Extracting user interests from search query logs: A clustering approach

Lyes Limam, David Coquil, Harald Kosch
Fakultät für Mathematik und Informatik
Universität Passau, Germany
{limam,coquil,kosch}@dimis.fim.uni-passau.de

Lionel Brunie
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
lionel.brunie@insa-lyon.fr

Abstract — **This paper proposes to enhance search query log analysis by taking into account the semantic properties of query terms. We first describe a method for extracting a global semantic representation of a search query log and then show how we can use it to semantically extract the user interests. The global representation is composed of a taxonomy that organizes query terms based on generalization/specialization ("is a") semantic relations and of a function to measure the semantic distance between terms. We then define a query terms clustering algorithm that is applied to the log representation to extract user interests. The evaluation has been done on large real-life logs of a popular search engine.**

Keywords-Query terms Taxonomy, log analysis, clustering

I. INTRODUCTION

Mining processes can be applied to large search query logs in order to extract knowledge about user interests. This is in particular a necessary step for the design of true user-centric applications in which user search behaviors are identified and taken into account [1,2,3]. In recent years, much research has been done in the domain of search query logs analysis. To date, researchers have mostly focused on statistical methods for extracting knowledge from these data. These proposals are not applicable to problems related to the semantics of the data such as the identification of users search interests. To deal with such issues, we propose a two-step method. The first step aims to produce a semantically enhanced global reference for the whole log while the second step considers this global reference as a platform on which the user interests are identified.

We define the global reference as a data structure organizing the query terms in a taxonomy equipped with a semantic distance function. Thus, we describe an approach to construct this taxonomy over the terms used in keyword search logs by means of the WordNet lexical database of English terms. The construction of the taxonomy is based on two principal aspects. The first is the hierarchical relation between the terms established by the hypernymy and generalization/specialization relations ("is a"). This kind of relation enables to sort the terms into different levels of abstractions and organize them in a tree structure. The second aspect is the semantic distance between two terms connected in a IS-A relation; it represents the weight of the edges of the tree obtained previously. Here, we introduce a new weighting function that takes into account the abstraction level of terms. In fact, two terms in the bottom of the hierarchy are considered to be closer than two terms situated at the top of the hierarchy. Furthermore, the distance function is generalized to take into account every couple of terms in addition to those that are directly in a IS-A relation.

After building the query terms taxonomy, we apply a clustering algorithm, which extracts users' interests in form of clusters. The algorithm is based on a threshold that enables to control its precision and to adapt the results according to the target application. Thanks to the hierarchical nature of the taxonomy and the semantic distance the algorithm is particularly fast. This approach was evaluated using real-world Web logs from the AOL search engine. It is important to precise that the main contribution in this paper is the semantic distance function in addition to the clustering algorithm as we cannot separate between them as we will show further.

The remainder of this article is organized as follows. Section II presents the related work. Section III describes the process for extracting the global representation while the section IV is dedicated to the clustering algorithm. In the section V, we describe the experiments on the AOL logs. Finally, section VI presents some conclusions and perspectives of future work.

II. RELATED WORK

A. Search query log analysis and its challenges

Since the emergence of the first search engines, many researchers have proposed methods for extracting information from search logs [4,5,6,7]. This topic is frequently referred to as Search query Log Analysis, or SLA. In the SLA context, a search log may be defined as an electronic record of interactions between a search engine and users searching for information on that search engine [8]. Hence, the set of queries issued by a user in a specific period of time contains information about his/her topics of interest; one goal of SLA can be the extraction of this information. However, several authors [9,10,11] have criticized analysis processes that rely solely on search logs because they record neither the users' perceptions of search nor their satisfaction with the results. This may complicate the task of accurately identifying user search behavior. On the other hand, this valid criticism does not imply that search logs are useless. First, a number of applications of SLA can provide useful results without needing the missing information identified by the critics. This is for example the case of query recommendation and distributed Web searching optimization [1,3] (see section II.*C* for more details on applications of SLA). Moreover, the quality of the knowledge produced by SLA can be enhanced by using external sources of information in addition to query logs. Indeed, in this paper

we argue that it is possible to extract a large amount of information from search logs by including the implicit semantic relations between query terms in the process.

