# Application-Driven Web Resource Location Classification and Detection

Chuang Wang*, 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

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 locations from web resources has become a key challenge to location-based web applications. Previous work has been focusing on deducing web locations from various geographical sources such as geographical names, postal codes, telephone numbers, and so on. None of them, however, notice the intrinsic differences between application needs for different types of locations. Multiple locations may co-exist in a web resource; designing a general-purpose algorithm ignoring their differences usually leads to a low detection precision. In this paper, we first explicitly distinguish the locations of web resources into three types to cater to different application needs: 1) *provider location*; 2) *content location*; and 3) *serving location*. Then we describe a novel system that computes each of the three locations, employing a set of algorithms and different geographical sources. Experimental results on large samples of web data show that our solution outperforms previous approaches. Finally, we identify some promising web applications based on the three proposed locations.

## 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

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

## 1. INTRODUCTION

Intuitively, web resources (including web pages, sites, etc.) have geographical features [3][8][14][20][21][22]. For instance, a web page with information about or within a special geographical scope, such as listings on houses for sale in a given region, could be regarded as a local page with a certain location. In contrast, another page with general information, such as an introduction of mathematics knowledge that is more likely to interest users from any locations, could be considered as a global page.

With rapid pervasion of the web into users' daily lives, in the increasingly popular mobile and local search environments, location-based web applications are emerging (such as web geographical information navigation and retrieval, location-based search, local advertisements, and context-aware services etc. [15][16]). The common principle of these applications is to detect the geographical attribute from web resources then match it with current user's location. User location can be acquired from his/her context environment or be provided explicitly. Therefore, how to correctly and effectively deduce the web location, taking full advantage of relevant geographical sources, is the key to the success of these location-based web applications.

Due to the importance of geographical features of web resources, much work has been carried out to improve the accuracy in web location detection and estimation. None of them, however, notice the intrinsic differences between application needs of locations. That is, multiple web location definitions co-exist in a web resource, and different web applications may need different web locations. Ignoring the type of location an application needs or using a wrong type of location may result in undesirable results.

Having investigated different needs on web locations from a number of location-based web applications, we conclude that there are at least three types of web locations that co-exist in the same web resource. Most geographical information navigation and retrieval work, as well as location-based search, usually focuses on extracting the geographical location of the web resource content. Since organizations or persons often issue their contact information on their web sites, generally in the home or contact page, web resources also contain the physical location attribute at site level. Services such as Map or Yellow Pages can benefit from collecting this type of location. Each web resource also has its affecting or serving geographical scope, which can be estimated by the geographical distribution of its access users or inbound

hyperlinks. This type of location should be of interest to business owners of the web resources, marketers or advertisers.

In this paper, we explicitly distinguish the locations of web resources into three types, namely *provider location*, *content location* and *serving location*, to cater to different application needs. Although some researchers have proposed various geographical sources for estimating web resource locations, none of them, to our best knowledge, have explicitly classified web locations into types and stated the intrinsic differences among these types.

We also introduce a novel system employing a set of algorithms to compute the three types of locations by extracting geographical information from the web resource content, and mining hyperlink structures as well as user logs. Only relevant geographical sources (thus not all of possible sources) are extracted and used in computation of each type of location.

The main contributions of this paper include:

- Locations of web resources are classified into three categories: *provider location*, *content location*, and *serving location*.
- A system is devised, employing a set of novel algorithms, to compute all types of the locations from a given web resource, only using relevant geographical sources for each location type.
- Our algorithms not only take geographical names, but also consider other geographical information sources such as user access logs, hyperlink structure to achieve better location detection accuracy. We have also introduced an extensible differential weighing structure to represent support of true geographical entities from all types of geographical sources.
- Some promising web applications based on our proposed three types of locations are identified.

We carried out experiments on large samples of real-world web data to evaluate the quality and performance of our algorithms against a generic (i.e., none-typed) location detection algorithm. The findings and comparisons should also be beneficial to various location-based web applications.

The rest of the paper is organized as follows. Section 2 defines provider location, content location and serving location. The system and algorithms for computing these locations are presented in Section 3. Section 4 provides the experimental results to evaluate the quality and performance of our approaches. In Section 5, some promising applications based on the three locations are discussed. Section 6 describes related work. Finally, we conclude the paper and discuss our future work in Section 7.

# 2. WEB LOCATION TYPES

To better understand the definitions of provider location, content location and serving location, we first introduce a few basic characteristics of geographical location and our assumptions used in this paper.

