ALTERNATIVE QUERY DISCOVERY FROM THE WEB FOR DAILY MOBILE DECISION SUPPORT

Shun Hattori
School of Computer Science - Tokyo University of Technology - 1404-1 Katakura-cho, Hachioji - Tokyo 192-0982 JAPAN

ABSTRACT

We daily make decisions on whether or not we should do an activity (behavior) and further on whether or not we had better do its alternative activity. If we can acquire the necessary information for making such decisions situation by situation in mobile/ubiquitous computing environments, we would be able to make better decisions and might be happier. So, this paper proposes a Mobile DSS (Decision Support System) that discovers the alternative activities to a mobile user's intending (original) activity from the Web by using text mining techniques and then automatically generates her/his activity-based query to the existing location-free information retrieval systems such as mobile Web search engines.

KEYWORDS

Mobile Decision Support System (MDSS), Mobile Web Search, Location-Based System, Activity-based Querying, Real-world Context-aware Querying, Alternative Query, Web Mining, Text Mining.

1. INTRODUCTION

In our daily lives, we continuously make an absolute decision on whether or not we should do an activity (behavior), and further make a relative decision on whether or not we have a better alternative activity to do than the original activity. If we can acquire the necessary information for making such a decision situation by situation in mobile/ubiquitous computing environments, we would be able to make a better decision.

In recent years, the amount of information available on the Web has been growing exponentially and also mobile computing environments have been improved and maintained. Therefore, the Web might be now the best source of information for daily mobile decision-making. However, the Web would not be useful in practice if we cannot filter the noisy information for making such a decision and collect only the necessary information for making the decision as soon and easily as possible, because we have to make the decision more quickly in mobile computing environments than in immobile ones at home or office.

This paper proposes a novel Mobile DSS (Decision Support System) that allows a mobile user to search the Web for the necessary information for making a decision on whether or not s/he should perform her/his intending activity. When a mobile user or AI robot firstly gives an activity expression, the system discovers its alternative activity expressions and offers them to the mobile user or AI robot's database. When s/he secondly selects one of them, the system generates the activity-based query and searches the Web by submitting it to the existing location-free information retrieval systems such as mobile Web search engines.

The remainder of this paper is organized as follows. Section 2 introduces expressions of activities (behaviors) in my proposed Mobile DSS. Section 3 gives an overview of my proposed system. Section 4 describes my method to discover typical activity expressions for a given place-name (i.e., typical activities at a given place) from the Web by using text mining techniques. Section 5 explains my method to discover alternative activity expressions to a mobile user's intending (original) activity from the Web. Section 6 presents related works to this paper. In Section 7, I conclude this paper and mention future works.

2. EXPRESSION OF ACTIVITY IN MOBILE DSS

This section introduces expressions of activities in my proposed Mobile DSS (Decision Support System). The system helps mobile users or AI robots to search the Web for the necessary information for making their decision on whether or not they should do an activity (behavior). Therefore, the system has to receive what their intending activity is in a specific expression from the mobile users.

A mobile user's intending (or pre-done or now-doing) activity on which s/he requests information would be specified by the following five components:

1. **Action**: what action (verb) does s/he perform?
   – e.g., "buy", "read", "enjoy", "play", "eat" or "drink".
2. **Object**: what object does s/he perform the action on?
   – e.g., names of class to which it belongs such as "book" or "novel", names of its concrete instance such as "Harry Potter" or "Lord of the Rings".
3. **Place**: where does s/he perform the action on the object?
   – e.g., names of class to which it belongs such as "bookstore", names of its concrete instance such as "Barnes & Noble" or "Amazon.com".
4. **Time**: when does s/he perform the action on the object in the place?
   – e.g., date, time of day, time intervals such as "morning" or "night".
5. **Subject**: who is s/he?
   – e.g., sexuality such as "boy" or "girl", ages such as "young" or "elder".

