 407–418 (2008)
5. Beg, M.M.S., Ahmad, N.: Soft computing techniques for rank aggregation on the world wide web. World Wide Web **6**(1), 5–22 (2004)
6. Borges, J., Levene, M.: Ranking pages by topology and popularity within web sites. World Wide Web **9**(3), 301–316 (2006)
7. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Finding authorities and hubs from link structures on the world wide web. In: Proceedings of the 10th international conference on World Wide Web, pp. 415–429. ACM, New York (2001)
8. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Link analysis ranking: algorithms, theory, and experiments. ACM Trans. Inter. Tech. **5**(1), 231–297 (2005)
9. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)
10. Chen, Y.-Y., Suel, T., Markowetz, A.: Efficient query processing in geographic web search engines. In: SIGMOD Conference, pp. 277–288, Chicago, 26–29 June 2006
11. Ding, J., Gravano, L., Shivakumar, N.: Computing geographical scopes of web resources. In: VLDB, pp. 545–556 (2000)
12. Greco, G., Greco, S., Zumpano, E.: A probabilistic approach for distillation and ranking of web pages. World Wide Web **4**(3), 189–207 (2001)
13. Gyngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: VLDB, pp. 576–587. (2004)
14. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J ACM **46** (1999)
15. Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: The web as a graph: measurements, models, and methods. In: COCOON, pp. 1–17 (1999)
16. Lakhina, A., Byers, J.W., Crovella, M., Matta, I.: On the geographic location of Internet resources. IEEE J. Sel. Areas Commun. **21**(6), 934–948 (2003)
17. Lempel, R., Moran, S.: The stochastic approach for link-structure analysis (salsa) and the tkc effect. Int. J. Comput. Telecommun. Netw. **33**, 387–401 (2000)
18. Lipsman, A.: 61 billion searches conducted worldwide in august. In: ComScore: Measuring the Digital World, October 10 2007. http://www.comscore.com/press/release.asp?press=1802
19. Markowetz, A., Chen, Y.-Y., Suel, T., Long, X., Seeger, B.: Design and implementation of a geographic search engine. In: WebDB, pp. 19–24 (2005)
20. Ourioupina, O.: Extracting geographical knowledge from the internet. In: Proceedings of the ICDM-AM International Workshop on Active Mining, Maebashi, December 2002
21. Uryupina, O.: Semi-supervised learning of geographical gazetteers from the Internet. In: Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references, pp. 18–25. Association for Computational Linguistics, Morristown (2003)
22. Wang, C., Xie, X., Wang, L., Lu, Y., Ma, W.-Y.: Detecting geographic locations from web resources. In GIR '05: Proceedings of the 2005 workshop on Geographic information retrieval, pp. 17–24. ACM, New York (2005)
23. Watters, C., Amoudi, G.: Geosearcher: location-based ranking of search engine results. J. Am. Soc. Inf. Sci. Technol. **54**(2), 140–151 (2003)
24. Woodruff, A.G., Plaunt, C.: Gipsy: automated geographic indexing of text documents. J. Am. Soc. Inf. Sci. **45**(9), 645–655 (1994)
25. Zakos, J., Verma, B.: A novel context-based technique for web information retrieval. World Wide Web **9**(4), 485–503 (2006)