## 2.1 Location Characteristics

- **Local vs. Global.** Web resources that have identifiable geographical features are local. Not all web resources have geographical features when it comes to their content. The locations of these web resources are therefore global.
- **Divisional Perspective.** Administrative district and natural geography are two general perspectives to divide and organize geographical locations. Administrative district perspective is used in this paper, considering that most geographical references on the web are from this perspective.
- **Geographical Representation.** Among possible geographical representations such as latitude-longitude pair, postal code, telephone number, and geographical name [3][22], geographical name is more convenient for expressing location hierarchy and can be transformed to other presentations easily. Therefore, all references of geographical locations in this paper are expressed in geographical names.
- **Point Location vs. Shaped Region.** A web location can be a point location, an arbitrarily shaped region, or a combination of both (because a web resource may be relevant to multiple locations [5]). In our proposed definitions, provider location is often a point location, while content location and serving location are shaped regions in most cases.
- **Highest Level Expression in Location Hierarchy.** Due to the hierarchical structure of geographical locations, given a web resource, one can choose a certain geographical level to describe detected locations. In this paper, we will always use the highest level of description. For example, if a web page covers all 50 USA states, we will use the country of USA instead of the collection of all USA states to present the web location of the page.

## 2.2 Location Type Definitions

A web resource may have the following three different types of locations:

- **Provider location:** *The physical location of the provider (organization, corporation or person) owning the web resource*. This location is crucial to web geographical information retrieval and navigation such as online map and Yellow Pages services.
- **Content location**: *The geographical location that the content of the web resource is about*. As a spatial attribute of the web content, this location can be used to classify and organize web resources to better satisfy user' information needs. One of its applications is location-based search.
- **Serving location** *The geographical scope that the web resource reaches*. Knowing the serving location of a web resource can benefit many business applications such as local advertisements and e-commerce.

Figure 1 is an illustration of these three location types. In this example, the provider location (contact address of the web site owner) is in state of Oklahoma, USA. The content location of page 1 (which is about the tourism in Nevada) is state of Nevada, USA. The serving location (user reach) covers mid-region states of USA.

We will take MSN site [24] as another example to further illustrate these three types of locations. On the site, our algorithms found that provider location of the site is "Microsoft Corporation One Microsoft Way Redmond, Washington 98052, USA", namely the physical location of Microsoft Corporation. And the computed serving location is "Global" due to msn.com is a general web site with world-wide user reach. The content location of MSN's New York local page [25] is "New York, NY, USA".

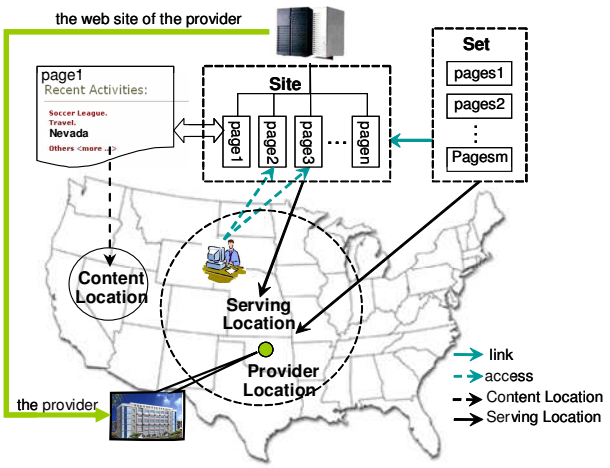


**Figure 1. An illustration of provider location, content location and serving location.**

In real world, every entity, whether it provides "products" or "services" to its intending users, has its physical location and its affecting geographical scope, namely provider location and serving location, respectively. Furthermore, the two locations usually have different geographical scopes and scales. Generally, provider location is a point location where the provider locates or multiple point locations if the provider entity is geographically distributed. Serving location is often a shaped region or multiple regions where its "products" or "services" can reach.

Content location is the spatial feature of web content and is usually considered more meaningful at page level, even at block level [26], while both provider location and serving location are spatial attributes of the entity behind web resources and are often detected at the site level. Content location is the common focus of previous work. We will also pay attention to the other two locations in our discussions in following two subsections.

It is especially noticeable that, as the closest work to ours, Ding et al. in [5] have attempted to define the geographical scope of a web resource as "the geographical area that the creator of the web resource intends to reach". In fact, this definition is more similar to our proposed serving location, but it was mistaken for content location as it is computed using content-based technique.

## 2.3 Usage Scenarios

As the spatial attribute of web content, content location is crucial to classify, organize and retrieve web information. Currently, more and more business applications based on content location are emerging, such as location-based search.

Provider location and serving location have different functions, although relationship often exists between them, such as the latter is based on and usually hierarchically contains the former. Provider location can be applied in web geographical information navigation and retrieval services such as map and Yellow Pages, which need definite and accurate physical locations. Provider locations of some entities such as hospitals, theaters and shopping centers are more important to users, when they have to go to physical locations of these entities to consume "products" or "services". In contrast, serving locations of other entities such as food delivery services are more interesting to their users since users will care for whether their current locations are within the serving scopes of these entities.

Obviously, due to different meanings and functions, one location type cannot or should not be used in place of another. Returning to our previous MSN example, supposing that you would like to visit the MSN team, you will need to know the provider location. On the other hand, advertisers from around the world will be more interested in the globalness of MSN's serving location. A user from New York city may find the New York local page on MSN useful to him/her since the content location of the page is his/her city.