B. Search query log and semantics

Jansen [8] proposed a unified methodology to conduct the analysis of search query logs. He defined three levels of analysis: term level (terms as the basis for analysis), query level (queries as the base metric) and session level (in-session interactions). A closer examination of the first two levels reveals that the analysis is mostly statistical and does not consider semantics. Indeed, for term level analysis, all proposed metrics (term occurrence, total terms, unique terms, high usage terms and term co-occurrence, etc.) are statistical measures that consider neither the problem of the polysemy of query terms nor semantic relations between terms. The query analysis level is also based on statistical metrics: initial query, modified query, identical query, unique query, query complexity and failure rate. These metrics are actually based on the classic definition of a keyword search query as a string list of zero or more terms submitted to a search engine [8]. Besides, the session level characterizes the time aspects (session duration, queries frequency, etc.) in a user interaction which is out of scope of our issue.

Another limitation of existing metrics is that they are exclusively based on string comparison. For instance, the queries "Hotel Booking" and "Booking Hotel" are considered to be different though they are in fact semantically equivalent. In our work, we strive to address these shortcomings by enhancing the SLA methodology with semantics. We argue that a larger amount of more meaningful information may be extracted from the logs if external information about semantic relations between terms is coupled with the query logs before the start of the analysis phase. We propose a method to achieve this, which results in a novel way of representing the logs in the form of a taxonomy of query terms. Such a global representation lends itself very well to the definition of a metrics that measures the distance between query terms. Subsequently, the taxonomy equipped with the distance function enables us to define a semantic query clustering method, which can extract user interests from search data.

C. Applications of search query log analysis

The analysis of search logs yields information about the search topics, which is useful in many applications. In information retrieval, the system may use the extracted view on user interests to personalize its search process in order to better match the user expectations. In [12] the authors proposes a method that exploits the click history of each user to build a topical ontology, i.e., a model of the different topics that interest the user. The topical ontology is used to guide the search engine when it receives a query. In addition, the authors define a ranking function to sort the search result according to the user preferences. Their experiments show that user preferences can be learned accurately thanks to the topical ontology. In the content management domain, the analysis of queries is used for instance to determine the most frequently asked questions. [2] deals with this issue by treating it as a query clustering problem. They propose to combine the queries with the documents selected by the user. Query clustering is conducted based on the idea that queries are similar if the selected documents are the same. [13] constructs a hierarchical classification of query terms, which is called a taxonomy. The terms are grouped together in hierarchical clusters based on a vector space model. Each term is represented by a vector V_i of characteristics and each component V_{ij} (i.e. the j^{th} unique term corresponding to the i^{th} query term) is calculated by a tf/idf weighting function on the top ranked documents. The clusters are then constructed by comparing the terms using the Euclidian distance. It is clear that the addressed issue is different from the one that we address, which is a semantic issue.

III. KEYWORD TAXONOMY: A GLOBAL SEMANTIC REPRESENTATION

A. External source of semantics

One of the objectives of this paper is to enhance search query logs analysis with semantics, i.e. to take into account the meanings of log terms and the semantic relations between them. To this end, we make use of the concept of synsets. A synset is a set of words that are synonymous in a specific context; a specific meaning is attached to each synset. Typically, in the natural language, words are often polysemous, and can thus belong to several synsets. In this context, what we mean by semantic relations between log terms is more precisely defined as relations between synsets.

As we deal with textual queries, a thesaurus can provide us with information about synsets and the relations between them. To this end, we use the WordNet lexical database [14], an English thesaurus that includes several types of semantic relations e.g. hyponymy vs. hypernymy (is a), holonymy vs. melonymy (is part of), etc. We have made this choice for two reasons. First, WordNet is one of the richest thesauruses; it comprises more than 150000 synsets. Second, it is compatible with a large number of dictionaries and other semantic sources e.g. DBpedia [15], which provide mappings to the synsets, therefore extending its semantic scope. WordNet defines two kinds of word characteristics: structural (i.e. relations between synsets) and non structural (i.e. gloss: definition of a synset, list of words included in a synset etc.). In our work, we use the hypernymy relation as a basic structural characteristic that defines a hierarchical order between terms. Semantically, it is the relation of being super-ordinate or belonging to a higher rank or class, e.g., "vehicle" is the hypernym of "car". Moreover, we use WordNet to determine all possible synsets for each term in order to distinguish between all possible senses of a term.