However, we mainly have the following two types of retrieving information for making our daily decision in mobile computing environments:

– Search for **Object** and information about it

We want to know what object we can perform our intending action on here and also information for going there such as map of surrounding area or route guidance. For example, when a mobile user intends to buy a "Harry Potter" book here, s/he would be thankful to know the following information:

1. Whether or not should s/he buy the book here?
2a. Whether or not does there exist a better book than the book?
2b. Whether or not does there exist a better class (category) of goods named by "Harry Potter" than "book"?
3a. What is the alternative book? (e.g., "Lord of the Rings")
3b. What is the alternative class? (e.g., "movie", "dvd" or "game")

– Search for **Place** and information about it

We want to know where we can do our intending activity and also information for going there such as map of surrounding area or route guidance. For example, when a mobile user intends to buy a book at a certain bookstore, s/he would be glad to know the following information:

1. Whether or not should s/he go to the bookstore to buy a book?
2. Whether or not does there exist a better place than the bookstore?
3. Where is the alternative bookstore?

Therefore, in this paper, my proposed Mobile DSS expresses an activity as a set of a verb of **Action** and three nouns of **Object Class**, **Object Instance**, and **Place**. For example, an activity to buy a "Harry Potter" book at a bookstore is expressed by

[a:"buy" oc:"book" oi:"Harry Potter" p:"bookstore"].

3. SYSTEM OVERVIEW

Figure 1 gives an overview of my proposed system for daily mobile decision support, which consists of the following three steps:

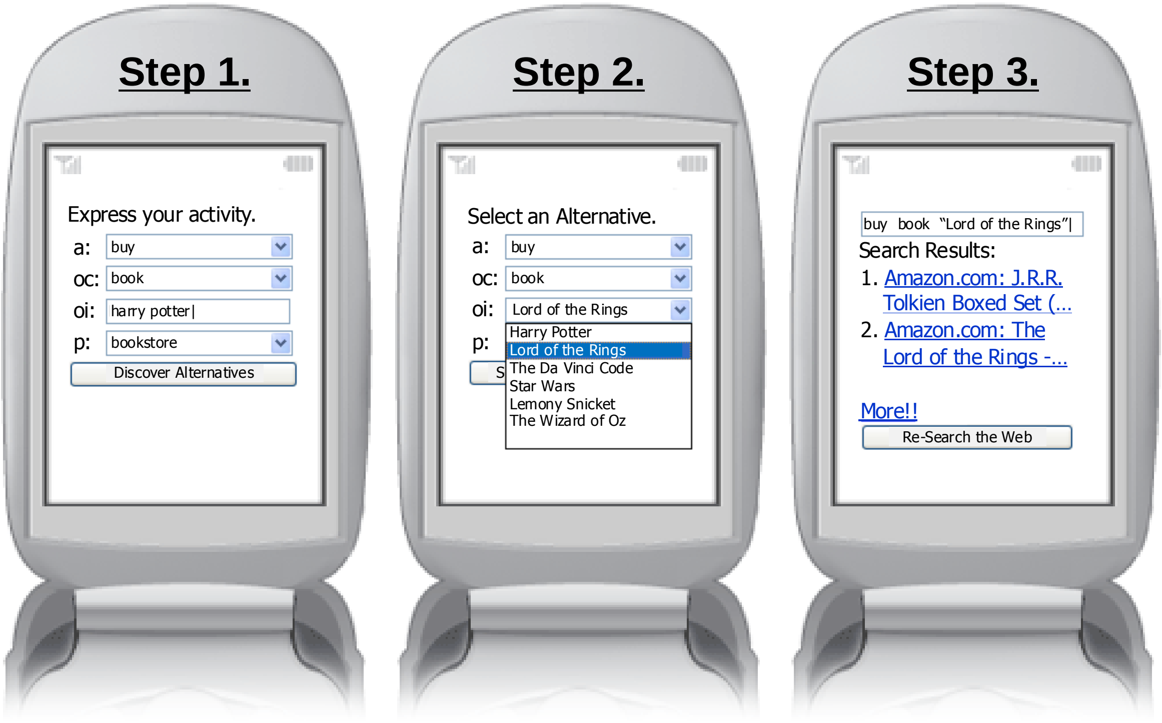
Step 1. Discovering Typical Activities at Current Place

When a mobile user accesses the Mobile DSS via her/his mobile device at a place, the system automatically discovers the typical activity expressions of her/his current place by mining the Web/Blog as described in detail in Section 4, and then offers them to her/him in order to allow her/him to use them as reference to specify an expression of her/his intending activity. For example, "bookstore" → [a:"buy" oc:"book" p:"bookstore"].