## 2.4 Sources for Computing Locations by Type

Table 1 lists a number of geographical sources that are useful to estimate the three locations.

**Table 1. Sources for computing the three types of locations.**

| Location type | Geographical sources |
|---|---|
| Provider location | Yellow Pages, address information databases, address segments in web content, etc. |
| Content location | Location information appears in web content such as geographical names, telephone numbers, postal codes, people/organization names, geographical meta-datas, and languages, etc. |
| Serving location | Hyperlinks, user access logs, etc. |

**For provider location**. Provider location of some web resources can be easily acquired from existing business databases such as Yellow Pages or other commercial contact information databases. Unfortunately, this is not true for many small businesses, organizations, or persons. For them, building a web site to introduce themselves and release their contact information is a more feasible and convenient way of announcing their provider location to the world than paying for and registering themselves in some business databases. Thus, more generally, we can acquire provider locations of non-registered entities based on their web content.

In many cases, the provider location of a web site can be found in its contact page or those pages under the same path if its contact information is distributed in more than one page. There are also some sites that do not have explicit contact pages and they tend to issue the provider location on their home pages. Generally, a common clue to identify the provider location is that geographical name, postal code, and telephone number should appear together, or at least the former two, in an address segment.

**For content location.** Content location can be computed by extracting geographical keywords from the web content that are significant to the content. We will also have to deal with geographical name false positives and ambiguities. Most of existing research on web location detection falls in this category [3][10][19][20][21][29].

**For serving location**. When user logs of a given web resource are available, we can calculate its serving location by analyzing the users' geographical distribution. However, since the access logs of web resources are usually unavailable outside the resource providers' organizations, hyperlinks between web pages are often the remaining main clue for estimating the serving location. Here

we state that serving locations of web resources can be transferred along hyperlinks. That is, given a web resource *w*, if the serving location of a significant fraction of web resources that have links to *w* is location *l*, then the serving location of *w* is *l*. The experimental results in Section 4 support our rationale.

# 3. LOCATION DETECTION BY TYPE

## 3.1 Workflow

Since existing algorithms for computing web location do not distinguish among provider location, content location and serving location, a more appropriate approach, taking into consideration of different and unique characteristics of these location types, is needed to achieve better location detection accuracy.

Figure 2 shows the workflow of our novel location detection system by the three location types. We first utilize the extracted address segments to acquire the provider location. Then all extracted geographical keywords will be used to compute the content location. The serving location is estimated based on the content locations computed in the above step, plus the geographical information carried from inbound hyperlinks and/or from user logs.

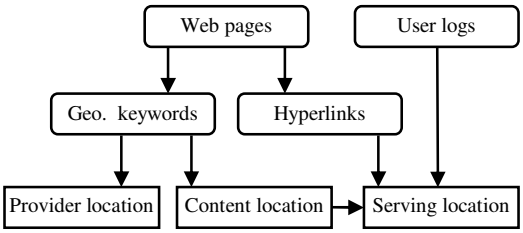


**Figure 2. The flowchart of location detection system by location types.**

In our system, we calculate all three types of locations from a given web resource, using only relevant sources for each type. We also consider user information from logs to aid in location detection accuracy.

## 3.2 Computing Provider Location

Challenges of computing the provider location of a web resource lie in: 1) to accurately recognize and extract address strings from the web content, and then 2) to correctly estimate whether extracted address strings are the provider location.

First, some concrete address segments must be recognized and extracted from the web content. The common representing format of contact addresses can facilitate this recognition. For example, in the USA, the contact format is usually "street address, city, state, Zip code, country or region". In addition, address initializations and abbreviations, along with punctuations and separating tags of HTML, are all important clues to improve extraction precision.

After an address string is recognized and extracted, we employ Support Vector Machine (SVM) [1], which has been found quite effective for text categorization problems [9], to learn whether the string is a provider location in the binary classification setting.

Our investigational results found the following features can facilitate SVM to estimate provider location: URL, title, anchor text, page content, referred frequency, hierarchical level, and even the spatial position of extracted address strings on the page.

The title of a web page usually describes the topic of the page content. The URL and anchor text of a web page usually also summarize the page content [17]. We also found that the provider location often appears in multiple pages of a site, defined by a common page template used by the site. Thus, more times an address string is referred, higher the possibility the string is a provider location. In addition, in most sites, web pages that contain provider locations are often habitually placed in the first or second level directory of these sites. We also found that a provider location is more often referred in the page footer.

In this research, we developed an integrated and effective solution for estimating provider locations of web resources. First, we investigated a number of web sites, and identified useful features to help the correct recognition and extraction of provider address strings from web content. Then we fed these features into the SVM model to estimate the provider location.

## 3.3 Computing Content Location

We borrow two basic measures: *power* for measuring interest and *spread* for measuring uniformity, from the CGS/EGS approach. Authors in [5] first defined these two measures, and then pointed out that the geographical scope of a web resource must satisfy two conditions: smooth distribution (CGS, the candidate geographical scope) and then significant interest (EGS, the estimated geographical scope), namely enough spread and power. Content-based and link-based techniques are proposed to estimate the geographical scope.

