Caching Schema
for Mobile Web Information Retrieval

R. Lee[†], K. Goshima[†], Y. Kambayashi[†] and H. Takakura[‡]
Graduate School of Informatics, Kyoto University[†]
Data Processing Center, Kyoto University[‡]
Sakyo Kyoto 606-8501, Japan
{ryong, gossy, yahiko}@db.soc.i.kyoto-u.ac.jp,
takakura@rd.kudpc.kyoto-u.ac.jp

Abstract

The web cache management in mobile devices becomes to be an important problem. In this paper, we discuss that the traditional cache method LRU(Least Recently Used) is not sufficient in mobile environments. Since the data related to the places already visited by the mobile user may not be used again, *a user behavior in the mobile environments* should be considered. In another aspect of web data uses, we are often connecting to the web with the wired broadband networks. Before traveling with the mobile devices that are limited in the storage space and the power, *carrying out the web data in the mobile cache* will be a good strategy to reduce the efforts in mobile-based web information retrieval. In the pre-fetching of web data, we can improve the cache efficiency significantly by dealing with *metadata* such as selected URL's (with proper keywords) instead of taking out the whole web contents. In this paper, caching algorithms considering web contents, metadata and user behavior history are developed. In order to determine priority of web pages, word relationships obtained from a volume of web pages are used.

1 Introduction

One of the serious bottle-necks of mobile systems is slow communication speed. If required information is stored in the cache of the mobile system, it will be retrieved rapidly. As cache size is limited in mobile systems, we need to develop proper algorithms for such a purpose.

In this paper, we will discuss priority computation algorithms for mobile systems to be used for sight-seeing. Although the application is limited to identify the problems, it will be rather easy to extend the result to other cases.

Compared with traditional CPU cache and web cache, we have to develop different algorithms to determine the priorities of contents in the cache. For example, in the traditional cache, data used recently will have high priority. In mobile applications, if a user visited a place, information related to the place may not be required later. As the cache size is limited we will store URL's instead of web contents if the priority is not very high.

If a user is located at place A, there are some possibility to visit place B, if both are related. Strength of relationships is calculated by the number of appearance of (A,B) pair in one paragraph (or one web page). Usually the pair (A,B) appear frequently if they are located in a short distance. There are, however, cases they are far a part. In such cases these locations are semantically related (for example, having similar nature).

In order to find out relationship among words, we first classified words into G-words(geo-words, related to location names) and N-words(non-geo-words). We have collected 2 million web pages related to Kyoto City in Japan, which are the basis of deriving word relationships.

In Section 2, we will discuss the characteristics of mobile cache algorithms by comparing other typical cache algorithms. Section 3 shows organization of mobile cache for travel guide as example. How to determine priorities of cache contents using word relationships is discussed in Section 4. Consideration of web environments is discussed in Section 5. Section 6 shows some problems found by simulation.

2 Characteristics of Mobile Cache

In this section we will compare traditional CPU cache, web cache, and mobile cache to be discussed in this paper, to identify problems of mobile cache.

2.1 Traditional CPU Cache

Requirements and characteristics for CPU cache are as follows.

1. All the data are equal size.
2. Cache size is very small.
3. Update of data is performed in cache.
4. In many case data are accessed sequentially.

As speed of CPU is very high, we need to use very efficient algorithm for cache replacement. LRU(Least Recently Used) is very popular, since the computational overhead is very low and reasonably good results are obtained[7].

2.2 Web Cache

Unlike CPU, the communication speed is very slow for the case of web cache, thus we can use complicated algorithms for web cache[1, 2].

1. Data sizes distribute from very small to very large(for example, video).
2. We can use disks for cache, so the cache size can be rather large.
3. Update is performed at web pages. Some cache contents will become out-of-dates easily.
4. There are some related pages (shown by links).
5. There are several different usage patterns, which will appear periodically. For example, usage pattern from 9 a.m., just before 5 p.m., and nights may be different.

The most serious problem is size[2, 4]. If we put one large data object in the cache, other data cannot be stored in the cache. We need to consider the size beside the recency treated by LRU. Furthermore, because of the above 5, we need to adjust usage patterns. A simple way is to use popularity (how many times a page is used in 24 hours).

We have developed a complicated web cache algorithm considering size, recency and popularity[3].

2.3 Cache for Mobile Systems

Mobile systems usually have a small amount of cache and data usage patterns is quite different from the above two cases[6, 8].

