Detecting Geographic Locations from Web Resources

Chuang Wang[1]*, Xing Xie†, Lee Wang‡, Yansheng Lu*, Wei-Ying Ma†
†Microsoft Research Asia, 5F, Sigma Center, No. 49, Zhichun Road, Beijing, 100080, P.R China
{xingx, wyma}@microsoft.com
‡Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA
leew@microsoft.com
*Department of Computer Science, Huazhong University of Science & Technology, Wuhan, 430074, P.R. China
{chwang, ysl}@mail.hust.edu.cn

ABSTRACT
The rapid pervasion of the web into users' daily lives has put much importance on capturing location-specific information on the web, due to the fact that most human activities occur locally around where a user is located. This is especially true in the increasingly popular mobile and local search environments. Thus, how to correctly and effectively detect geographic locations from web resources has become a key challenge to location-based web applications. In our previous work, we proposed to explicitly distinguish three types of locations for web resources, namely provider location, content location and serving location. Provider location is the physical location of the provider who owns the web resource; content location is the geographic location described in the web content; while serving location is the geographic scope that a web resource can reach. In this paper, we present a system that comprehensively employs a set of algorithms and different geographic sources by extracting geographic information from the web content, and mining hyperlink structures as well as user logs. As the result, only relevant geographic sources, rather than all of possible ones are used in computation of each category of web location. Finally, experimental results on large samples of web data show that our solution outperforms previous approaches.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process, retrieval models, information filtering*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*

General Terms
Algorithms, Experimentation, Performance

Keywords
Location-based web application, web location, provider location, content location, serving location, dominant location

[1]This work was performed when the first author was a visiting student at Microsoft Research Asia.

GIR'05, November 4, 2005, Bremen, Germany.

1. INTRODUCTION
Intuitively, most web resources including web page and site, have geographic features [1][3][16][17][19]. For instance, a web page with information about or within a special geographic scope, such as listings on houses for sale in a given region, could be regarded as a local page with a certain location. With the rapid pervasion of the web into users' daily lives, it becomes more important to capture location-specific information from the web to cater for users' local information needs. For example, location-based web applications are emerging to provide more customized and tailored services according to users' location, such as web geographic information navigation and retrieval, location-based web search, local advertisements, and context-aware services [10][11].

The common principle of most applications is to detect the geographic attribute from web resources, and then match it with current user's location to provide more tailored services. As we know, users' location can be easily acquired, for example, by their IP address or location-aware devices. Therefore, how to effectively and precisely deduce the web locations, with taking full advantage of relevant geographic sources, becomes the key challenge to these location-based web applications.

Due to the increasing importance of geographic features for web resources, much work has been carried out to improve the accuracy of web location detection and estimation. However, none of them has exploited the fact that various applications require diverse categories of geographic location and the detection of each location needs only relevant geographic sources. According to our experiences, we found that various categories of location may coexist in the same web resource. For instance, a user should know the headquarters of MSN site [23] if he/she wants to visit the MSN team; advertisers put more interest in the serving or influencing scope of MSN; while users from New York tend to find the NY local pages [21] on MSN. As a result, the ignorance of the location differences will cause a mismatch between location deduction and application needs, and ultimately result in undesirable or unreliable results.

In our previous work [26], we explicitly distinguish the locations of web resources into three categories: provider location, content location and serving location, to cater for different application needs. To correctly and effectively detect each type of web locations, in this paper, we propose a system that comprehensively employs a set of algorithms and different geographic sources by extracting geographic information from the web content, and mining hyperlink structures as well as user logs.

As the result, only relevant geographic sources, rather than all of possible ones are used in computation of each category of web location. Finally, experimental results on large samples of web data show that our solution outperforms previous approaches.

The main contribution of this paper lies in: instead of employing a general-purpose algorithm to acquire geographic information from web resources, we develop a set of algorithms to compute the different categories of web locations based on their specific characteristics, including the proposed top-down dominant location detection algorithm.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 introduces geographic location categories for web resources. The algorithms for computing the three categories of web location are presented in Section 4, and the experimental results are given in Section 5. Finally, we conclude this paper and describe our future work in Section 6.

2. RELATED WORK

Due to the increasing importance of geographic features, a number of studies have been carried out to detect geographic locations from web resources. Generally, there are three major identifiable research directions: 1) exploiting various geographic information sources, 2) identifying and disambiguating place names, and 3) developing effective computation approaches.

Previous work has exploited a wide range of geographic sources, such as IP address, Whois database, route packet and DNS can be used to deduce the host location from computer network aspect, various extracted geographic references from web content, including place name, postal code, telephone number, language, people/organization name, geographic metadata, and additionally, hyperlink and user access log [3].