Step 2. Discovering Alternative Activities to Original Activity
When s/he gives an expression of her/his intending (original) activity while referring the typical activities of the current place, the system discovers expressions of its alternative activities by my method described in detail in Section 5, and then offers them to her/him in order to allowing her/him to open up the alternatives from her/his original intending activity. For example,

[a:"buy" oc:"book" oi:"Harry Potter" p:"Waldenbooks"]
→ {
[a:"buy" oc:"cd" oi:"Harry Potter" p:"Waldenbooks"],
[a:"buy" oc:"book" oi:"Lord of the Rings" p:"Waldenbooks"],
[a:"buy" oc:"book" oi:"Harry Potter" p:"Barnes & Noble"].

Step 3. Generating Activity-based Query
When s/he selects one of the alternative activity expressions offered in Step 2, the system generates the activity-based query and searches the Web by submitting it to the existing location-free Web search engines such as [Google Mobile, 2011].



Step 1.

Express your activity.
a: buy
oc: book
oi: harry potter|
p: bookstore

Step 2.

Select an Alternative.
a: buy
oc: book
oi: Lord of the Rings
p:

Step 3.

buy book "Lord of the Rings"|
Search Results:
1. Amazon.com: J.R.R. Tolkien Boxed Set (...
2. Amazon.com: The Lord of the Rings -...

Figure 1. An Overview of My Proposed Daily Mobile DSS (Decision Support System)

4. DISCOVERING TYPICAL ACTIVITIES OF PLACE FROM THE WEB

This section describes my method to discover the typical activities that people often do at a given place by mining enormous text documents from the Web/Blog, which consists of the following two steps:

Step 1. Extracting Candidates for Typical Activities from the Web
When a name of class or instance of place *p* is given (e.g., "bookstore" or "Barnes & Noble"), my proposed Mobile DSS (Decision Support System) firstly collects sentences from the Web/Blog, each of which contains the expression "at ... *p*" and whose subject is either "I" or "we", by submitting ["at * *p*"] as a wildcard (*) query to [Google Blog Search, 2011].

Next, the system extracts a set of a verb and its objective noun (if any) from each sentence
and regards the set as a candidate for typical activities at the place named by *p*.

Step 2. Weighting Candidates for Typical Activities

Secondly, the system assigns the following weight, to each candidate which is a set *s* of a verb *v* and its objective noun *n* (if any) for the given place-name *p*:

$$\mathrm{weight}_p(s) := \frac{\mathrm{df}(q_p^{\mathrm{at}})^2}{\mathrm{df}(q^{\mathrm{at}}) \cdot \log_2 \mathrm{df}(q^{\mathrm{at}})}$$

$$\mathrm{activity}(s) := \begin{cases} "v_{\mathrm{ed}}" & \text{if n is null} \\ "v_{\mathrm{ed}} * \mathrm{n}" & \text{otherwise.} \end{cases}$$

$$q^{\mathrm{at}} := ["\mathrm{activity}(s)\ \mathrm{at}"]$$

$$q_p^{\mathrm{at}} := ["\mathrm{activity}(s)\ \mathrm{at} * p"]$$

where df(*q*) stands for the number of documents retrieved by submitting a query *q* to Google Blog Search and v_{ed} stands for the preterit of a verb *v*.

Table 1 shows the discovered typical activities of a place-name “bookstore”. The candidates with their weight than a certain threshold value (e.g., 0.01) are accepted as the typical activity expressions for the place.

Table 1. Discovering the Typical Activities of p:“bookstore”

Candidate *s*		activity(*s*)	$\mathrm{df}(q_p^{\mathrm{at}})$	$\mathrm{df}(q^{\mathrm{at}})$	$\mathrm{weight}_p(s)$
v:“buy”,	n:“book”	“bought * book”	64	1507	0.257446
v:“see”,	n:“book”	“saw * book”	26	572	0.129021
v:“pick up”,	n:“book”	“picked up * book”	25	1095	0.056530
v:“work”,	n:null	“worked”	437	275477	0.038360
v:“find”,	n:“book”	“found * book”	15	855	0.027018
v:“get”,	n:“book”	“got * book”	8	673	0.010122
v:“meet”,	n:null	“met”	139	224058	0.004851
v:“spend”,	n:“hour”	“spent * hour”	6	2102	0.001551
v:“drink”,	n:“coffee”	“drank * coffee”	1	192	0.000686
v:“meet”,	n:“girl”	“met * girl”	4	3501	0.000388

5. DISCOVERING ALTERNATIVE ACTIVITIES FROM THE WEB

A mobile user or AI robot sometimes searches the Web for making her/his decision on whether or not s/he should do an activity (behavior). However, s/he misses better activities than her/his original intending activity, because s/he is usually unaware of the alternative activities to her/his original intending activity.