In our algorithm, the power is extended as the following to satisfy the needs of our approach. Given a web resource *w* and location *l* in the location hierarchy:

$$Power(w,l) = Wt(w,l) + \sum_{j=1}^{n} Wt(w, Child_j(l)) + \sum_{i=1}^{m} \frac{Wt(w, Ancestor_i(l))}{|Sibling_i(l)|} \quad (1)$$

where $Wt(w, l)$ is the weight of $l$ in $w$; $Child_j(l)$ means a direct or indirect offspring location node of $l$ ($1 \le j \le n$, $n$ is the number of all offspring nodes of $l$); each $Ancestor_i(l)$ denotes a direct or indirect higher-level location node of $l$ ($1 \le i \le m$, $m$ is the number of all higher-level nodes of $l$ in the hierarchical tree), $|Sibling_i(l)|$ is the number of location nodes in the same level with $l$ within the sub-tree with $Ancestor_i(l)$ as the root.

In the new definition of power, our contribution is that we comprehensively consider weights of both ancestor and offspring nodes, in addition to the weight of current location node. Since referring a node also means indirectly referring all its ancestor nodes due to the belong-to relationship, it is more reasonable to traverse the location hierarchy from bottom up, and increase the weight of each node with the sum of its offspring's weights. By the same token, referring an ancestor node also means referring all its offspring nodes, the weight of each node is increased by a fraction of the weight of its ancestors[1]. We believe that our power definition better represents the "weight" of each node in the hierarchical tree, and our experimental results comparing our approach and CGS/EGS support that.

[1] Here, the weight of ancestor node is simply averagely divided to each its children nodes. In fact, there exist more complicated distributing strategies if considering the geographical scope or population density of each children node.

As already mentioned, extracted geographical keywords are the data sources for computing the content location. In our approach, we further expand the exploited geographical keywords from only using geographical names (as in CGS/EGS) to including more reliable postal codes and telephone numbers. We know that some geographical names are ambiguous and deficient to identify a location, such as those geographical names with multiple corresponding nodes on the location hierarchy, or those can be used in a non-geographical context (such as people names, etc.). Therefore, in our approach, we assign different weights to different kinds of geographical keywords. Individual geographical names will also have different weights, according to their supports to true geographical locations.

In summary, another contribution from us is to assign a weight to a given geographical keyword, *gk*, by its category:

$$Wt(gk) = \begin{cases} Wt(zt) & \text{when } gk \text{ is Zip or telephone number} \\ (1\text{-}Wt(zt)) \times (ldf(gn)/idf(gn)) & \text{when } gk \text{ is geo. name} \end{cases} \quad (2)$$

where, $Wt(zt)$ represents the common weight of Zip code and telephone number, $ldf(gn)$ is the document frequency of each geographical name ($gn$) as a geographical keyword, and $idf$(gn) denotes the document frequency of $gn$ as a general keyword.

Due to the high reliability of Zip codes and telephone numbers in correctly identifying unique geographical locations found in our experiments, we do not distinguish them and use the same constant $Wt(zt)$ (is greater than zero but less than one) to represent their common weight.

Weights of all geographical names are subject to a common factor (1-$Wt(zt)$). In essence, we use $Wt(zt)$ to control the balance of weights between Zip codes/telephone numbers and geographical names to achieve the best possible accuracy. The weight of each individual geographical name is further adjusted by $ldf(gn)/idf(gn)$. To calculate $ldf(gn)$ and $idf(gn)$, two document corpora are needed. One is a geographical relevant document corpus, where we assume that each reference of $gn$ is from geographical perspective and $ldf$ ($gn$) is calculated according to the referred frequency of each geographical name. The other is a general document corpus that is used to compute $idf(gn)$. underlying reasoning of weighing each geographical name by the ratio of $ldf$ ($gn$) to $idf(gn)$ comes from the observation that a geographical name will be more likely to be truly about a geographical entity if it is more frequently referred in the geographical corpus than in the general corpus.

For a given web resource $w$ and a location $l$, the weight of $l$ in $w$, namely $Wt(w, l)$ is defined as:

$$Wt(w,l) = \sum_{gk} rf(w, gk) \times Wt(gk) \quad (3)$$

where $gk$ (geographical keyword) stands for any possible representations or aliases of $l$; $rf(w, gk)$ is the referred frequency of $gk$ in $w$, $Wt(gk)$ is the weight of $gk$.