Much work has focused on identifying geographic references and disambiguating place names. These are commonly referred to as geoparsing and geocoding respectively. The latter is also usually called grounding [14][15]. First, a gazetteer including various exploited geographic information sources must be prepared in advance to recognize and extract geographic references from web content, and then a footprint must be assigned to each identified geographic reference. The process exist two types of ambiguities: geo-nongeo and geo-geo [1][18]. For instance, "Washington" can mean a person name as well as place name, even the reference is determined to denote a place name, we still needs to confirm whether it means a state or a city. Some Natural Language Processing (NLP) techniques and context information in web content can be used to help distill correct senses of recognized geographic references [14][15].

Ding et al. contain the maximal similarity with our work. They proposed the CGS/EGS algorithm [5] based on geographic content and context sources. In this approach, the authors first defined two key measures, namely power for measuring interest and spread for measuring uniformity, and then pointed out that the geographic scope of a web resource must satisfy two conditions: smooth distribution (CGS, the candidate geographic scope) and then significant interest (EGS, the estimated geographic scope), that is, enough spread and power. Finally, content-based and link-based techniques are proposed to estimate the geographic scope. Although it is time-consuming, their work is highly indicative for our approach. A similar geographic-focus algorithm [1] proposed by Amitay et al. is more apt for individual web pages that contain very few geographic references.

There exist some research prototype systems such as Columbia GeoSearch [4], Geotags GeoSearch [6] and Kokono Search [29], which have attempted to estimate web location by extracting geographic references from the web content. However, the accuracy of traditional applications based on Directory [28] or Yellow Pages [7][20] are still as good, if not better than, these research systems, as the results of ignoring the intrinsic differences of web location categories.

3. GEOGRAPHIC SOURCES FOR WEB LOCATIONS

3.1 Web Location Categories

In our previous work [26], we categorize web locations into three categories, and each is defined as follows:

- **Provider location:** The physical location of the provider who owns the web resource, such as organization, corporation or person. This kind of location is crucial to web geographic information retrieval and navigation such as online map and Yellow Pages services.
- **Content location:** The geographic location that the content of a web resource describes. As a spatial attribute of a web resource, this type of location can be utilized to better satisfy users' information needs according to user's location. Location-based web search is one of its classical applications.
- **Serving location:** The geographic scope that a web resource can reach. Knowing the serving location of a web resource can benefit many business applications such as local advertisements and e-commerce.

To make these definitions more explicit, we present an example to illustrate these three location categories in Figure 1. In this example, the provider location lies in state of Oklahoma, USA; the content location of page1 (which speaks about the tourism in Nevada) is state of Nevada, USA; and the serving location covers mid-region states of USA.

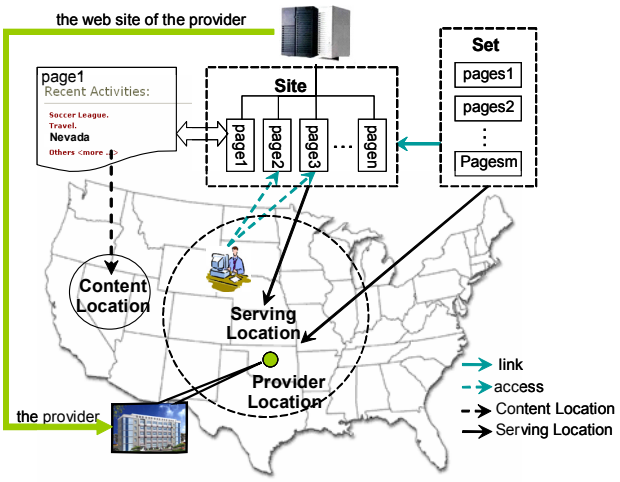

Figure 1. An illustration of provider location, content location and serving location

3.2 Geographic Information Sources

According to our definitions, we set out to exploit the available geographic sources to deduce each category of the location that is defined in our solution.

- **Provider location**. Provider location of some web resources can be easily acquired from the business databases such as Yellow Pages. Unfortunately, this is not feasible for all business since Yellow pages often requires charge fee for registration. Thus, some service providers choose to build web sites to introduce themselves and release their contact information, which is a more feasible and convenient way to announce their provider location to the world. Thus, we can acquire provider locations of non-registered entities based on their web content. Generally, we identify the provider location according to some explicit clues. For example, place name, postal code, and telephone number should appear together, or at least the former two, in an address string.