[Google Sets, 2011] is a web-based tool for discovering comparable phrases of a user-given objective phrase. But we cannot condition the object by one of its various contexts. Meanwhile, [Ohshima, 2006] proposed a method for allowing us to do so. Their approach is based on the following two assumptions:

(1) the conjunction “or” connects two comparable phrases,

(2) if two phrases p_1 and p_2 are comparable, there exist both “p_1 or p_2” and “p_2 or p_1” pattern
in the target corpus of documents such as the Web.

This section describes my method to discover the alternative activities to a given activity by mining the Web/Blog, which consists of the following two steps:

Step 1. Extracting Candidates for Alternative Activities

When an activity expression which is a set of components such as Action, Object and Place is given, my proposed Mobile DSS (Decision Support System) discovers the alternative phrases to each component *p* where the other components are regarded as its context *c*. Firstly, the system retrieves the top *k* retrieval items, each of which includes its title and snippet (brief summary), by submitting

[“p or” AND c] as a query to [Google Web Search, 2011] where c is a keyword-based Boolean expression which concatenates the other components except p by AND operators. Next, a text which is between “p or” and its immediate following separator such as “.”, “,”, “?”, “)” and “or” in the top k retrieval sets of title and snippet and whose number of words is less than certain threshold value, is regarded as a candidate for alternative phrases to the component p in the context c.

Step 2. Weighting of Candidates for Alternative Activities

Secondly, the system assigns the following weight, to each candidate p_i:

$$\mathrm{weight}_{p,c}(p_i) \coloneqq \mathrm{min}_{p,c}(p_i) + \left(1 - \frac{\mathrm{min}_{p,c}(p_i)}{\mathrm{max}_{p,c}(p_i)}\right)$$

$$\mathrm{min}_{p,c}(p_i) \coloneqq \min\left\{\mathrm{df}^{\mathrm{A}}_{p,c}(p_i), \mathrm{df}^{\mathrm{B}}_{p,c}(p_i)\right\}$$

$$\mathrm{max}_{p,c}(p_i) \coloneqq \min\left\{\mathrm{df}^{\mathrm{A}}_{p,c}(p_i), \mathrm{df}^{\mathrm{B}}_{p,c}(p_i)\right\}$$

$$\mathrm{df}^{\mathrm{A}}_{p,c}(p_i) \coloneqq \mathrm{df}([\text{"}p \text{ or } p_i\text{" AND } c])$$

$$\mathrm{df}^{\mathrm{B}}_{p,c}(p_i) \coloneqq \mathrm{df}([\text{"}p_i \text{ or } p\text{" AND } c])$$

where df(q) stands for the number of documents retrieved by submitting a query q to Google Web Search, and weight$_{p,c}(p_i)$ = 0 if min$_{p,c}(p_i)$ = 0. Finally, a new set of a candidate with higher weight and the other components is regarded as an expression of alternative activity to the given activity.

Table 2 shows the discovered alternative phrases of an objective phrase “Harry Potter” in various contexts. The candidates with their weight than a certain threshold value (e.g., 1.00) are accepted as the alternative activity expressions to the original activity.

When a mobile user or AI robot at a “bookstore” is intending to “buy” a “book” named by “Harry Potter” and thus inputs her/his intending activity expression,

[a:“buy” oc:“book” oi:“Harry Potter” p:“bookstore”],

to my proposed Mobile DSS, the system offers the following alternative activity expressions to her/him.

1. [a:“buy” oc:“book” oi:“Lord of the Rings” p:“bookstore”]
2. [a:“buy” oc:“book” oi:“Star Wars” p:“bookstore”]
3. [a:“buy” oc:“book” oi:“Da Vinci Code” p:“bookstore”]

Meanwhile, when s/he at a “theater” is intending to “buy” a “movie” named by “Harry Potter” and thus inputs her/his intending activity expression,

[a:“buy” oc:“movie” oi:“Harry Potter” p:“theater”],

to my proposed Mobile DSS, the system offers the following alternative activity expressions to her/him.