Usually cache is used to utilize part data. Requirement for mobile systems is quite different. For example, a person visit a famous sight-seeing spot, the map to approach the place may not be used again, although it may be used frequently before. Another example is that all the lunch information will not be utilized after the user takes a lunch.

Since the communication speed is very slow, the cache system have to store data to be used near future, by predicting user's behavior. For example, a person is going to a some place by a car, parking information near the place should be stored in advance.

As a cache size is small, we may have to store URL's instead of contents. To find proper URL's, keywords for each URL are required.

As a summary, requirements and characteristics of mobile cache are as follows.

1. The cache size is rather small.

2. Cache algorithm should use prediction information of the owner's behavior.

    - If the data object is used and it is predicted not to be used again, contents will be erased from the cache. The URL and some meta information (usage information) should be stored in the cache, unless usage is sure to be never occurred again.

    - By predicting future user behavior, we can order the URL's stored in the cache using metadata attached to them. Contents of important URL's are retrieved in advanced, since the communication speed is slow.

3. By the current web page used by the user, we can add URL's from its link information if required.

For traveling purpose, before travel we can store URL's, which may be required during the travel. By actual usage of web pages we have to modify the URL list by the above 3.

3 Mobile Travel Guide

In this section, we will discuss functional requirements for a mobile travel guide support system. Mobile environment has many constraints such as limited storages, slow communication, poor user interfaces and slow operation speed. To overcome these problems, we will discuss generalized cache algorithms for mobile systems. Unlike conventional cache, metadata are also cached and user behavior is reflected to cached contents as discussed in Section 2.

3.1 Cache Contents

As a strategy to use contents in mobile environment, there are the following three patterns.

1. *Storing selected web contents into mobile devices*
   If all necessary web contents are stored in cache, we do not need to use the web. Communication cost will be reduced. Since some web contents may be updated frequently, the user cannot set the latest information of such pages by this method.

2. *Downloading all necessary web information when requested*
   Conventional cellular phones use this method, since they cannot store a large amount of data.

3. *Storing metadata which help retrieval necessary web pages*
   Instead of storing all the web contents, we can store only URL's in order to reduce the storage cost. Web page selection is performed before getting the contents. We can store a large number of URL's for small cache area.

According to the travel plan, we can determine the priority of web pages.

Web pages with the top level priority
Contents of these web pages are actually stored in the cache, if they will not change frequently. Otherwise, only URL's are stored.

Web pages with the second level priority
Only parts of web contents are stored with URL's. These partial contents are used to find proper URL's. Link information may be also stored in cache. For partial contents, we can use frequently appearing keywords and/or the top parts (including the titles) of pages.

Web pages with the third level priority
Only URL's are stored. Figure 1 shows the organization of the system to be

discussed in this paper. According to the user behavior history, priorities of web pages are dynamically modified. We use word relationships to determine the priority.

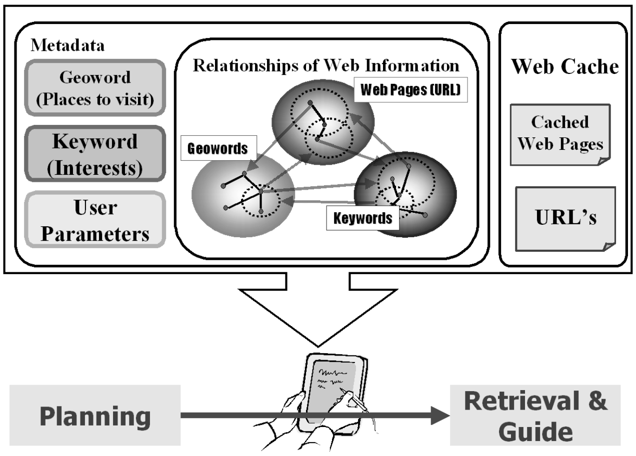

Figure 1: System Organization

3.2 System Functions

The system has functions of a web page cache management and a dynamic travel guide, using metadata such as relationships of geowords or keywords derived from the web, and user environment parameters. Two phases are considered in the system, *planning phase* and *retrieval-and-guide phase*.

Planning phase is performed at home before the departure of a tour. In this phase, the system works to support of surveying destination area by retrieving web pages, decision making of target spots and visiting order, and storing metadata (relations, user status) into the mobile devices.

On the other hand, retrieval-and-guide phase is during the travel in outdoors. This phase works for efficient management of web page cache based on page priority ranking, and active dynamic travel guide by suggestion of closely related web pages to the user.