- **Content location**. Most of existing research on web location detection falls into this category [1][3][9][13][16][17][27]. Just as its name implies, we can acquire content location by analyzing the content of a web resource. Here, according to a gazetteer that's constructed in advance, we identify the geographic references that occur in the content. Further, we need to deal with place name ambiguities. For example, Washington can refer to a geographic place or even a person name. We need to analyze the contextual information of the page to determine whether it indicates geographic features. If multiple locations coexist in the content of a web resource, our approach tries to estimate the dominant geographic location.

- **Serving location**. When user logs of a given web resource are available, we can calculate its serving location by analyzing the users' geographic distribution. However, such logs of various web resources are practically unavailable to us. Thus, we turn to utilize hyperlink structures among web resources as the main clue for estimating the serving location. The serving location of web resources can be transferred along hyperlinks and access links. That is, given a web resource *w*, if the geographic scope of most web resource which has hyperlinks to *w* is location *l*, or most access users located in *l*, then the serving location of *w* is *l*. Our experiment results further support this assumption.

Table 1 summarizes a number of geographic sources that are useful to estimate the three locations.

Table 1. Geographic sources for computing the three locations.

Location category	Geographic sources
Provider location	Yellow Pages, address information databases, and address strings in web content, etc.
Content location	Geographic references in web content such as geographic name, telephone number, postal code, institution or organization names, and geographic meta-data etc.
Serving location	Hyperlinks, and access users' location etc.

4. WEB LOCATION DETECTING

In this section, we set out to introduce a set of algorithms that compute various categories of web location with considering only relevant geographic sources.

4.1 Detection Workflow

Figure 2 shows the workflow of our algorithms for computing our three proposed location categories. We first analyze web pages to extract all geographic references and hyperlink structure in its content body. Besides, user logs are also processed if they are available. Then

- Provider location is detected and learned according to the extracted address strings and some selected features, which is presented in 4.2.
- Content location is computed through integrating various extracted geographic references, which is described in Section 4.3.
- Serving location is estimated based on the content locations, together with the geographic information from inbound hyperlinks and user logs, which is presented in Section 4.4.

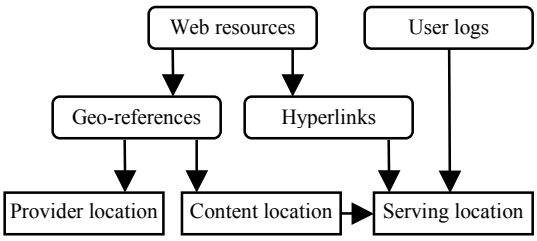

Figure 2. The flowchart of location detection system

In our implementation, we adopted two different granularities for computing geographic locations: page level and site level. Based on the fact that content location is often specific to individual web pages, we compute content location at web page level. However, provider location and serving location are computed at site level.

4.2 Computing Provider Location

The computation of provider location for a web resource includes two steps:

1) Identifying and extracting address strings from web content;
2) Estimating whether the extracted address strings are the candidate provider location.

The first step can be implemented by identifying the sequential address strings in web content. The representative format of contact addresses facilitates this recognition. For example, the USA address format is usually represented as: "street address, city, state, Zip code, country or region". Additionally, address initializations and abbreviations, along with punctuations and separating tags of HTML, are all important clues to improve extraction precision.

For the second step, we exploited some heuristic rules for deducing provider location. For example, the string of provider location often occurs in the footer of a web page. After a comprehensive study, we formulate the factors that determine whether an address string indicates provider location as follows:

- Referred frequency. We found that the provider location often appears in multiple pages of a web site. Thus, more times an address string is referred, the higher the possibility the string is the provider location.

- URL levels. In most sites, the web pages that contain provider locations are often placed in the first or second level directory of these sites.
- Title, anchor and content body. As we know, these features constitute the content information of web resources. According to [12], the title of a web page usually describes the topic of the content, and the anchor text usually summarizes the content. The motivation for selecting these features comes from the evidence that provider location often occurs in contact or home page, etc.
- Spatial position of extracted address strings on the page. We exploited an obvious evidence that a provider location is more often referred in the footer or header of a web page rather than other regions in a page.

Based on these features, we employ Support Vector Machine (SVM), which has been found quite effective for text categorization problems [8], to learn whether an extracted address string in a web page indicates the provider location of the web site owner. In our experiments, we investigated and labeled a number of web sites, and chose them as the training data to acquire a SVM model, which is utilized to extract provider location from the testing data.

4.3 Computing Content Location

To precisely estimate the content location for web resources, we first need to identify all geographic references from web content using a gazetteer and then ground them to specific locations. As a result, for given web document, all extracted geographic references as well as corresponding probabilities that measure the reliability of each reference as geographic location are achieved.