B. Log pre-processing phase

The goal in this step is to obtain a refined vocabulary of query terms that will form the initial set of elements of the taxonomy. There are two levels of pre-processing: the query level and the term level. Query pre-processing is mostly a filtering process. First, identical queries resulting from the format of the logs are deleted. Second, the "bad queries" are filtered out. We mean by a bad query a non interpretable

query, e.g., part of URL, IP address, email address, etc. This kind of query may for example occur when the user is trying to check information without expecting relevant result. At the term level, we apply a classical lexical analysis to the queries. First, each query is split into a set of terms by means of a tokenizer. The goal is to extract the set of meaningful keywords. The tokenizer deletes all stop words, such as articles and prepositions. Second, each term is processed separately using stemming and lemmatization methods. These tools group together the different inflected forms of a word into a single item, e.g., the plural and singular form of a noun are identified as a single item. Third, query terms that appear more than once in the log are deleted. Fourth, we match each term with a WordNet synset; terms that are not in WordNet are considered non-interpretable for now.

C. Building the taxonomy

1) Basic hypernymy tree construction

The aim of this step is to construct a hierarchical structure that semantically relates the terms of the search logs. To this end, we use the relation "is a" (hypernymy).

The hypernymy tree is generated as follows. First, for each term, we search its hypernym with the lowest abstraction level in the set of query terms. The process is then applied recursively; it stops at the last hypernym that has no hypernym in the log. For example, let us suppose that the terms "football", "sport" and "diversion" appear in the logs. These terms match the synsets "football.1", "sport.1", "diversion.1" respectively. According to WordNet, sport.1 is the first hypernym of football.1 and diversion.1 is the first hypernym of sport.1, hence we get the path "football.1" isa "sport.1" isa "diversion.1". We call a path of successive hypernyms an "Hpath". Each term has as many Hpaths as the number of synsets to which it belongs. Second, the produced Hpaths are merged based on their common items to form a tree structure. Fig. 1 shows the Hpaths of Handball.2, Football.1 and Soccer.1 connected in one sub tree.

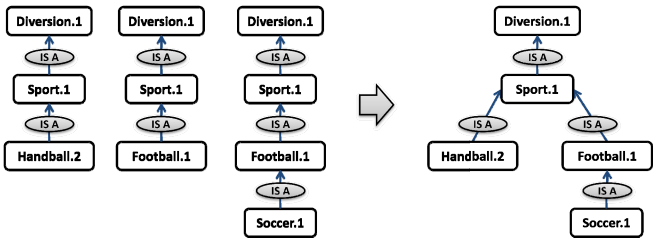

Figure 1. Hypernymy tree construction by Hpaths merging

2) Semantic distance function

In order to provide the taxonomy with a means of quantitatively evaluating the semantic distance between two terms, we first introduce a weight associated to the individual "is a" links. This weight is defined as a decreasing function of the semantic proximity between the parent and the child (i.e.) the higher the weight, the less related the terms. As all relations contained in the taxonomy are of type "is a", the nodes go from the most general at the top to the most specific at the bottom. Therefore, two connected terms at the bottom of the taxonomy are more closely related than two connected terms at the top. The weighting function should thus be decreasing with respect to the level of the terms.

The choice of a decreasing function is motivated by the fact that in the log analysis we give more attention to relations between concrete objects and less abstract terms rather than on relations between general terms. In addition, the weighting function enables to characterize the dimension of the domain covered by the query. This means that depending on the terms being used, the query varies from general (i.e., large domain dimension) to specific (small domain dimension). In fact, the more general the terms, the larger is the distance between them and hence the larger is the domain covered by the query. For example, let us consider two queries Q_1 and Q_2 where Q_1 is "Sport leisure" which is a general query about sport and Q_2 is "football player X", a more specific query about a football player.

Based on this, we define the weight function:

$$W(x,y) = 1/l(y) \qquad (1)$$

where "x" and "y" are two terms related by the direct relation y "is-a" x and "l" is the function that returns the level of "y". Fig. 2 illustrates the use of this function.