3.2.1 Planning Phase

In planning phase, we assume that the user can use the wired communication environment with enough bandwidth. This phase consists of two kinds of works. One is to decide visiting places in the target area. The other is to download data to be used offline which will be needed for guide.

First, a user can get knowledge of the places by browsing web pages, and determine his objective spots and visiting order.

Then, required metadata are stored into the mobile device. The required metadata are

- lists of G-words, keywords, and URL's, and their relationships
- G-words of user's destination and keywords of his interests
- initial values of user status (e.g. money conditions, previously visited, etc.)

In addition, web page contents themselves are downloaded to store into web page cache. That reduces access frequency of mobile retrieval.

3.2.2 Retrieval-and-Guide Phase

In retrieval-and-guide phase, two functions are provided:

- web page cache management based on page priority ranking
- active dynamic travel guide and page pre-fetching

Active dynamic travel guide is a function to recommend the user web pages related to the current location, interesting keywords or his other conditions. This function considers users who have not well prepared or unexpected plan changes. Guide selects pages which have high priority in the cache and show the user, as if it says "Why don't you visit this spot written on this page?" In this way, the user can find the next destination suitable to him. We regard this function as a kind of guide agent. Moreover, the speed of retrieval can be improved by pre-fetching the target pages.

4 A Page Priority Ranking Algorithm with Geographical Relationships

In this section, we will discuss a replacement algorithm for web page cache which stores web contents in mobile environment. The algorithm calculates the priority ranking of pages based on user environment and makes low priority pages to replace. Though various factors (parameters) can be used as user environment, we first introduce a cache model based on only location information (geowords), and we will extend to the model using keywords representing user's interests in the next step.

4.1 Word Relationships

From the web pages related to Kyoto, we have derived relationship among words[5]. Words are classified into G-words(geowords, location names) and N-words(non-geo-words). If two words are appeared frequently in one paragraph, or one web page, we assume that there are strong relationships among the two words we use frequency to show the strength of the relationship.

4.2 G-G Model

We first introduce the model using only G-G relations (relations between geowords). We call this model *G-G model* in the later of the paper.

G-G model has a graph structure shown in Figure 2. Nodes in this graph are classified into three groups. The first group in the left side is a page cache P, which is the set of web pages p_i. The second group in the center is a geoword set G, called *index* of P. Each g_j represent regional references of each of the web pages. The third group in the right side is a single node, user's current location.

There exist *weighted G-G relations* between g_U and each g_j. A pair with no G-G relations is treated as a zero-weighted G-G relation.

The purpose of this algorithm is to rank the pages p in cache P in the order of priority

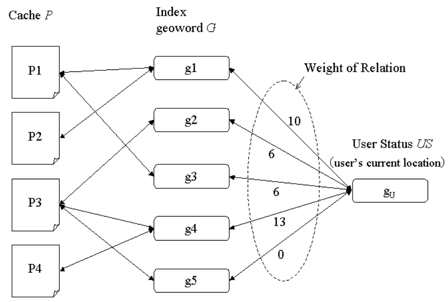

Figure 2: G-G model

for current user and possibility to be accessed in the near future, based on weight values of G-G relations.

4.3 Priority Computation for G-words

To describe the algorithm based on above-mentioned basic policies, we introduce the following definitions.

Definition 1
$Weight(g_i, g_j)$ is a **weight** of G-G relation between geoword g_i and g_j. If there are no G-G relations between g_i and g_j, $Weight(g_i, g_j) = 0$.

Definition 2
Given a geoword g_j related to a page p, a **score** of p by g_j, $Score[g_j](p)$, is a priority value of p determined by g_j. In G-G model, $Score[g_j](p) = Weight(g_j, g_U)$.

Definition 3
Given geowords $g_1, ..., g_n$ related to a page p_i, a **score set** of p_i, $ScoreSet(p_i)$, is a set $\{Score[g_1](p_i), ..., Score[g_n](p_i)\}$.

Using these definitions, the algorithm of page priority ranking in G-G model is described as follows.

1. Calculate $ScoreSet(p_i)$ for each p_i in P. (In each $ScoreSet(p_i)$, scores are sorted in descending order)

2. Sort the pages p_i by their score sets with higher score. (i.e. by descending lexicographic order)

This order is the priority ranking of web pages in P.

4.4 Keywords

Geowords in G-G Model can be equivalently replaced by keywords, which represent interests of the user. Moreover, an integrated model *GK-GK Model* can be introduced as illustrated in Figure 3. By using this extended model, the same algorithm can be applied for calculating page priority to which both current location and a user interest are considered.