Then, for web resource *w*, the weight of given geographic location *l* in *w* is defined as follows:

$$Weight(w,l) = \sum Georef(w,l) + \sum_{l_i \in offspring(l)} Georef(w,l_i) \quad (1)$$

where Georef(w, l) means the probability of extracted geographic references as l.

In the above variation of weight definition of current location, in addition to including the weight of current location node itself, we also comprehensively consider the weights of all its offspring location nodes in geographic hierarchical tree, whose theoretical basis is that referring an offspring node also means indirectly referring all its ancestor nodes, due to the belong-to relationship between them. In fact, the specific reason lies in our following observation.

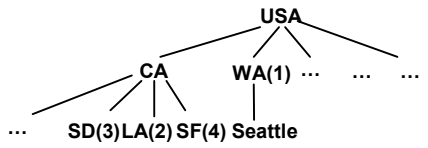

Figure 3. Illustration of power variation.

To avoid unnecessary computation cost, we propose a top-down algorithm to estimate the dominant content location. Obviously, the result should be CA, rather than WA. However, when we traverse the geographic tree in top-down fashion, the output will be WA if the weight of CA doesn't include all that of its offspring nodes, namely 3, 2 and 4 of SD, LA and SF, respectively.

Due to the high reliability of Zip codes and telephone numbers in correctly identifying unique geographic locations, compared with ambiguous place names, found in our experiments, we use the same constant *Georef*(*zt*) (is greater than zero but less than one) to represent their common weight. On the other hand, we know that some place names are ambiguous and deficient to identify a specific location, such as those place names with multiple corresponding location nodes on the geographic hierarchical tree, or those can be used in a non-geographic sense, such as people names. Therefore, in our approach, we assign different weights to each identified place name according to extracted contextual environment in web content.

Next, we borrow two basic definitions, namely power and spread, from CGS/EGS approaches [5]. Further, the power measure is adapted as follows to satisfy our top-down dominant location detecting algorithm by normalizing weight value:

$$Power(w,l) = \frac{Weight(w,l)}{Weight(w,rt)} \quad (2)$$

where *rt* denotes the root node of the geographic hierarchical tree.

Spread is defined as same as that in [5] and its entropy definition is chosen for the best performance based on their results:

$$Spread(w,l) = \frac{-\sum_{i=1}^{n} \frac{Power(w,l_i)}{\sum_{j=1}^{n} Power(w,l_j)} \times \log\left(\frac{Power(w,l_i)}{\sum_{j=1}^{n} Power(w,l_j)}\right)}{\log n} \quad (3)$$

where, l_i or l_j is a direct children node of l ($1 \le i, j \le n$, n is the number of all children of l).

Once multiple location nodes are identified from web resources, after having achieved weight and power values of each node, content location can be computed by traversing the geographic hierarchy in a top-down fashion starting from the root node as follows:

Input: rt, the root of geographic tree with weight and power values of each location node have been obtained
Output: ResultSet, including dominant locations
Parameters: Tw, Tp, Ts, thresholds for weight, power, spread, respectively

```
Algorithm ComputeDominantLocation (rt)
    ResultSet = TempList = null
    If Weight(rt) > Tw and Power(rt) > Tp Then
        If Spread(rt) > Ts Then
            Insert rt into ResultSet
        Else
            Insert rt into TempList
    For each node n in TempList
        For each child c of node n
            If Power(c) > Tp Then
                If Spread(c) > Ts Then
                    Insert c into ResultSet
                Else
                    Insert c into TempList
        Delete n from TempList
    Return ResultSet
```

Our proposed algorithm will traverse the geographic tree starting from the root node if its weight and power value are more than

given thresholds, then visit those directly nodes whose power is enough. For each current visited node, if its spread value, which is computed dynamically, is enough, then insert the node into returned result set; else continue to examine its children.

In the algorithm, we first adapt the power definition to include effects from off-springs on the geographic hierarchy. Furthermore, we compute spread value lazily just when it's necessary. Finally, Due to avoiding visiting the absolutely majority of unrelated location nodes, our top-down dominant location detection algorithm achieves significant improvement in time efficiency. Besides, in addition to place names, we also consider more reliable geographic sources, i.e. postal codes and telephone numbers, in the power calculation. We also introduced weight factors to control the balance between different types of geographic keywords, as well as weighing each place name by their likelihood to be truly about a geographic entity. Our more comprehensive understanding of web locations gives us better accuracy in our experiments.

4.4 Computing Serving Location

Since the algorithm of computing serving location is similar to that of content location, in this section, we will only cover the differences between them.

As shown in Figure 2, content locations, user locations and inbound hyperlinks are the data sources of computing the serving location. Our computation process of detecting serving location is similar to the iteration and convergence process of PageRank algorithm [2]. In our algorithm, the serving location is translated between web resources along hyperlinks, like the importance does in the PageRank algorithm.