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

$$Spread(w,l) = \frac{-\sum_{i=1}^{n} \frac{Power(w,l_i)}{\sum_{j=1}^{n} Power(w,l_j)} \times \log\left(\frac{Power(w,l_i)}{\sum_{j=1}^{n} Power(w,l_j)}\right)}{\log n} \quad (4)$$

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 power and spread are defined, content location can be computed by traversing the geographical hierarchy in a top-down fashion starting from the root node. For each current node, the sub-tree with itself as the root will be pruned if its *spread* or *power(node)/power(parent)* value does not exceed given threshold $Ts$ or $Tp$, respectively. Otherwise we continue traversing to its offspring nodes if there are any. When the traverse stops, the remainder leaves in the hierarchal tree constitute the content location. At the beginning of traverse, a given threshold value $Tw$ is used as the weight of "the parent" of root node.

In this content location computation process, we modified the power definition to include effects from both ancestors and off-springs on the geographical hierarchy. Furthermore, in addition to having geographical names, we also consider more reliable geographical sources such as postal codes and telephone numbers in the power calculation. We also introduced weight factors to control the balance between different types of geographical keywords, as well as weighing each geographical name by their likelihood to be truly about a geographical entity. Our more comprehensive understanding of web resource locations gives us better accuracy in our experiments.

## 3.4 Computing Serving Location

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

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 location hierarchy, the weight of $l$ in $w$, namely $Wt(w, l)$ is calculated as follows:

$$Wt(w,l) = \begin{cases} \alpha_1 UserFreq(w,l) + (1\text{-}\alpha_1) ContentLoc(w,l) & i = 0 \\ \alpha_2 \sum_{j=1}^{n} SrvLoc_{i-1}(w_j, l) + (1-\alpha_2) SrvLoc_{i-1}(w,l) & i > 0 \end{cases} \quad (5)$$

where, $Userfreq(w,l)$ is $w$'s access frequency by all users within location $l$; $ContentLoc(w,l)$ 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 $ContentLoc(w,l)$, $SrvLoc_{i-1}(w,l)$ 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 5 when i=0. Then using the same power and spread definitions and computation process of 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.

# 4. EXPERIMENTS

## 4.1 Settings

### *4.1.1 Geographical Thesauruses*

To recognize and extract the geographical keywords referred in web resources, a set of geographical thesauruses must be constructed in advance. First, needed location entities are collected from various sources, including USA Zip codes from [13], telephone numbers from [12], and geographical names from [11]. Since the widest geographical coverage of our test data set is in the country of USA, we collected data sources within the country. After analyzing and integrating these location entity data, we constructed four tables to constitute our geographical thesauruses: the hierarchical location tree table represented with standard geographical names, geographical name table including various geographical aliases, Zip code table, and telephone number table. The latter three tables each contain a standard geographical name attribute that points to the location in the first table.

Given a Zip code (we extracted the first 5 digits if it was in the Zip+4 format), there were usually several corresponding geographical names in our geographical data sources. One of them is the standard geographical name for the Zip code while the others are aliases of the standard name. After synthetically considering Zip codes and corresponding geographical names, we could obtain the standard hierarchical location tree table, the geographical name table and the Zip code table. Then the telephone number table is constructed by synthesizing the standard hierarchical location tree table and telephone number information.

As the result, our USA geographical hierarchy contains country (USA only), state (all 50 states, Washington DC, and other 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.

### *4.1.2 Web Resources*

The benchmark data set used in our experiments is a collection of real web resources of major USA governmental sties whose top domains are .gov. These data were crawled in 2002 and used in TREC2003. The data has a wide geographical range covering all levels of the USA geographical tree.

In our experiments, content location was computed at web page level, considering that the content location is often specific to individual web pages and changes from one page to another. Provider location and serving location were computed at web site level. In essence, we adopted two different granularities when it comes to web locations: page level for content location and site level for provider location and serving location.

We utilized the top three levels of domain name to distinguish different web sites. For example, for jsc.nasa.gov and jpl.nasa.gov, although obviously both of them belong to the same upper domain nasa.gov, we treated them as two different sites for simplicity. After eliminating those sites with fewer than 5 pages, which we think are improperly crawled or categorized, we were left with 4,430 sites and 1,053,111 pages to test our algorithms with.

**Table 2. Distributions of geographical keywords.**

| Keywords | Occurrence | Page (1,053,111) | Site (4,430) |
|---|---|---|---|
| Zip | 919,170 | 232,344 (22%) | 3,143 (71%) |
| Telephone | 1,139,677 | 236,516 (22%) | 3,191 (72%) |
| Geographical Name | 80,652,212 | 822,219 (78%) | 4,116 (93%) |
| Zip or Telephone | 2,058,847 | 323,587 (31%) | 3,440 (78%) |
| **Any of the three** | **82,711,059** | **835,969 (79%)** | **4,133 (93%)** |

The above table shows the distributions of the three geographical keywords in our test data. Since Zip codes rarely appear outside address segments (which are likely the provider locations of web resources), there are about 71% of sites that can have provider locations. About 79% of pages that contain at least one of the three types of geographical keywords. Content locations possibly exist in these pages.

Table 2 also shows that the distribution of Zip code is similar to that of telephone numbers. After having compared the computational results under different weights of them, we find that their confidences in estimating locations of web resources are also close. Therefore, the same weight $Wt(zt)$ is assigned, regardless whether a keyword is a Zip code or a telephone number.