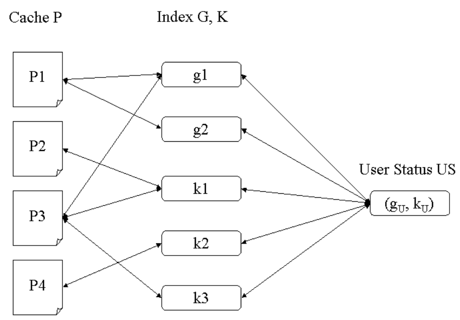

Figure 3: GK-GK Model

5 User Environments

In the previous section, we discussed a cache model and an algorithm using geoword sand keywords as user status. However, furthermore consideration is required in order to judge page priority ranking reflecting users' requests more effectively.

Assume that a user is trying to retrieve and get information about his current or neighborhood location from the web pages in order to his next visiting place. He queries the system the pages related to his current location and interest. But when he checked the returned web page, it is about the location where he visited just now. He will be disappointed because he does not to want information about places which he had visited.

In this case, it is natural that a user hopes to know the places where he has never visited. That is to say, priority of cached web pages will be modified by user's history of visits.

Another example is that pages about downtown which have a lot of restaurant information are likely to be more important at lunch or dinner time. How much money a user has, will change the priority of restaurants, high grade or fast foods.

These examples show that it is important to apply parameters for satisfying priority ranking reflecting various user environments. In this section, we discuss methods to introduce these parameters in addition to relationship to the ranking algorithm.

5.1 Score Control with Parameters

In order to control value of score, we define Score Control Parameter as follows.

Definition 4

Score Control Parameter $PARAM_i$ is expressed by the triplet (V_i, f_i, α_i).

V_i is a set of criterion values for increasing and decreasing scores. Each element of V_i means, for instance, user's condition such as history of visit or money conditions etc., and values of external environments such as time or weather etc. f_i is a function which gives increment of score, which is determined by an index c (either geowords or keywords) and a set of criterion values V_i. α_i is a constant number representing a weight of $PARAM_i$. By the parameters defined in above, following definition can be introduced.

Definition 5

Score by index c of page p, $FixedScore[c](p)$ is

$$\begin{aligned} FixedScore[c](p) &= Score[c](p) \\ &+ \sum_i \alpha_i \cdot f_i(c, V_i) \end{aligned}$$

When executing the above algorithm, $FixedScore[c](p)$ should be used as elements of score sets instead of scores. By adding the value $f(c, V)$ determined by c and V to the score, result ranking of priority can be controlled to reflect user's demand more. α_i is

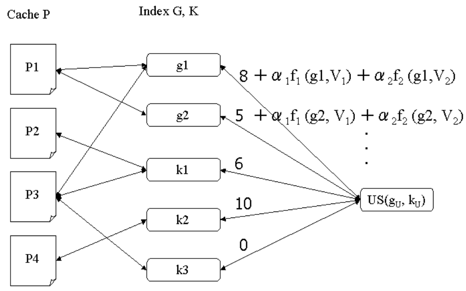

Figure 4: Score Control with User Parameters

a weight value which indicates how strongly each $PARAM_i$ should affect on the scores.

Concrete examples of user parameters for score control are as follows.

Order of Visit For a user at a particular time, information about a place which he has ever visited tend to be less important. Furthermore the later he visited a place, the less importance and motivation to visit.

In order to calculate, geowords sequence of visiting $\{g_1, g_2, \ldots, g_n\}$ and the past visited history $\{g_{-1}, g_{-2}, \ldots, g_{-m}\}$ are added to the set of user status values V. The function f is defined to be maximum at the nearest future, minimum at the latest, and the farther from present the closer to 0. For example, given present time t_0 and the place (geoword) visited at time t be g_t, the incremental value is represented as following formula:

$$f(g_t, V) = \frac{1}{e^{t-t_0}}$$

Temporal Changes of Priority Geowords of downtown, such as Kawaramachi and Gion for instance, are likely to have higher importance at lunch or dinner time because users wants to have a meal at that time. That fact has few concerns with relations between regions. Moreover, touristic facilities (e.g. museums or zoos) usually have business hours, therefore we must consider whether they are available or not now. Temporal events such as festivals are also to be applied this case.

In order to calculate temporal priority changes, present time t should be set as an element of V. To derive the function f, an appropriate database which provides a correspondences table between geoword and available time period is prepared.