For serving location, given a web resource w and location l in the geographic hierarchical tree, the weight of l in w is calculated as follows:

$$Weight(w,l) = \begin{cases} \alpha_1 Userfreq(w,l) + (1-\alpha_1) Contloc(w,l) & i = 0 \\ \alpha_2 \sum_{j=1}^{n} Srvloc_{i-1}(w_j,l) + (1-\alpha_2) Srvloc_{i-1}(w,l) & i > 0 \end{cases} \quad (4)$$

where, $Userfreq(w,l)$ is w's access frequency by all users within location l; $Contloc(w,l)$ equals to 0 or 1, which means whether l is contained in the content location of w; w_j is the web resource that has links to w, ($1 \le j \le n$, n is the number of all the web resources that have links to w); Similar to $Contloc(w,l)$, $Srvloc_{i-1}(w,l)$ denotes whether l is hierarchically contained in the intermediate serving location of w after the $(i-1)^{th}$ iteration; Values a_1 and a_2 are the weight of user access frequency and serving location of previous iteration, respectively.

To start, we first traverse all pages within the given site and calculate the content location for each page. We also collect locations of access users of the site. Different weights are assigned to the two kinds of locations according to Equation 4 when i=0. Then using the same power and spread definitions and top-down dominant location detection algorithm in estimating content location, we can obtain a first-iteration serving location.

The obtained serving location can be further refined based on the previous-iteration itself and serving locations of other sites that have inbound hyperlinks to the site of interest. Multiple iterations are often needed until the computational results converge to steady values on the location tree. The converged values are our final serving location.

In summary, we devised a novel iterative algorithm to compute the serving location. First, we estimate a given site's initial serving location using access users' locations and page content locations across the site. Then we iteratively refine (i.e., increase the accuracy of) this serving location using the location information from inbound links.

5. EXPERIMENTS

5.1 Experimental Settings

5.1.1 Gazetteer

To recognize the geographic references in web content, we need to construct a gazetteer in advance. In our approach, we collect various geographic information sources, including Zip codes [25], telephone numbers [24] and geographic names [22] under USA scope. After analyzing and integrating these geographic data sources, we use four tables to constitute our gazetteer, they are:

- Geographic hierarchical table with standard place names;
- Alias-Geography table;
- Zip-Geography table; and
- Telephone-Geography table.

As can be clearly seen, we use the latter three tables to map various geographic sources into the geographic hierarchical table, which is left for our various location computing algorithms.

In our experiments, the USA geographic hierarchical tree contains three divisional levels, such as country (USA only), state (all 50 states, including Washington DC, and official state-level entities such as Northern Mariana Islands), and city (34,546 cities or towns across the USA). On average, a state-level node has about 455 city nodes.

5.1.2 Web Resources

We use .GOV data as the benchmark dataset used in our experiments, which were mainly crawled in the year 2002 and are extensively used by TREC2003. We believe this data covers a wide geographic range at all levels of the USA geographic tree.

For each URL, we utilize the top three levels in its domain name to distinguish its web sites. For example, although jsc.nasa.gov and jpl.nasa.gov both belong to the same domain nasa.gov, we deem them as two different sites for simplicity. After eliminating the sites that includes no more than 5 pages, we acquire 4,430 sites and 1,053,111 pages to test our algorithms.

Table 2. Distribution of geographic references.

Keywords	Occurrence	Page	Site (4,430)
Zip	919,170	232,344 (22%)	3,143
Telephone	1,139,677	236,516 (22%)	3,191
Place Name	80,652,212	822,219 (78%)	4,116
Zip or Telephone	2,058,847	323,587 (31%)	3,440
Any of the three	82,711,059	835,969 (79%)	4,133

After each page in our testing data is scanned through our gazetteer, we list the distributions of the three geographic sources in Table 2. As shown in the table, the ratio of place name occurrence covers a dominating percentage in all the sites, about 93%. We list several observations as follow:

- Zip codes cover a percentage of 71% in all web sites. As we know, Zip codes usually occur in a postal address string, which are most likely to be the provider location. Thus, it can be estimated that about 71% of web sites will include provider locations.
- In total about 79% of web pages contain at least one of the three kinds of geographic sources, which indicates that content location is probably available in these pages.
- The distribution of Zip code is similar to that of telephone number, about 22% in all the pages and 71% in all the sites. Further, we found that their confidences in estimating locations are close, after comparing the computational results under different weights of them. Therefore, the same *Georef*(*zt*) is assigned for Zip code and telephone number.

Due to the lack of space, we only highlight the key results and observations from our experiments in the following two subsections.