Out of all 4,430 sites, 1,000 sites were randomly selected and manually labeled with provider locations and serving locations. One web page was also randomly selected from each chosen web site and labeled with its content location. Furthermore, we utilized the labeled web resources to complete the geographical name table in our geographical thesauruses. As already mentioned, two document corpora was needed to decide the individual weight for each geographical name. In our experiments, documents from 829 sites whose serving location were labeled as local, out of these 1,000 labeled sites, constitute geographical relevant document corpus. The general corpus includes all these 4,430 sites.

### *4.1.3 Evaluation Method*

In our experiments, precision is used to measure the fraction of locations in the computational results that are correct, and recall is used to measure the fraction of locations in the labeled data that are captured in our computational results. Note that precision often can be increased at the expense of recall, and vice versa. Therefore we combine precision and recall into a single metric using the F-measure [27]:

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (6)$$

In addition, due to the sheer volume of web resources that need to have web locations detected, the processing time has become a critical factor to their performance in location-based web applications. Therefore, we also report the running time cost of

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

## 4.2 Finding Best Parameters

For provider location, we implemented our approach using several SVM variations to select the best algorithm and parameters. The features used in our experiments include URL, anchor text, hierarchical level of current page. As the results shown in Table 3, the SVM using Gaussian kernel achieved the best performance with F-Measure 0.94 in our experiments. The results also indicate that a nonlinear combination of the features is better than a linear combination.

**Table 3. Comparison of SVM learning methods.**

| Methods | Precision | Recall | F-Measure |
|---|---|---|---|
| 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 |

We compared our proposed algorithm with CGS/EGS approaches of both content-based and link-based techniques. Figure 3 shows the impact of *Wt(zt)* on the F-measure for these algorithms. To make fare comparisons, we extended the data sources of CGS/EGS approaches to include Zip codes and telephone numbers.

When *Wt(zt)* is less than 0.3, increasing it does not affect the F-measure much. This is mainly due to the fact that Zip codes and Telephone numbers are much fewer in web resources compared with geographical names. We also observe that the impact of *Wt(zt)* in our algorithm is similar to that in CGS/EGS. The best F-measure can be acquired when *Wt(zt)* is around 0.8 for each algorithm. Figure 3 also shows that the impact of *Wt(zt)* is lighter on serving location, since the serving location more heavily depends on hyperlink structures.

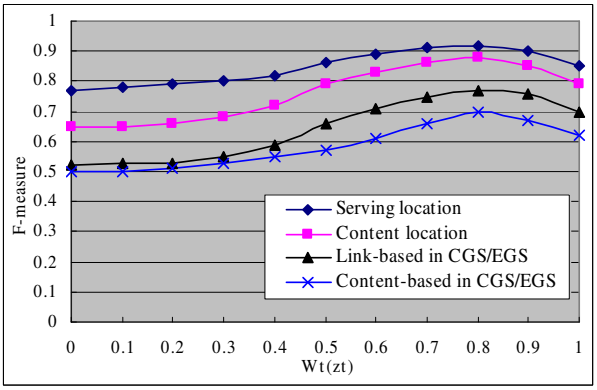


**Figure 3. Impact of Wt(zt) on F-measures.**

Figure 4 shows the impact of *Ts* (threshold for spread) in computing the serving location. Impact of *Ts* on content location and on CGS/EGS content-based and link-based geographical scopes are similar to what is shown in Figure 4. The data tell us that the best F-measure can be acquired when *Ts* is around 0.75 (0.7 for both techniques of CGS/EGS). From Figure 4, we also find that when *Ts* is larger than 0.8, recall and F-measure will drop dramatically, this means that an extreme large spread threshold will result in major serving locations being excluded.

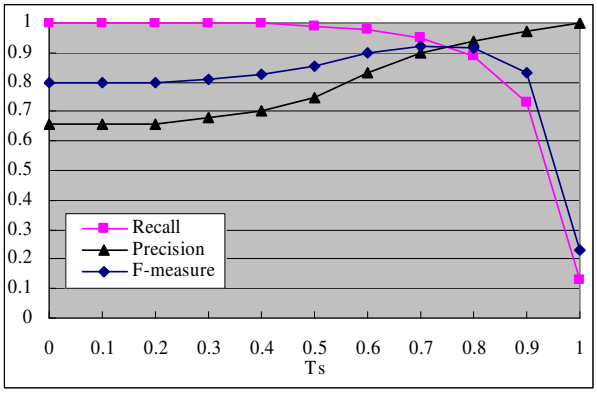


**Figure 4. Impact of *Ts* on serving location.**

**Table 4. Parameters used in our experiments.**

| Parameters | CGS/EGS | | Our algorithm | |
|---|---|---|---|---|
| | Content based | Link based | Content location | Serving location |
| Wt(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 |
| Tw | - | - | 1.00 | 4.00 |
| $\alpha_1$ | - | - | - | 0.85 |
| $\alpha_2$ | - | - | - | 0.00 |

Table 4 is the summary of parameters chosen by the best F-measures from our experiments. Since web logs were unavailable in our experiments, $\alpha_2$ was set to 0. Finally, our experimental environment is a machine with Intel Xeon CPU 3.06 GHz, 2 GB RAM and running Microsoft Windows Server 2003.

