Detecting Geographical Serving Area of Web Resources

Qi Zhang[1], Xing Xie[2], Lee Wang[2], Lihua Yue[1], Wei-Ying Ma[2]

[1]Department of Computer Science,
University of Sci. & Tech. of China
Hefei, Anhui, 230026, P.R. China
wizard@mail.ustc.edu.cn, llyue@ustc.edu.cn

[2]Microsoft Research Asia,
5F, Sigma Center, No.49, Zhichun Road
Beijing, 100080, P.R. China
{xingx, leew, wyma}@microsoft.com

ABSTRACT

Most human activities occur around where the user is physically located. Knowing the geographical serving area of web resources, therefore, is very important for many web applications. Here serving area stands for the geographical distribution of online users who are interested in a given web site. It can be also seen as the geographical area that this web site intends to reach. For example, the serving area of a restaurant site is usually restricted to a town or city, while the serving area of an airport site can be as large as a state. In this paper, we proposed a set of novel methods to detect the serving area of web resources by analyzing search engine logs, our example of web usage data. We use the search logs to detect serving area in two ways. First, we extracted the user IP locations to generate the geographical distribution of users who had the same interests in a web site. Second, query terms input by users were considered as the user knowledge about a web site. From the experimental results, we found that the approach based on query terms was superior to that based on IP locations, since search queries for local sites tended to include location words while the IP locations were sometimes erroneous.

Keywords

Location-based web application, serving area, geographical information retrieval

1. INTRODUCTION

In this paper, we are interested in a special characteristic, "serving area", which can be regarded as the expected geographical distribution of online users who are interested in a given web site. For instance, in a search engine, given a query of "*pizza seattle*", the search engine should better return pizza related web sites whose serving area is *Seattle*. The serving area of a web resource can be different from the street address of the entity who owns that web resource.

Many research works have been carried out to detect the "geographical scope" of web resources. They are mainly based on analyzing web content and hyperlink structures. Geographical names, postal codes, telephone numbers and a number of other features are extracted from the web content to help get the geographical scope of a web page or a web site [1][2][3][4]. The underlying assumption is that if a web resource does have a non-global (thus local) geographical scope, it will be more likely to contain the location names or other named entities covered by the geographical scope. The geographical scope of a web resource can be used as an approximation to its serving area. However, geographical scope is different from serving area in that it describes content, not user. For example, www.newzealand.com has a clear geographical scope of New Zealand, but it will interest global users. Its serving area, therefore, should be global.

In this paper, we propose two novel methods for detecting serving areas by analyzing search logs, which are direct hints of user interests. Experiments on large samples of real world data are carried out to evaluate the performance of our algorithms.

2. SERVING AREA DETECTION

2.1 Computing Serving Area by Analyzing User IP Locations

In search logs, the relationship between user locations and clicked URLs can be estimated by analyzing the collection of user IP locations. In our algorithm, we use two measures: *weight* and *spread*, which were originally defined in [1]. *Weight* is used to measure the percentage of users in a certain location who are interested in a web site. *Spread* of a certain location is used to measure the uniformity of *weight* in its child locations on an administrative hierarchy. The user's interest here is regarded as the number of clicks on a web site URL in search logs. The more clicks on the URL, the higher interests the users put on.

In our paper, *weight* is defined as follows:

$$Weight(w,l\ ) = \frac{Click(w,l)\ /\ Population\ (l)}{Click(w,Pa\ rent\,(l)\ )/\ Population\ \ (Parent\ (l))} \qquad (1)$$

where *Click(w, l)* is the number of clicks on web resource *w* by people in location *l* (include its children). *Population (l)* is the population of location *l*. *Parent(l)* is the parent location of *l* on an administrative hierarchy.

In this definition we use population of a location *l* to weight *l*'s click counts. The reason is that we assume the number of web users in *l* is proportional to its population. Therefore the percentage of users who have clicked on a certain web resource in location *l* can truly reflect the interest degree of that location. Though more precisely, we need to use the Internet population, we do not use that simply because that type of data is not available to us.

Spread is defined as same as that in [1] and the entropy definition is chosen for the best performance based on their results.

GIR'06, August 10–11, 2006, Seattle, Washington, USA.

Once *weight* and *spread* are computed, user logs can be used to detect the serving area of a web site:

1. Map all user IPs to locations.
2. Map all the locations got from step 1 onto a geographical hierarchy, where location nodes distribute on different geographical levels such as country, state or city.
3. Travel the geographical hierarchy down from the root. For each node, *weight* and *spread* values will be calculated and the node will be pruned if its *spread* or *weight* values do not exceed given thresholds. Otherwise we continue the traveling to its offspring nodes if there are any. When the algorithm stops, the nodes where we stop at constitute the serving area.

2.2 Computing Serving Area by Analyzing Query Terms

When a user wants to find a local web resource, he or she is very likely to input a location term in the query. For instance, a user will input a query "*pizza seattle*" or "*seattle pizza*" if he or she wants to find some pizza related sites in *Seattle*.

From search logs, we can build up a relationship between query terms and user clicked URLs. If we get all the query terms which lead to clicks on URLS in a specific domain, we can then detect the geographical distribution by analyzing the location information in these query terms.

