Searcher's Activity in Standalone and Web Applications as a Source for Search Query Expansion

Lubomir Vnenk and Maria Bielikova

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology

Ilkovicova 2, 842 16 Bratislava, Slovakia

Lubomir.Vnenk@zoho.com, maria.bielikova@stuba.sk

***Abstract*—The Web has so much variable information, therefore searching a specific one is complex task. To find something valuable, specifying good search query is crucial. However, average search query consists of only about two words which cannot specify intent of a searcher well. We suppose that these words follow in most cases the searcher's activity. Based on this assumption we propose an approach for search query extension by the searcher activity context. The activity context contains words reflecting the searcher's activity context expressed by keywords gathered from the content of one of the very recent activities within applications provided by the searcher. We present an approach for finding out searcher's activity context by analyzing the content and his interaction between applications considering both standalone applications and applications running inside a browser. Our method uses the activity logs to find a connection between the search query and specific application. Then we extend the query by terms gathered by analyzing the selected application content. We evaluate our approach in a series of experiments based on data gathered by monitoring a group of searchers by means of developed logger prototype.**

Keywords—search query expansion; search context; activity context; activity logging; personalized search

I. Introduction

At the age of information overflow it is not easy to find exactly what we are searching for. Even if modern search engines try their best to choose the most valuable search results, they do not have enough information to show only desirable results. It is mostly because of simplicity of a search query. Searcher does not think much about making the search query specific and well-formed at the first attempt. An average search query consists of 2-3 keywords [1] that may exactly describe searcher's interest, but it may be not enough. The query may have several meanings because words themselves are ambiguous by nature and may have many meanings. Ambiguous query may be also because of phrases or names of specific thing, when just splitting these words leads to wrong meaning of the query.

It is up to a search engine to choose which meaning it is going to search. The search engine however is not able to choose right meaning, because it is not able to know which one searcher has intended due the lack of information of searcher's actual context.

If the search is successful and the searcher finds the information he was looking for, it usually will not take longer than 5 minutes. However, if he is not successful, it can take more than 10 minutes to give up searching [2]. Moreover, the longer actual session lasts the more frustrated searcher will become [3]. Our goal is to make a search query more specific to find requested information faster to save searcher's time and to not become frustrated.

To find out right meaning we need to know more about the query and the searcher. Searcher's query must reflect searcher's past actions and his knowledge. He is always searching information about a problem he has met, so if a searcher never heard about the animal Puma he cannot be searching for the animal with a query that contains the word Puma. But if he was recently reading about clothes and trainers, we can assume that he is looking for information about the brand Puma. All information about searchers' past actions and theirs knowledge create their context. Context is that which adds a meaning to the activity. A corollary is that there is no context outside of, or without, an activity [4]. Therefore we can assume that a searcher's query is connected to the searcher's context and the query has arisen from his context, so we should be able to find intended meaning of the query by examining the searcher's context.

However, there is no way to capture whole searcher's context just using information gathered in a web browser. But there is a way to capture searcher's context while he is working with computer. We aim in this paper on capturing searcher's context this way. However, searcher's context still includes lots of information about his state and state of his environment, e.g., weather, mood. We aim at analyzing his activities while performing computing tasks, i.e. we are evaluating his computing activity context. We analyze this context to find out right meaning of a searcher's query. If we find out right and unique meaning of the query we will be able to extend query by words describing the meaning to make query unique. The more unique and specific query is the more precise and valid results searcher will get.

Web personalization aims at customization of a Web site to the needs of particular user or a set of users considering information on users' behavior and their individual interests, in combination with the content of the Web site [5]. By analyzing searcher's activity context and extending a search query by the context, the search becomes personalized. It means that the searchers with different contexts will get different search results. Searcher will also get different results if his context has changed.

II. Related work

Nowadays personalizing web search is a need. There are many attempts on personalizing web search results with aim of achieving more accurate search results for each searcher individually. Search engines themselves try to personalize search results. There exist also many other research projects aiming at modifying search results. To personalize web search we can use existing context-aware strategies. We aim in this paper on activity based strategies. Popular method is evaluating click-through data to infer searcher's preferences.