## 4.3 Results

Table 5 displays the percentages of the pages/sites that have non-global locations and the corresponding precisions of our algorithms.

**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 location and serving location | | **758 (76%)** |

About 71% of the web sites have issued their provider locations, and 96% of them can be computed correctly by our algorithm. We have mentioned earlier that about 79% of the pages contain at least one of the three types of geographical keywords, which means that content location of these pages can be computed in theory. However, our labeled results show that only 54% of the pages are local. This is because: 1) some keywords are used in non-geographical context, e.g. people names, and 2) some referred

geographical keywords are not significant enough to be the content location of the page.

Our results also show that 76% of the pages in the test set have different content locations and serving locations (of the same site). This proves the necessity to distinguish between content and serving locations.

**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. of 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** |

Table 6 is the summary of our experimental results. The results show that our proposed algorithms achieve better performance in both F-measure and computational cost, compared with CGS/EGS. For example, the serving location computed by our proposed algorithm outperforms both the content-based and link-based methods of CGS/EGS by 18% and 15% in F-measure, respectively. Both our serving location algorithm and link-based CGS/EGS algorithm use links, but our algorithm is 3 times as fast as the CDS/EGS algorithm.

Having analyzed the results, we find that the improvements mainly come from 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 geographical sources and intrinsic characteristics.
- Our proposed algorithms of computing content location and serving location start from the root node and do pruning. Location nodes with abnormal spread and power value caused by the ambiguity of geographical names 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 location 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 removed. 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 location hierarchy. Obviously, it is more reasonable to increase the weight of current node with partial that of its ancestor nodes and total that of its offspring nodes considering the belong-to relationship existing among them.

# 5. IMPORTANT APPLICATIONS

Web location plays an important role in a wide range of business applications [15][16][23][28][30]. As shown in Table 7, different applications should be using different types of web locations.

For example, location-based search engines may benefit from each of our proposed three types of web locations in different aspects: First, the search quality can be refined according to the relevance of content locations of web resources with users' current location. Second, we can further improve users' browsing experience through displaying provider locations of the returned results on a map. Finally, we can use serving locations of the results to estimate the geographical localness of search query, a critical aspect of location-based search. That is, given a query, if serving locations of a majority of the search results are non-global, we think the query is local.

**Table 7. Applications based on different web location types.**

| Location Type | Applications |
|---|---|
| Provider location | Map services, Yellow Pages, driving directions, travel/shopping guidance, customer address management, real estate management etc. |
| Content location | Location-based search, web information classifying, organizing and retrieval, geographical information navigation and retrieval etc. |
| Serving location | Local advertisements, e-services such as e-commerce, personalized services, target marketing etc. |

Another application of the provider location is web geographical information navigation and retrieval. For web resources with released provider locations, we can visually display the locations on a map and offer users with more relevant geographical context information about the locations (such as the recent local news). The function is similar to traditional Yellow Pages, but both the scale and the coverage from utilizing extracted provider location from web resources are far greater than what Yellow Pages can provide.

Serving location can be applied to facilitate the deployment of local advertisements. USA alone has about 10 million small- and medium-sized enterprises (SMEs). Most of those SMEs, conduct the majority of their business within 50 miles of their business locations. Although the market of local advertisements is huge, the full potential of it has yet to be materialized in online marketing. SMEs, due to their limited resources, would rely on marketing service providers to detect accurate serving locations to target SME's online users.

There are a large number of applications based on different locations of web resources. We just discussed a few here. Recently, location-based web mining [14][18] becomes a hot topic. Using the achievements of location-based log mining, e-services, especially e-commerce, can adapt their strategies and provide personalized services according to the geographical context of their users.

# 6. RELATED WORK

Due to the importance of geographical features of web resources, much work has been carried out to improve the accuracy of detecting web locations. There are two major identifiable research

directions: 1) exploiting various geographical sources (such as IP, Whois databases, route packet, DNS, Postal code, telephone number, geographical name, people/organization name, language, geographical metadata, Hyperlink and user access log [3] etc.), and 2) developing effective computation approaches.

Ding in [5] proposed the CGS/EGS algorithm based on geographical content and context sources. In that algorithm, they defined two key measures: power for measuring interest and spread for measuring uniformity, and then proposed that the geographical scope of a web resource must satisfy two conditions: significant interest and smooth distribution, namely enough power and spread.

There exist many location-based web applications, such as Yahoo Regional [30], Microsoft MapPoint [23], Google Local Search [7] etc. Most of these location-based applications, however, are founded on Yellow Page databases or manual classification of web resources. Thus, they merely transfer traditional geographical information services from off-line to on-line, rather than utilize the ubiquitous geographical features of web resources. Hence, many non-profit organizations or persons, even some small business not recorded in the Directory or Yellow Pages are ignored, although their geographical scopes can be acquired from the web.