1. [a:“buy” oc:“movie” oi:“Lord of the Rings” p:“theater”]
2. [a:“buy” oc:“movie” oi:“Star Wars” p:“theater”]

Table 2. Alternative Phrases of the Object Named by “Harry Potter”

Context c = [“buy” AND “book” AND “Waldenbooks”]			
Candidate p_i	$\mathrm{df}^{\mathrm{A}}_{p,c}(p_i)$	$\mathrm{df}^{\mathrm{B}}_{p,c}(p_i)$	$\mathrm{weight}_{p,c}(p_i)$
“LOTR”	1	1	1.00
“Lord of the Rings”	2	0	0.00
“latest issue of Guns and Ammo”	2	0	0.00
“very popular items”	2	0	0.00
“anything”	2	0	0.00
“J.K. Rowlings”	1	0	0.00
“The Da Vinci Code”	1	0	0.00

Context c = [“buy” AND “book” AND “bookstore”]			
Candidate p_i	$\mathrm{df}^{\mathrm{A}}_{p,c}(p_i)$	$\mathrm{df}^{\mathrm{B}}_{p,c}(p_i)$	$\mathrm{weight}_{p,c}(p_i)$
“(The) Lord of the Rings”	(7+) 31	39	38.03
“Star Wars”	11	15	11.27
“(The) Da Vinci Code”	(11+) 5	10	10.38
“Lemony Snicket”	12	4	4.67
“Artemis Fowl”	4	5	4.20
“The Wizard of Oz”	5	2	2.60
“John Grisham”	2	3	2.33
“Nancy Drew”	22	1	1.95
“Dan Brown”	4	1	1.75
“none HP”	13	0	0.00
“Pokemon”	7	0	0.00
“Hogwart’s witches and wizards”	2	0	0.00

Context c = [“buy” AND “movie” AND “theater”]			
Candidate p_i	$\mathrm{df}^{\mathrm{A}}_{p,c}(p_i)$	$\mathrm{df}^{\mathrm{B}}_{p,c}(p_i)$	$\mathrm{weight}_{p,c}(p_i)$
“(The) Lord of the Rings”	(8+) 105	80	80.29
“Star Wars”	30	37	30.19
“James Bond”	5	1	1.80
“Frodo”	3	1	1.67
“(The) Chronicles of Narnia”	(2+) 0	1	1.50
“Mission: Impossible”	2	0	0.00
“ROTK”	2	0	0.00
“White Chicks”	1	0	0.00

6. RELATED WORK

6.1 Mobile Decision Support

Decision Support Systems (DSS) are generally a class of computer-based information systems or knowledge-based systems which support decision making activities in very different ways. With the recent advances in wireless and mobile computing environments, Mobile DSS have been also needed and thus developed for more urgent decision-making situations, such as Mobile Recommendation [Heijden, 2005] for consumers’ making decisions on purchasing in retail stores. Especially, various systems for mobile decision support in healthcare have been proposed [Fortier, 2003; Tsumoto, 2005]. For example, iTriage [Pedro, 2005; Padmanabhan, 2006] can help a clinician to identify the urgency of medical intervention when her/his patient presents with ambiguous triage case, based on a heuristic approach which selects the best triage category, identifies corresponding discriminating attribute of the patient and allows her/him to attach a level of confidence in the decision.

6.2 Alternative Query Generation

Query Refinement systems offer the Alternative Queries to a user's original query for solving the so-called "mismatched query problem" in Information Retrieval that her/his information demand does not match the results retrieved by her/his original query which s/he has composed in order to express it to an IR system.
For instance, Query Expansion [Xu, 1996; Cui, 2002] by inserting another retrieval condition for disambiguating a user's original query, Query Relaxation [Jones, 2003; Mirzadeh, 2004] by deleting a part of its conditions for relaxing it, Query Substitutions [Terra, 2004; Jones, 2006] by replacing a part of its conditions with another condition for shifting it, or Query Recommendation [Baeza-Yates, 2004; Zhang, 2006] for offering its related (nearby-surrounding) queries to her/him. Meanwhile, Query Optimization systems transform a given query into its equivalent Alternative Queries for selecting the optimum query from among them based on their execution costs.