5.2 Parameter Tuning

5.2.1 SVM Kernel Setting

For provider location, we implemented our approach using several SVM variations to select the best algorithm and parameters. We employ various kernel settings in SVM to test the location computing over our testing data. We list the testing results in various kernels in Table 3.

Table 3. Comparison of SVM learning methods.

Methods	Precision	Recall	Micro-F1
SVM-Linear	0.87	0.89	0.88
SVM-Polynomial	0.93	0.88	0.90
SVM-Sigmoid	0.92	0.90	0.91
SVM-Gaussian	0.96	0.92	0.94

As shown in Table 3, the SVM using Gaussian kernel achieved the best performance, which can achieve precision and recall of 96% and 92%, respectively, and Micro-F1 measure of 0.94. The results also demonstrate that nonlinear combination of the features is better than a linear combination. In our later experiments, we adopted Gaussian kernel for computing the category of provider location.

5.2.2 Spread Threshold Setting

Figure 4 shows the impact of *Ts*, i.e. the given threshold for the Spread value, in computing the serving location. *Ts* has similar impacts on content location and on CGS/EGS content-based and link-based geographic scopes. The data demonstrates that the best Micro-F1 measure can be acquired when *Ts* is around 0.75 (0.7 for both techniques of CGS/EGS).

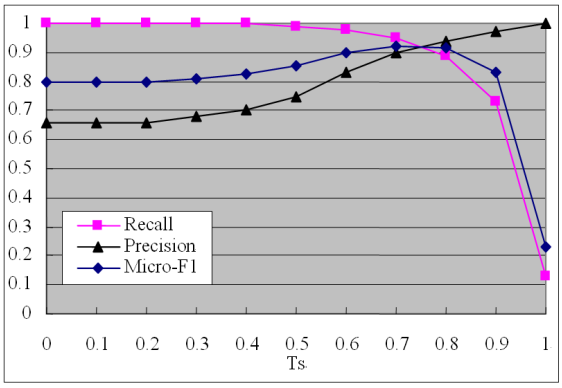

Figure 4. Impact of *Ts* on serving location

From Figure 4, we also found that when *Ts* is larger than 0.8, recall and Micro-F1 measure will drop dramatically, which indicates that an extreme spread threshold will result in a major exclusion of serving locations.

5.2.3 Geo-Reference Weight Setting

We compared our proposed algorithm with the CGS/EGS approach that includes content-based and link-based techniques. Figure 5 shows the impact of *Georef*(*zt*) on the Micro-F1 measure for these algorithms. In practice, to make fair comparisons, we extended the geographic sources to include Zip codes and telephone numbers for the CGS/EGS approach.

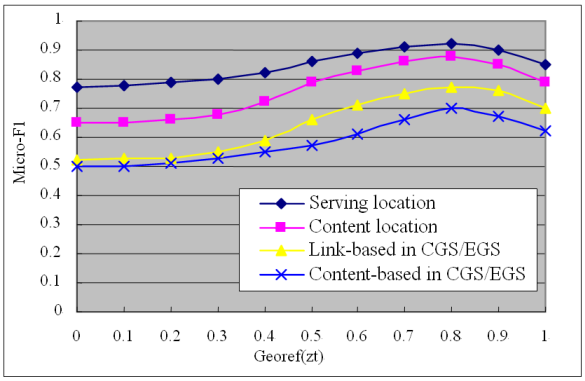

Figure 5. Impact of *Georef*(*zt*) on Micro-F1 measure

When *Georef*(*zt*) is less than 0.3, its variation does not obviously affect the Micro-F1 measure. This is mainly due to the fact that, generally Zip codes and Telephone numbers are rarely available in web resources in comparison with geographic names. As can be readily seen from Figure 5, the best Micro-F1 measure can be acquired when *Georef*(*zt*) is around 0.8 for each algorithm. We also found that there exist lighter impacts of *Georef*(*zt*) on serving location than content location, since serving location depends more heavily on hyperlink structures. We also observe that the impact of *Georef*(*zt*) in our algorithm is similar to CGS/EGS.

5.2.4 The Summarized Parameter Settings

After finishing the tuning of all the parameters that are required in our later computing approach, we summarize them in Table 4. The parameters settings in the table ensure us to achieve the best Micro-F1 measures. Besides, since users' logs in various web sites are unavailable to us, α_2 is set to 0.

The experiments employ Micro-F1 and running time to measure the accuracy and time cost of our algorithms, Finally, our experimental environment is a machine with Intel Xeon CPU 3.06 GHz, 2 GB RAM and running Microsoft Windows Server 2003.

Table 4. Parameters used in our experiments.