Based on the research of search engine searchers' browsing behavior, Joachims et al. [6] developed a framework that analyzed click-through data to get information about searcher's preferences on documents and then learned to adapt the ranking function. They find out that relevance of the results influences searcher's clicking decision but it is biased by the order they are presented to searcher.

Leung and Lee [7] developed several profiling methods that consider not just searcher's positive preferences but also the negative ones. By considering each search query individually, methods capture the searcher's preference with fine granularity and achieved better performance than those based on document preferences. Experiment showed that best results had given mix of positive and negative searcher preferences. They could also separate similar and dissimilar queries into clusters thanks to negative preferences.

Based on AOL log analysis (the log that contains search engine captured information about searchers' queries and their click-through) Jiang et al. [8] developed a method to capture searcher's preferences with high accuracy. They also propose very accurate query session segmentation method that consists of choosing right time threshold and finding right type of change in consequent query. They designed specific strategy of evaluating click-through data for 5 types of query reformulation.

Contextual data model can be built from different sources. Various sources are analyzed Kramár in [9]. His model consists of temporal context in form of behavioral search patterns, activity-based context in form of past queries and social context in form of searcher similarity. Each source of contextual data has potential to increase precision of context model based on log analysis of searcher's activity in a web browser [10]. Proposed search context model captures the lightweight semantics of the documents and evaluates implicit feedback to get document relevance.

All analyzed approaches evaluate searcher's activity context only in a browser environment, which represents searcher's activity just partially. To our best knowledge there is no approach analyzing searcher's activity outside a web browser. However, the searcher's activity outside the browser may be significantly important and have even high influence on actual search context (in dependence on particular activity and related applications). Our contribution to the field of search personalization is in considering and evaluating searcher's activity context also in standalone applications outside a web browser.

III. Method for activity context-aware search

Activity context is created dynamically. It is monitored and updated continuously. It is known that each previous social interaction and usage of various tools and utilities for everyday use affects somehow our actual activities [11]. Also our actual activities influence our next activities, so we need to capture each actual activity to be able to model actual activity context. However, it is not possible to capture each user's activity to have exact activity context.

We aim at modeling user's search activity context so important is a context at the moment of the user's search. We hypothesize that the searcher's activity context is related to an activity the searcher performed within an application he has been using recently. It means, the searcher's activity inside all the applications should be captured and analyzed.

We proposed a method which extends search query by few the most relevant keywords from the search activity context. Search activity context is get by analyzing context of an application related to the query, because context of this application is the same as, or very similar to, search context. Therefore capturing and modeling actual activity context for each application is essential and also finding an application related to the query with similar context is an important task.

Our method consists of the following steps:

1. Modeling a searcher's activity context
2. Finding related application to the query
3. Extending the query by proper keywords from the activity context

A. *Modeling activity context*

Activity context is built from the activities inside searcher's applications. However, each application has its own purpose so each one has different activity context and we need to model specific activity context for each application. Application context is modeled based on searcher's activities done in the application considering also the contents of these activities.

To capture searcher's actions we proposed and developed an activity logger aimed at several selected activity types that we considered as the most influential for our evaluating scenario – novice researcher searching relevant research resources [12]. To log the most important sections of searcher's activity we designed and realized an activity logger tool that consists of three independent parts each monitoring specific activity type to particular level of detail:

- *Tabber* for basic standalone applications monitoring;

- *Wordik* for more detailed monitoring of specific application, which we consider as important activity context for the search in our scenario of novice researcher – a text editor, in our prototype MS Word;
- *Annota* for monitoring searcher's activity related to applications executed within a browser.

Tabber is a standalone application that captures searcher's interaction between applications. It captures name of application the searcher has switched to, and the title of the application window. They are the only content information capturing for every possible application. It also catches each copy and paste event and copied text for particular application.

Wordik is Microsoft Word addin. We consider that our searcher searches on the Web often information about something he is actually writing about. To get content of a document, addin is a must. It captures names of topics the searcher is actually writing and the text that is around actually editing position.

The last part of the activity logger monitors web applications. Web application is presented to its users through a website with assigned domain on the web space. Web application monitoring is realized using monitoring component of the *Annota* service. Annota [13] is a service enabling the researchers to annotate and organize scientific resources on the Web (mainly scientific publications) and to share them with colleagues. It is realized as a Mozilla Firefox extension. Annota captures URL address and title of web-pages a searcher is actually browsing. Application context includes also a text that searcher has selected or copied.