Money Conditions It is expected that electrical payment using mobile devices will be more popular in the near future. In such environments, since users' money can be managed with their mobile equipments, that is also possible to be used for parameters of the page priority ranking algorithm. One of its advantages is, for example, a guide for rich users to area which has many luxury shops or facilities for which expensive entrance fee is required. Instead, users without much money can be lead to reasonable shops or spots.

For this purpose, the user's available money m is set as V. Incremental function f is prepared a database providing correspondences between geowords and money ranges.

6 Simulation

We had made simulations of ranking web pages in order to evaluate our algorithm.

First, random 100, 500, and 1000 pages (URL's) related to Kyoto City are prepared as target web page sets to rank. They are extracted from our web page collection which have been gathered for our developing regional search engine. Each of the pages has its related geowords.

Next, assumed current locations of the mobile user are determined. we chose two geowords *Ginkakuji* and *Gion* at random, which are names of regions in Kyoto.

Then these page sets, their related geowords, and user's current locations are adapted to our page priority ranking algorithm. The most basic model G-G model (considering only relations of geowords) is used. Six trials are done by the combination of three page sets and two user locations.

As a result, we succeeded to place pages about current locations or neighborhood to the upper ranking. However, an important tendency was found. In every trial of ranking, similar pages are found with high priority, for example,

- `www.momonga.org/kyoto/spot-j.html` (lists of attractive spots with beautiful flowers in Kyoto)
- `gourmet.yahoo.co.jp/gourmet/restaurant/Kinki/Kyoto/list/area_genre/260005_0207` (lists of restaurants in Kyoto)
- `www.kyoto-np.co.jp/kp/topics/2000jun/bk_index.html` (news back numbers in Kyoto)

Common characteristics of these pages are that they includes many kinds of geowords in Kyoto. We should exclude such pages to get better results.

In order to avoid such cases, we must consider more semantics and meanings of pages, such as geographical coverage/popularity or degree of details.

7 Conclusion

In this paper, we proposed a web page cache management method in mobile environments.

An application system for travel guide support using the cache management methods is suggested. Metadata are stored in mobile devices at planning phase, and web retrieval during travel is supported by efficient cache management at retrieval-and-guide phase. The system provides another function, an active and dynamic travel guide.

A priority ranking algorithm for web pages is suggested for the system. It uses relationships between users' current locations and geographical information of web pages. This algorithm can be extended into using keywords of interests, or other various user parameters such as history of visit, time, money conditions, and so on.

As a result of some simulations, a problem is revealed that pages which have many kinds of geowords tend to ranked higher. In order to find variable pages without many geowords but have rich information, we must consider more semantics and meanings of pages, such as geographical coverage/popularity or degree of details. That is one of our future work.

Acknowledgment

This work has been supported by 'Universal Design in Digital City' Project in CREST of JST (Japan Science and Technology Corporation).

References

[1] M. Abrams, C. R. Standridge, G. Abdulla, S. Wiliams, and E. A. Fox, "Caching Proxies: Limitations and Potentials," Proc. of the 4th International WWW Conference, 1995.

[2] C. Aggarwal, J.L. Wolf, and P.S. Yu, "Caching on the World Wide Web," IEEE Transactions on knowledge and data engineering, 11(1), 1999.

[3] K. Cheng, Y. Kambayashi, "LRU-SP: A size adjustable popularity-aware LRU replacement algorithm for Web Caching," IEEE COMPSAC, IEEE CS Press, pp. 48-53, 2000.

[4] R. Karedla, J. S. Love, and B. G. Wheery, "Caching Strategies to Improve Disk System Performance," IEEE Computer, 27(3):38-46, 1994.

[5] R. Lee, H. Takakura, and Y. Kambayashi, "Visual Query Processing for GIS with Web Contents," Proc. of the 6th IFIP Working Conference on Visual Database Systems, May 2002. (to appear)

[6] Q. Ren, M.H. Dunham, "Using Semantic Caching to Manage Location Dependent Data in Mobile Computing," The 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00), August 2000.

[7] A. Silberschatz and P. B. Galvin, "Operating Systems Concepts," Addison Wesley, Reading, MA, fourth edition, 1994.

[8] B. Zheng, D. L. Lee, "Semantic Caching in Location-Dependent Query Processing," The 7th International Symposium on Spatial and Temporal Databases, Springer-Verlag, LNCS 2121, pp. 97-113, 2001.