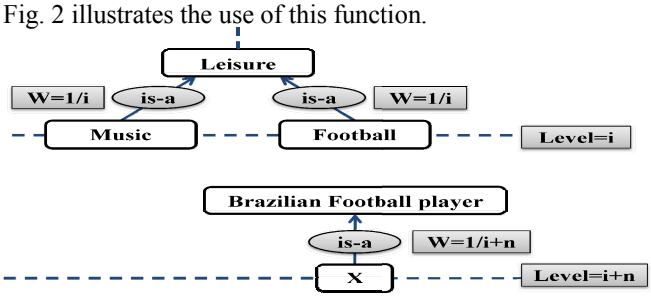

Figure 2. Decreasing weight function

In order to be able to compare every couple of terms of the taxonomy, we introduce a distance function that depends on the weights of the edges composing the path between the terms. Depending on the type of path, there are two ways to compute the distance. If the path is straight, the distance is the sum of all weights of its edges. If the path is deviated (see Fig. 3), the distance is the sum of distances of the two sub-paths which have the common hypernym as the upper bound. Hence, the distance is defined as:

$$D(x,y) = \begin{cases} \sum_{i=l(x)+1}^{l(y)} (1/i) & if\ x \to y\ is\ a\ straigh\ path \\ \sum_{i=l(c)+1}^{l(x)} (1/i) + \sum_{i=l(c)+1}^{l(y)} (1/i)\ otherwise \end{cases}, \qquad (2)$$

"l" being the level function and "c" the common hypernym of terms "x" and "y".

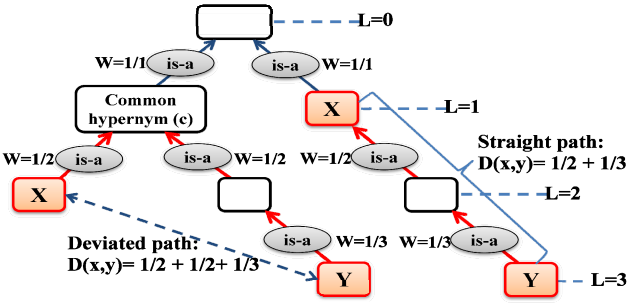

Figure 3. Distance measurement

With this formula, we want to focus on the abstraction level of terms to evaluate their distance. We are aware of the amount of work that has been done on semantic distance, especially in IR area [16,17,18,19]. These formulas are either depth based, path based or information content based parameters. The two first parameters are related to the number of edges separating the compared terms while the last is a value that represents the importance of a term within a corpus; it is used to identify the best common hypernym for the compared terms, i.e. with the highest IC. In the proposed semantic distance "D" we aimed to model the semantic variation (the weight function) of the "is-a" relation depending on the abstraction level of the related terms (see Fig. 2). This property was never been explicitly discussed in earlier work. However, we plan to verify its effectiveness with respect to the other functions.

IV. QUERY TERMS CLUSTERING: ALGORITHM

One of the major goals behind the search log analysis is the extraction of the user interests. Technically, the problem is defined as a query terms clustering problem. This problem has been discussed earlier in different ways [2,13] but without considering the semantic relations between query terms. On the contrary, we propose an algorithm based on the semantic distance defined above (see Fig.4). Here, we define a cluster as a set of query terms such that the distance between each pair of query terms of the set is inferior to a predefined threshold.

The proposed algorithm is designed to take advantage of the decreasing property of our distance function while none of the reviewed algorithms is adapted to this property. It acts as follows: First, it starts by finding the deepest term in the taxonomy (e_d). The goal is to get the most specialized terms in one cluster. Second, it tries to find the closest terms to this term that respect the threshold by switching in two dimensions, height (parents) and width (children) of the taxonomy. Thus, first it checks the distance between e_d and the first parent term with respect to the threshold using the function "*cluster_up*" then, if the condition is satisfied, it uses *cluster_down* to similarly check the children terms related to that parent. The process is recursively applied on the next parents until there is no term that respects the threshold with the initial term. The result of this iteration is one cluster; all terms of the cluster are excluded from the next iterations. In the next iteration, the deepest term is sought again. The algorithm iterates on the rest of the terms until they are all clustered.