6.3 My Previous Work

My previous work [Hattori, 2006] has proposed a method to expand a mobile user's original query by its related contextual words such as names of Action, Object or Place, based on the typical activities at her/his current geographic location in the real world, for enhancing context-awareness to the existing location-free IR (Information Retrieval) systems such as [Google Mobile, 2011]. This paper has proposed a Mobile DSS which discovers the alternative activities to a mobile user's intending one from the Web and then automatically generates her/his activity-based query to the existing location-free IR systems, for supporting her/his decision-making process in her/his daily life.

7. CONCLUSION

This paper has proposed a Mobile DSS (Decision Support System) that allows a mobile user or AI robot to search the Web for the necessary information for making decision on her/his intending activity. When a mobile user or AI robot at a place firstly gives an activity expression while using the typical activities of the place as reference, my proposed system discovers its alternative activity expressions and offers them to the mobile user or AI robot's database. When s/he secondly selects one of them, my system generates the activity-based query and searches the Web by submitting it to the existing location-free information retrieval systems such as mobile Web search engines. The system would allow a mobile user or AI robot to select a better activity from among the alternative activities to her/his original intending activity.

In the future, I plan to develop and evaluate a prototype system based on my proposed methods in more detail. And moreover, I challenge to invent a method for discovering the alternative activities dependent on not only Action, Object and Place but also Time or Subject, that is Temporized or Personalized.

REFERENCES

Google Mobile. 2011. http://www.google.com/mobile/

Google Blog Search. 2011. http://blogsearch.google.com/

Google Sets. 2011. http://labs.google.com/sets

Google Web Search. 2011. http://www.google.com/

Conference paper or contributed volume

Baeza-Yates, R. et al, 2004. Query Recommendation using Query Logs in Search Engines. *Proceedings of the International Workshop on Clustering Information Over the Web (ClustWeb)*. LNCS Vol.3268, pp. 588-596.

Cui, H. et al, 2002. Probabilistic Query Expansion Using Query Logs. *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*. pp. 325-332.

Fortier, P. et al, 2003. Development of a Hand-held Real-time Decision Support Aid for Critical Care Nursing. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) –Track 6*. pp. 160b.

van der Heijden, H. et al, 2005. Mobile Recommendation Systems for Decision Making “On the Go”. *Proceedings of the International Conference on Mobile Business (ICMB’05)*. pp. 137-143.

Jones, R. et al, 2003. Query Word Deletion Prediction. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’03)*. pp. 435-436.

Jones, R. et al, 2006. Generating Query Substitutions. *Proceedings of the 15th International Conference on World Wide Web (WWW’06)*. pp. 387-396.

Mirzadeh, N. et al, 2004. Supporting User Query Relaxation in a Recommender System. *Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web’04)*. LNCS Vol.3182, pp. 31-40.

Ohshima, H. et al, 2006. Searching Coordinate Terms with Their Context from the Web. *Proceedings of the 7th International Conference on Web Information Systems Engineering (WISE’06)*. LNCS Vol.4255, pp. 40-47.

Padmanabhan, N. et al, 2006. A Mobile Emergency Triage Decision Support System Evaluation.

Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06) - Track 5. pp. 96b.

Pedro, J.S. et al, 2005. On Development and Evaluation of Prototype Mobile Decision Support for Hospital Triage. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS’05) - Track 6*. pp. 157c.

Terra, E. et al, 2004. Scoring Missing Terms in Information Retrieval Tasks. *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM’04)*. pp. 50-58.

Tsumoto, S. et al, 2005. Clinical Decision Support Based on Mobile Telecommunication Systems. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI’05)*. pp. 700-703.

Xu, J. et al, 1996. Query Expansion Using Local and Global Document Analysis. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’96)*. pp. 4-11.

Zhang, Z. et al, 2006. Mining Search Engine Query Logs for Query Recommendation. *Proceedings of the 15th International Conference on World Wide Web (WWW’06)*. pp. 1039-1040.

Hattori, S. et al, 2006. Activity-Based Query Refinement for Context-Aware Information Retrieval. *Proceedings of the 9th International Conference on Asia Digital Libraries (ICADL’06)*. LNCS Vol.4312, pp. 474-477.