Annota also catches links the searcher has clicked and web-page change events, so we may examine searcher's click-through. Click-through consists of search results the searcher has clicked, active time spent on the clicked pages and interesting parts of visited pages like text, which the searcher has copied. We are able to evaluate click-through more precisely comparing to standard analyses thanks to this information. We do not analyze just which results the user has or has not clicked but also time he spent on the result page and an estimation whether the result was useful (user selected or copied text).

Purpose of described three modules is an evaluation of our proposed method. New specific modules can be added to capture activity from other applications on various level of detail. We also tried to hook each application GUI text but due to unpredictable behavior in case of a random exception we decided not to include it into present version.

Activity context expressed as information about user activity is processed to keywords in normalized form using natural language processing techniques. Firstly, we remove all stop-words that have no meaningful value. Secondly, we stem the rest of the keywords, i.e. group together the different inflected forms of the keyword, because two words may be the same but they may have different verbal form and we need to know that they are the same.

However, different languages have different linguistic processing methods, so language recognition is crucial to choose proper method, otherwise the keyword will be distorted. However, content of activity consists of just few words (commonly no more than 7 keywords) so it is hard to detect the right language. For this reason we constrained our experiments to two languages: Slovak as our native language and English. Both languages were used by searchers who were selected for our experiments.

An application activity context model is represented by a vector of keywords consisting of keywords in natural language and their normalized forms. Each keyword is represented together with its timestamp, which serves for computing the keyword relevancy to the application activity context. Keywords that were captured before selected time threshold are not considered for particular search query at all, because they for the most cases have a very small or no impact on actual activity context.

B. Finding related application to a query

Important task for finding related application to given query is to choose which application has had direct influence on current search intention. We assume that application context is the same as, or similar to, a search context, so our task is to compare each application context to the search context and find similar one.

To determine, which application is connected to the query we propose two strategies that consider various types of connections between an application and a query: syntactic comparison and semantic comparison. *Syntactic comparison* is used to find out how often keyword in a query occurs in application context. We try to find conjunction between the query keywords and an application context by comparing stemmed keywords and also keywords in natural language.

However, people mostly search queries without diacritic. Diacritic is very important for our Slovak language processing methods because we are able to use them only if we have right word. To add diacritic where it is missing we use query search result list as it contains correct diacritic.

Semantic comparison is used to find a connection between words that have the same or similar meaning but different lexical form. Such connection specifies meaning of a query. There are many types of semantic relations. In our research we aim on synonyms and hyponyms of a query. To get synonyms and hyponyms of the actual query we use *Wordnet.API* (wordnet.princeton.edu/). It groups English words into sets of synonyms called synsets, and records the various semantic relations. Semantic comparison compares synonyms and hyponyms of the query keywords to each application context.

We can tell that an application is related to a query only if we are able to find a connection by one of above described strategies. Otherwise it would be high probability of connecting inappropriate application to the query which would result in analyzing of wrong search context and extending query by wrong keywords.

We may find connections to more applications, even to those applications that are not related to a query. It is possible because some keywords are easy to find in many application contexts, e.g. do, nice, hard. To select only those applications that are related to the query we used rank-based evaluating method. The applications were sorted by indicators evaluating different types of activity and connections to the search query and best gained maximum 5 points, second best 4 points etc. We propose following eight indicators: active time spent (rank based), count of switches in an application (rank based), count of copy activity (rank based), rank in the last used application, count of semantic connections, and count of syntactic connections

We get values for each indicator from the activities captured in each application. Probability that the application is related to a query is equal to a relevance of the application calculated as sum of weighted values. We get indicators weights experimentally by comparing our results to explicit feedback. We select the most relevant application as the application related to a query.

C. Extending query by keywords from search context

An application related to the query provides larger context of what a searcher is trying to find because context of the application related to the query is the same as, or similar to, search context. We extend the query by few most relevant keywords from this extended search context.