Parameters	CGS/EGS		Our algorithm	
	Content based	Link based	Content location	Serving location
Georef(*zt*)	0.80	0.80	0.80	0.80
Tp	0.50	0.50	0.50	0.50
Ts	0.70	0.70	0.75	0.75
α_1	/	/	/	0.85
α_2	/	/	/	0.00

5.3 Experimental Results

In this part, we measure the precision of our solution to compute various categories of web location. Further, we compared our proposed algorithm with the CGS/EGS approaches over two aspects.

5.3.1 Precision

We first randomly selected 1,000 sites out of all the 4,430 sites, and manually labeled their provider locations and serving locations. For each chosen web site, we also randomly selected a web page and labeled its content location. As a result, the serving locations of about 829 sites out of these 1,000 sites were labeled as geographically related.

Table 5. Precision of our algorithms.

Test set	Labeled as local	Precision
Provider location (1,000 sites)	714 (71%)	685 (96%)
Content location (1,000 pages)	537 (54%)	510 (95%)
Serving location (1,000 sites)	829 (83%)	771 (93%)
Pages with different content and serving location		758 (76%)

In Table 5, we present the percentages of the labeling results and the corresponding precisions of our algorithms. As shown in the table, about 71% of the web sites have declared their provider locations, and 96% of them can be precisely estimated by our computing algorithm.

As aforementioned in Section 5.1.2, about 79% of the pages contain at least one of the three geographic sources. However, the labeled data shows that only 54% of these pages are really geographically related. The gap between two ratios is caused by two reasons: 1) some false place names represent non-geographic sense actually, e.g. person names or common senses; and 2) some place names are not significant enough to represent the content location for a web page. Besides, we found that about 83% web sites in our testing dataset contain serving location, and our approach can achieve a precision of about 93%.

In addition, we exploited a significant difference between content location of a web page and the serving location of its corresponding web sites. More detailed, we found that about 76% of them are different, which further proves the high necessity to distinguish content location and serving location.

5.3.2 Comparison with CGS/EGS

In this part, we compare our solution with the CGS/EGS approaches on the same data set. The experimental results are presented in Table 6. Our proposed algorithms can achieve better performance in both Micro-F1 measure and computational cost.

More specifically, the serving location computed by our proposed algorithm outperforms both the content-based and link-based methods in CGS/EGS by 18% and 15% in Micro-F1 measure, respectively. Although both our serving location computing algorithm and link-based CGS/EGS algorithm utilize hyperlinks, our algorithm can achieve 3 times as fast as the CGS/EGS algorithm in the computational cost.

Table 6. Summary of experimental results.

Results	Our algorithm			CGS/EGS	
	Provider location	Content location	Serving location	Content based	Link based
Precision	0.96	0.95	0.93	0.81	0.84
Recall	0.82	0.80	0.91	0.75	0.76
F-measure	0.88	0.87	0.92	0.78	0.80
No. Iterations	1	1	4	1	7
Time(hr) / Iter.	0.6	2.8	3.3	4.9	5.7
Total time (hr)	0.6	2.8	13.2	4.9	39.9

5.4 Discussions

Having analyzed the results, we find that the improvements mainly come from the following three aspects:

- The main contribution to the quality of our algorithm is that we have distinguished the locations of web resources into provider location, content location and serving location rather than mixing them into one location, and each location is computed by only considering its relevant geographic sources and intrinsic characteristics.
- Our proposed algorithms of computing content location and serving location start from the root node and do traversing. Location nodes with abnormal spread and power value caused by the ambiguities of place name can be correctly removed from our final results since the spread value of their parent nodes will be less than the given threshold. This is often the case for some leaf nodes on the geographic tree. Furthermore, the algorithms' running costs are largely reduced due to that we do not need to compute on offspring nodes when the current node is ignored. In contrast, in the CGS/EGS algorithm, all nodes are computed, which increases not only the time cost, but also the possibility of introducing false positives.
- The modified power definition better represents the "weight" of locations on the geographic hierarchy. Obviously, it is more reasonable to increase the weight of current node with total that of its offspring nodes considering the belong-to relationship existing among them.

6. CONCLUSIONS

In this paper, we presented a system that computes three categories of web locations, i.e. provider location, content location and serving location. It employed a set of effective location detection algorithms, including the proposed top-down dominant location detection algorithm, and used only relevant geographic sources for each web location definition to achieve high accuracy and fast speed. To improve the accuracy, we extended existing algorithms by including more reliable geographic sources such as postal codes, telephone numbers, and locations of web users, in addition to place names. Experimental results on a large set of web sites showed that our approach outperformed a generic location detection algorithm that did not distinguish web location categories in both accuracy and speed measures. In the future, we will further test our algorithms on more types of data sets and include locations outside the US.