The number of query terms is often short, so it is more difficult to analyze query terms than to analyze web pages. In our algorithm, we solve geo/geo and geo/non-geo ambiguities by looking at query context. The query context here is the query terms input by other users. If we see query terms like "*lombardi's new york*", we don't know whether "*new york*" here stands for *New York City* or *New York State*, we will go forward to look at other query terms. From other query terms, we found that users have explicitly stated *New York City* instead of *New York State*, like "*lombardi's pizza nyc*" and "*Lombardi pizza in new york city*". This information is what we call query context. Finally, we know that "*new york*" here is more likely to represent *New York City* than *New York State.*

The serving area detection algorithm includes three steps:

1. Given a web site, extract all the related query terms and build a query document.
2. Run a content location detection algorithm on the query document. Any content location detecting algorithms can be applied here, such as the algorithm proposed in [4]. The query context will be considered in the algorithm.
3. The content location computed from step 2 is regarded as the serving area of the web site. The result can be seen as the user knowledge about the location information of the web site.

3. EXPERIMENTS

We used *precision* and *recall* to evaluate the performance of our algorithms. Serving area detection result is usually a collection of locations. Thus, *precision* and *recall* here represent the fraction of returned locations that are correct and the fraction of correct locations that have been returned, respectively. In our experiments, we use population to weight each location.

The log used in our experiments is a 30-day collection of search logs from MSN search [5]. We define web resources in the unit of web sites. Each URL will be converted to the corresponding domain. Besides, a commercial IP to location database is used in our experiments, which contains the mapping relationship between IP addresses and corresponding geographical locations.

3.1 Results for Using User IP Locations

We first evaluated the performance of detecting serving area of web resources from user IP locations. We chose 535 USA governmental sites whose top domains are *.gov* as the test set. The sites were selected so that they have more than 200 clicks in the search log. We manually labeled all the test sites with correct serving areas in advance.

The performance of the algorithms is greatly affected by the number of clicks available in the log. If there are more clicks for a web site, more precise results are expected to be obtained. From Figure 1, we can clearly see that with the increasing of click count, F-measure increases quickly. In addition, the click count has a much bigger impact on recall than precision. On the other hand, when the click count increases, the number of web sites that can be computed by our log drops fast.

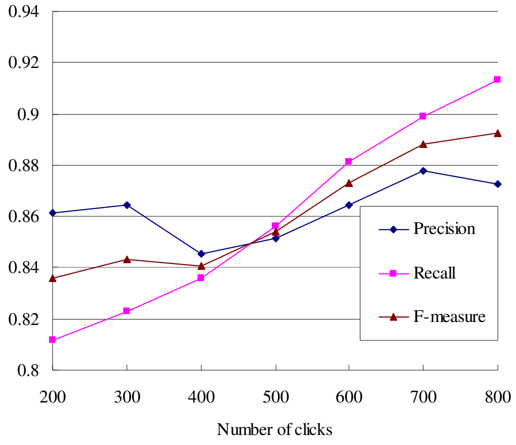

Figure 1. Impact of click count on computing serving area from IP locations.

3.2 Results for Using Query Terms

We use the same test set of 535 USA government sites as in the previous sub-section. According to our log, every log item will have a user IP and a query. For each web site, we do not group identical queries together. Therefore, the number of queries equals to the number of user IPs here.

We define a *Well-Known Degree* (*WKD*) in the algorithm. When *WKD* is 0.05, it means a web site is serving location *l* only if more than 5% of query terms contain *l*. In the following experiments, we fix *WKD=0.12*.

Figure 2 shows the performance of computing serving area from query terms. Comparing Figure 2 and Figure 1, we found that the performance of using query terms is more stable than that of using IP locations. The F-measure of query term based approach is better than that of IP location based approach when the click count is less than 600. The main reason here is that IP locations are usually not very precise. Therefore, in most cases, query term information will be considered superior to user IPs.

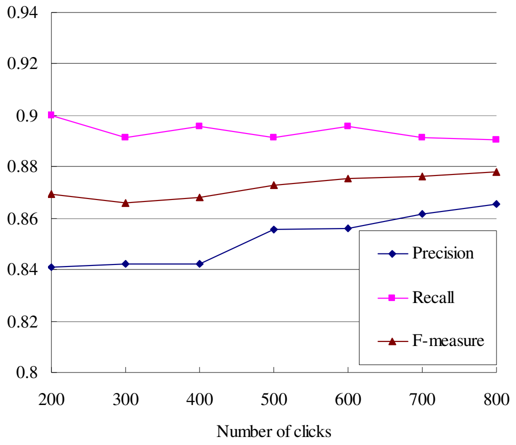

Figure 3. Impact of click count on computing serving area from query terms.

4. CONCLUSIONS

In this paper, we studied the serving area of web resources, which stands for the geographical distribution of their potential users. Knowing the serving area is important to improve the performance of certain web applications such as local search and local advertisement. Experimental results showed that both the algorithms worked well while the query term based algorithm was more effective than the IP location based approach.

5. REFERENCES

[1] Ding, J., Gravano, L., and Shivakumar N. Computing geographical scopes of web resource. *VLDB 2000*.

[2] Buyukkokten, O., Cho, J., Garcia-Molina, H., Gravano, L., and Shivakumar, N. Exploiting geographical location information of web pages. *WebDB'99*.

[3] Amitay, E., Har'El, N., Sivan, R., and Soffer, A. Web-a-where: geotagging web content. *SIGIR 2004*.

[4] Wang C., Xie X., Wang L., etc. Detecting geographic locations from web resources. *GIR 2005*.

[5] MSN Search. http://search.msn.com/