Choosing right keywords characterizing search context is crucial. The application's context consists of lots of words and only few of them are relevant, so we need to assign weight to each keyword to be able to choose the most relevant keywords. We proposed two types of weights: weight in general and weight related to a query. Weight in general reflects a keyword's relevance to the current application context. We need to assign weight to each application activity content. If the activity content is a huge text, weight is calculated by third party text metadata extraction service *Metallurgy* (metallurgy.fiit.stuba.sk). Otherwise, if the activity content is just a short text (often a name), each keyword in the activity content has the same weight. Weight is also a function of time so keywords stored longer time ago have lower weight. Keywords stored before threshold are not considered at all.

Not every application activity context is connected to current query, so we need to specify how much is the query related to it. Syntactic and semantic connections between the query and an application reflect content weight related to the query. Each connection has its own weight and each keyword weight in specific activity content is equal and inherits its content weight. Different homonyms have different semantic connections which means that extending query by semantically connected keyword lessen search ambiguity; hence, this keyword gets maximum weight.

We reorder keywords from the application context by their total weight. Total weight is weight in general multiplied by weight related to a query. We extend the query by top keywords that achieved weight higher than a threshold chosen experimentally. To not change meaning of a query we reduce number of keywords to 3 for queries consisting of maximum 3 keywords and to 4 for longer queries. We also suggest next top 4 keywords that may help a user to reformulate query under the search bar. There is no threshold for these keywords.

For example, a user is writing an essay and he searches for "*mount weather"*. We find out that this query is related to the essay and evaluate Microsoft Word activity context. Keyword *mountain* gets maximum weight because it is semantically related to a query. Keyword *Everest* occurs 4x next to query's keyword *mount* and also 2x in the title of actually editing text, so it gets 4**so*+2**to* points, where *so* is weight of syntactic occurrence , *to* is weight of title occurrence and *so* < *to*. No other keyword has weight higher than selected threshold, so just these keywords would expand actual query.

IV. EVALUATION AND EXPERIMENT RESULTS

We hypothesize that searcher's information need comes in many cases from a recently used application. To prove this we proposed an experiment with aim to find a connection between an application and a query and evaluate accuracy of the connection. We asked 9 students of a technical university to log their activity while using their own computers for normal activities. Moreover, we asked them provide us explicit feedback – when searching on the Web they picked out an application related to their actual query. We modified general search engine (we selected Google search) results page and allowed the searcher to select the application that activated the search (if any) by clicking on its name from the list of possible applications (see Fig. 1). To keep it simple, we showed 4 recently most used standalone applications and 4 recently most used web applications. We also provided an

Figure 1. Explicit feedback during the experiment – selecting a connection between an actual query and an appliction.

option if the query is related to no application. Experiment took 6 weeks and users gave explicit feedback 472 times.

Table 1 shows count of queries that users explicitly connected to an application and type of the application. Notice that 88.14% of queries are related to an application the searcher has recently used, i.e. the searcher's search need comes in most cases from an application used recently so search context is mostly related to an application context.

TABLE I. RESULTS OF EXPLICIT CONNECTED QUERIES

	All	No connection	Application	
			Software	*Domain*
Count	472	11.86%	46.16%	37.17%

Our goal was to find an application related to a query without need of user's explicit feedback. We assume that search context is similar to a related application context, i.e. there is a connection between a search context and an application context. We proposed and realized an experiment to evaluate the success rate of our strategies for finding a connection between a query and a related application.

In our experiment we considered that an application has been used recently. For all experiments we chose constant time 15 minutes and considered only context of applications gained within 15 minutes range. We tested more time ranges and this was sufficient because we could not get much better results for longer time range and we got much worse results for shorter time range. In our experiment we were finding a connection between a query and explicitly connected application context.

A. Syntactic connections between application and query

Firstly, we evaluated our strategies for finding a connection by syntactic comparison between a query and an application. Table 2 shows success rate of this strategy. We were able to find a connection for 58.7% queries. We were able to find a connection for more web applications than standalone applications. It is mainly because our prototype loggers gather less information on standalone applications context than on web applications context. Moreover, some information is more expressive for web applications, e.g. title of a web-page in most cases expresses content of the web-page and it changes at each single web page of the web-application. On the other hand, title of a standalone application mostly consists of an application name and name of a save file, which does not provide often a valuable information.