Other systems such as Columbia GeoSearch [4], Geotags GeoSearch [6] and Kokono Search [31], have attempted to estimate web location by extracting geographical keywords from the web content. However, these systems employing location detection techniques suffer from the ambiguity and aliasing of geographical names, the narrow coverage of geographical thesaurus, the low precision of geographical name recognizing algorithms and the inefficiency of web location estimating approaches. Traditional applications based on Directory or Yellow Pages are still as good if not better than these research systems.

In our opinion, one key factor is that these efforts have ignored the intrinsic differences of various location types. Therefore, designing a general-purpose algorithm usually leads to a non-satisfactory detection precision.

# 7. CONCLUSIONS AND FUTURE WORK

In this paper we categorized web locations into distinctive types by their business needs, and discussed a novel approach to compute them. We first defined the following web location types: *provider location*, *content location*, and *serving location*, and described their unique characteristics by examples. Different business applications need different types of web locations. Multiple locations often co-exist in the same web resource. Ignoring location types or choosing a wrong type to use could result in poor detection accuracy in web applications.

We proposed a novel system that compues all types of locations from a given web resource. It employs a set of effective location detection algorithms and only uses relevant data sources for each location type to achieve high accuracy and fast speed. To improve the accuracy, we extended some existing algorithms by including more reliable data sources such as postal codes, telephone numbers, and locations of visiting users, in addition to geographical names. We also introduced an extensible differential weighing structure in our algorithms to represent geographical supports from different types of geographical sources as well as from individual geographical names.

Experimental results on a large set of web sites showed that our approach outperforms a generic location detection algorithm that does not distinguish location types in both accuracy and speed measures.

We are planning to implement a location-based search engine based on our current results. In the system, the retrieved results are further filtered and refined, so that just web pages whose content locations are related to users' current location are returned to the users. Furthermore, if necessary, provider locations of returned sites will be displayed in a map to improve users' search experience. In addition, advertisements will be more relevant to the users since only those whose serving locations are overlapped with users' current locations are pushed to users. Finally, the geographical context of the search engine will be expanded to be capable to deal with more countries.

There are some open issues that need further research. For example, besides the proposed three types of locations, one need to answer whether there are any other types of web locations. Another question is whether our methodology can be applied to other aspects or entities of location-based applications. For instance, we need to investigate on whether web resource users have intrinsic and distinctive location types. If they do, we need to find out the best matching between web resources and user location types. Similar to users, we need to conduct research on location attributes to search queries, i.e., quantification and classification of geographical context of search queries.

# 8. REFERENCES


[1] Boser, B.E., Guyon, I.M., and Vapnik, V. A training algorithm for optimal margin classifiers. 5th Annual Workshop on Computational Learning Theory (COLT'92), Pittsburgh, USA, Jul. 1992.

[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 geographical 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 geographical 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] Gravano, L., Hatzivassiloglou, V., and Lichtenstein, R. Categorizing web queries according to geographical locality. 12th ACM Conference on Information and Knowledge Management (CIKM'03), New Orleans, USA, Nov. 2003.

[9] Hearst, M.A. Trends and controversies: support vector machines. IEEE Intelligent Systems, 13(4), Jul. 1998, pp18-28.

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

[11] Geographic Names Information System (GNIS). http://geonames.usgs.gov/

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

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

[14] Iko, P., Takahiko, S., Katsumi, T., and Masaru, K. User behavior analysis of location aware search engine. 3rd International Conference on Mobile Data Management (MDM'02), Singapore, Jan. 2002.

[15] 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.

[16] Kaasinen, E. User needs for location-aware mobile services. Personal and Ubiquitous Computing 7(1), May 2003, pp70-79.

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

[18] Kosala, R. and Blocakeel, H. Web mining research: a survey. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00), Boston, USA, Aug. 2000.

[19] 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, pp 81-123.

[20] 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.

[21] Ma, Q. and Tanaka, K. Retrieving regional information from web by contents localness and user location. 1st Asia Information Retrieval Symposium (AIRS2004), Beijing, China, Oct. 2004.

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

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

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

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

[26] Song, R.H., Liu, H.F., Wen, J.R., and Ma, W.Y. Learning block importance models for web pages. 13th International World Wide Web Conference (WWW'04), New York, May 2004.

[27] Van Rijsbergen, C.J. Information retrieval. Butterworths, London, second edition, 1979.

[28] Watters, C. and Amoudi, G. GeoSearcher: location-based ranking of search engine results. Journal of the American Society for Information Science and Technology, 54(2), 2003, pp140-151.

[29] Woodruff, A.G. and Plaunt, C. GIPSY: geo-referenced information processing system. Journal of the American Society for Information Science, 45(9), 1994, pp645-655.

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

[31] 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.