Currently, we are planning to implement a location-based web search engine based on the geographic locality information that is categorized and computed by our proposed solution. In this system, we first filter and refine the retrieved results, so as to reserve the web pages whose content locations are related to users' location. Further, we display the search results on a geographic map according to the provider locations of returned sites. In addition, advertisements will be tailored to users with serving locations overlapping with the users' locations.

7. REFERENCES

[1] Amitay, E., Har'EI, N., Sivan, R., and Soffer, A. Web-a-where: geotagging web content. 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04), Sheffield, UK, Jul. 2004

[2] Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. 7th International World Wide Web Conference (WWW7), Brisbane, Australia, Apr. 1998

[3] Buyukkokten, O., Cho, J., Garcia-Molina, H., Gravano, L., and Shivakumar, N. Exploiting geographic location information of web pages. ACM SIGMOD Workshop on the Web and Databases 1999 (WebDB'99), Philadelphia, USA, Jun. 1999

[4] Columbia GeoSearch. http://geosearch.cs.columbia.edu

[5] Ding, J., Gravano, L., and Shivakumar N. Computing geographic scopes of web resource. 26th International Conference on Very Large Data Bases (VLDB'00), Cairo, Egypt, Sep. 2000

[6] Geotags GeoSearch. http://geotags.com

[7] Google Local Search. http://www.google.com/local

[8] Hearst, M.A. Trends and controversies: support vector machines. IEEE Intelligent Systems, 13(4), Jul. 1998, 18-28

[9] Hill, L.L., Frew, J., and Zheng, Q. Place names: the implementation of a gazetteer in a georeferenced digital library. Digital Library, 5(1), Jan. 1999

[10] Jones, M., Jain, P., Buchanan, G., Marsden, G. Using a mobile device to vary the pace of search. 5th International Symposium on Human Computer Interaction with Mobile Devices and Services (Mobile HCI'03), Udine, Italy, Sep. 2003

[11] Kaasinen, E. User needs for location-aware mobile services. Personal and Ubiquitous Computing 7(1), May 2003, 70-79

[12] Kan, M.Y. Web page categorization without the web page. 13th International World Wide Web Conference (WWW'04), New York, USA, May 2004

[13] Larson, R.R. Geographic information retrieval and spatial browsing. Smith, L.C. and Gluck M. (Eds), Geographic Information Systems and Libraries: Patrons, Maps, and Spatial Information, University of Illinois, Urbana, IL, USA, 1996, 81-123

[14] Li, H., Srihari, R. K., Niu, C., and Li, W. Location normalization for information extraction. Proc. 19th COLING, Aug. 2002, Taipei, Taiwan

[15] Li, H., Srihari, R. K., Niu, C., and Li, W. InfoXtract location normalizations: a hybrid approach to geographic references in information extraction. Workshop on the Analysis of Geographic References, May 2003, Edmonton, Canada

[16] Ma, Q., Matsumoto, C., and Tanaka, K. A localness-filter for searched web pages. 5th Asia Pacific Web Conference (APWeb'03), Xi'an, China, Sep. 2003

[17] Ma, Q. and Tanaka, K. Retrieving regional information from web by contents localness and user location. 1st Asia Information Retrieval Symposium (AIRS'04), Beijing, China, Oct. 2004

[18] Markowetz, A., Chen, Y., Suel, T., Long, X. and Seeger, B. Design and implementation of a geographic search engine. Technical Report TR-CIS-2005-03, Polytechnic University, Brooklyn, New York, 2005

[19] McCurley, K. S. Geographic mapping and navigation of the web. 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001

[20] Microsoft MapPoint. http://mappoint.msn.com

[21] MSN New York local page. http://local.msn.com/NewYork/

[22] Place names Information System (GNIS). http://geonames.usgs.gov

[23] MSN Portal. http://www.msn.com

[24] North American Numbering Plan. http://sd.wareonearth.com/~phil/npanxx

[25] USPS – The United States Postal Services. http://www.usps.com

[26] Wang, C., Xie, X., Wang, L., Lu, Y., and Ma, W.Y. Web resource geographic location classification and detection. In Proceedings of the 14th International World Wide Web Conference (WWW'05), poster, Chiba, Japan, 2005

[27] Woodruff, A.G. and Plaunt, C. GIPSY: geo-referenced information processing system. Journal of the American Society for Information Science, 45(9), 1994, 645-655

[28] Yahoo Regional. http://www.yahoo.com/regional

[29] Yokoji, S., Takahashi, K., and Miura, N. Kokono search: a location based search engine. 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001