Notice that 81.8% queries that the user explicitly selected as related to an application are searched in English. It is more than 90% queries for the queries related to a standalone application. It is mainly because support for standalone applications is mainly in English. Each user in our experiment speaks fluent English, so it was easier for them to find help in English. They searched in Slovak mostly when they needed to find local information. More than 80% Slovak queries are connected to a web application so there is much higher probability that Slovak query is connected to a web application than a standalone application.

TABLE II. SYNTACTIC COMPARISON BETWEEN A QUERY AND AN APPLICATION CONTEXT

	Slovak queries		English queries		Total	
	Found	*Total*	*Found*	*Total*	*Found*	*Total*
Standalone apps	21.4%	7.4%	52.0%	92.6%	49.7%	100%
Web application	73.1%	29.9%	66.4%	70.1%	68.4%	100%
Total	62.2%	18.2%	57.9%	81.8%	58.7%	100%

B. Semantic connection between application and query

Next, we evaluated our strategy for finding a connection between a query and an application using also semantic relations. We were able to analyze only English queries because there has not been way to get Slovak hyponyms or synonyms yet. Table 3 shows success rate of semantic comparison between a query and an application context. We were not able to find much more connections because we were able to find just few connections by comparing also query synonyms and hyponyms.

TABLE III. SYNTACTIC AND SEMANTIC COMPARISON BETWEEN A QUERY AND AN APPLICATION CONTEXT

	Slovak queries		English queries		Total	
	Found	*Total*	*Found*	*Total*	*Found*	*Total*
Standalone applications	21.4%	7.4%	60.6%	92.6%	57.7%	100%
Web application	73.1%	29.9%	66.4%	70.1%	68.4%	100%
Total	62.2%	18.2%	63.0%	81.8%	62.8%	100%

We were able to find connection for 62.8% queries connected to an application. In most cases we found syntactic similarity between a query and an application. We were not able to find more connections because we wanted to log no private information so we logged just texts we considered as interesting.

Users always selected that query is connected to only one application. However, we were able to find a syntactic or semantic connection in more than 1 application many times (Fig. 2). It means that it is not guaranteed that the application that is connected to a query is also related to the query.

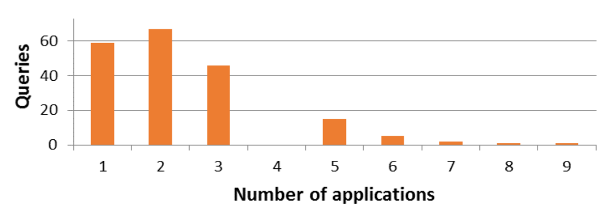

Figure 2. Distribution of number of applications with identified connection for queries in the experiment.

Many times a query is not related to any application. In these cases we cannot extend the query as we are not able to get search context. To prevent query expansion in such cases we must find no connection between a query and an activity context. We analyzed searches in our experiment that were determined as no related searches. We found no connection and no distorted search results for 82% queries.

V. Conclusions

We proposed and evaluated a method for automatic activity context-aware query expansion. It includes our proposal for modeling activity context. Each application has different activity context so we model activity context for each application. To capture activity we implemented an activity logger that consists of three independent logging applications for three situations: standalone applications in general, selected standalone application for deeper analysis and web-based applications. For analysis we used natural language processing and transform natural language text into keywords representing actual activity context model.

Part of the method is finding a connection between basic search context and context of an application that is potentially related to the query. We proposed and implemented two strategies for this task. Syntactic comparison compared keywords trying to find syntactic similar keywords. Semantic comparison compared keywords semantically close to query, i.e. synonyms and hyponyms of a query. Finally, we extend the search context by relevant keywords from the application activity context. Relevancy of a keyword reflects how much is the keyword important for the application activity context and how much is related to a query.

To evaluate our approach we proposed and realized experiment. 9 people were logging their activity and explicitly selecting an application that is related to their actual query. They selected that 88.14% of queries are related to an application they have recently used, i.e. searcher's search need comes in most cases from an application used recently.

To find a related application to a query we firstly needed to find out if there is a connection between the query and the application. Syntactic comparison with semantic relations found a connection for 62.8% queries. The most of connections to the related application (58.7%) were found using just syntactic comparison. We also evaluated searches that had been related to no application. For these queries the task was to find no connection to an application by our strategies and we found no connection for 82% searches.

In further work we aim in further improvement of selecting which application that a connection has been found is related to the query and which is not. We plan to evaluate proposed implicit indicators for each connected application and the one that would get the highest score based on these implicit indicators would be the application related to the query. Moreover, we can consider also other types of contexts such as seasonality, which can help not only in direct search query expansion as it is proposed in [14], but also in selecting of related activity context.

We also plan to move our attention to evaluating activity context in specific domain, namely digital libraries. Our primary scenario is researcher beginner working on his domain survey. We may use activity context related to this domain, e.g., we suppose that the researcher is writing or reading a document when he initiates a search for new information in a digital library. Therefore we can plan to evaluate activity in Microsoft Office and other text editors (including those web-based) more deeply and examine influences of different text styles and properties on user's search context.

Acknowledgment

This work is partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10 and by the Scientific Grant Agency of the Slovak Republic, grant No. VG1/0971/11.

References

[1] M. Kamvar, M. Kellar, R. Patel, and Y. Xu, "Computers and iphones and mobile phones, oh my!," Proc. 18th Int. Conf. World Wide Web (WWW 09), ACM, 2009, pp. 801-810.

[2] S. Bateman, J. Teevan, and R. W. White, "The search dashboard," Proc. of the 2012 ACM annual conf. on Human Factors in Computing Systems (CHI 12), ACM, 2012, pp. 1785-1794.

[3] A. Aula, R. M. Khan, and Z. Guan, "How does search behavior change as search becomes more difficult?," Proc. of the 28th int. conf. on Human factors in comp. systems (CHI 10), ACM, 2010, pp. 35-44.

[4] H.-S. Teo, "An activity-driven model for context-awareness in mobile computing," Proc. 10th Int. Conf. Hum. Comput. Interact. with Mob. devices Serv. (MobileHCI 08), ACM, 2008, pp. 545-546.

[5] M. Eirinaki and M. Vazirgiannis, "Web mining for web personalization," ACM Trans. Internet Technol., vol. 3, no. 1, Feb. 2003, pp. 1-27.

[6] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search," ACM Trans. Inf. Syst., vol. 25, no. 2, Apr. 2007, Article 7, 29p.

[7] K. Leung and D. Lee, "Deriving concept-based user profiles from search engine logs," Knowl. Data Eng. IEEE Trans., vol. 22, no. 7, 2010, pp. 969–982.

[8] D. Jiang, K. W.-T. Leung, and W. Ng, "Context-aware search personalization with concept preference," Proc. of the 20th int. conf. on Inf. and knowledge management (CIKM 11), ACM, 2011, pp.563-572.

[9] T. Kramár, "Utilizing Lightweight Semantics for Search Context Acquisition in Personalized Search," Inf. Sci. Technol. Bull. ACM Slovakia, vol. 6, no. 1, 2014, pp. 1-8.

[10] T. Kramár, M. Barla, and M. Bieliková, "Personalizing Search Using Metadata Based, Socially Enhanced Interest Model Built from the Stream of User's Activity," J. of Web Engineering. Vol. 12, No. 1&2, 2013, pp. 65-92.

[11] A. K. Dey, G. D. Abowd, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," Proc. of the 1st int. symposium on Handheld and Ubiquitous Computing, 1999, pp. 304-307.

[12] P. Návrat, "Cognitive traveling in digital space: From keyword search through exploratory information seeking," Central European J. of Computer Science, vol. 2, no. 3, 2012, pp. 170-182.

[13] M. Holub, R. Móro, J. Ševcech, M. Lipták, and M. Bieliková, "Annota: Towards Enriching Scientific Publications with Semantics and User Annotations, " Proc. of 3rd Int. Workshop on Mining Scientific Publications (WOSP 14), D-Lib Magazine. To appear.

[14] T. Kramár and M. Bieliková, "Context of Seasonality in Web Search," Proc. of European Conf. on Information Retrieval (ECIR 14), LNCS Vol. 8416, Springer, 2014, pp. 644-649.