Starting with the deepest term clearly favors the construction of large clusters of specialized term at the expense of general terms as it checks all the children of the closest parent before going to the next parent. For example, the algorithm produces the football player cluster: "Messi, Rooney, Zidane" but puts the terms "sport" and "music" into two different clusters as they are situated at the top of the taxonomy and hence less related. One of the strong points of the algorithm is related to its complexity. Starting from the deepest term of the taxonomy limits the number of checking operations to "n", where n is the number of terms. For example, if we have to cluster the terms e, e1, e2, e3, where e2 isa e1, e3 isa e1 and e is the deepest term connected to e1. Following the algorithm, the cluster is initialized by e. If the distances between e and e1, e2, e3 respectively are less than the threshold, e, e1, e2, e3 form one cluster and we do not need to check the distances between e1, e2 and e3 since e is the deepest element. Thus the complexity of the algorithm is O(n), n being the number of terms.

```
QUERY TERMS CLUSTERING ALGORITHM:
  T   // Taxonomy with weighted links
  E= {e0, e1...}  // set of query terms (nodes)
  C = {∅}   // set of clusters
  ci = ∅    // ci ∈ C
  D    // distance function
  ts = Value   // threshold
  While Not (empty(E))
     e_d = deepest(E) // find the deepest term
     ci = ci ∪ {e_d }  // init. ci with the deepest term
     cluster_up(e_d , parentOf(e_d))
     C = C ∪ {ci}
     E=E-{ci}
  end
end
```

```
function cluster_up(predecessor, e)
If D(e_d,e)≤ ts
  While has_children(e)
     if  childOf(e)≠ predecessor
     cluster_down(pull_childOf(e))
  end
  ci=ci ∪{e}
endif
cluster_up(e , parentOf(e))
end
```

```
function cluster_down(e)
If D(e_d,e)≤ ts
  While (has_children(e))
     cluster_down(pull_childOf(e))
  end
  ci=ci ∪{e}
endif
end
```

Figure 4. Query terms clustering algorithm

V. EVALUTION

The conducted experiments are based on real life Web logs. They are recorded by the AOL search engine over 650k of users in a period of 3 months [20]. The log files are presented in the form of a table in which each entry corresponds to a query recorded when a user clicks on a result item or a duplicated query if the user browses different result items for the same query. In addition to the query terms the search engine records other information e.g. the user identifier, the query time, etc. Such information is not considered in the taxonomy construction process but it may be used for other plans.

TABLE I. QUERY LOG PRE-PROCESSING

Nb of considered queries	Nb of terms	Nd of considered terms	Nb of synsets
~1,6 M	~5,6 M	~47 k	~55 k

The collection of queries is pre-processed as described in the section III.2. To this end we used the Lucene Lexical Analyzer [21]. The resulting set of keywords represents the log vocabulary. Finally, each term is matched with its different synsets by means of the WordNet JAVA API [22].

Table 1 summarizes the results of the pre-processing phase. The log is characterized by a high redundancy in queries and in keywords (i.e. terms). The former is due to the click-based recording of queries and the latter is caused by

the polysemous usage of keywords by the users. As a result, the final set of keywords is significantly reduced. In addition, several keywords have no WordNet Entry. 68% of the keywords are successfully matched with WordNet; the rest are considered non-interpretable. Nevertheless, as explained above (section III.A), one advantage of WordNet is its capability to integrate other dictionaries to extend its semantic scope and hence, make possible to reduce the proportion of non-interpreted terms in the future.

The extracted taxonomy represents a hierarchical structure that contains 14 levels on which terms with their synsets are distributed. The taxonomy is generally well balanced on the number of children by each parent (hypernym), except a small number of nodes that aggregate a larger number of children. This satisfactory distribution of terms on the nodes comes from the fact that the topics requested by the users are diversified on different domains.

The clustering algorithm has been applied on the produced taxonomy. The number of the obtained clusters depends on the threshold (see Fig.5). For the moment the clusters are verified manually. This qualitative evaluation is encouraging. Thus, we plan in the near future to draw a comparison between the clustering techniques cited previously [2,13] including our method.

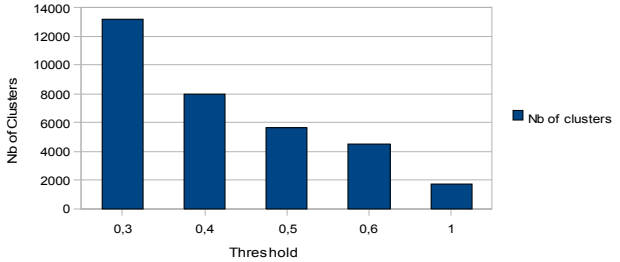

Figure 5. The number of clusters with respect to the threshod

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a method to extract user interests from search query logs. The global representation is composed of a semantic taxonomy of query log terms together with a function that evaluates the semantic distance between the query terms. The distance takes into account a new property related to the abstraction level of terms. In addition to that, we proposed a fast algorithm that enables to extract the user's topic of interest in form of clusters of query terms. Such precious information is an input for several applications e.g. recommendation systems. Finally, we are thinking about improving the source of semantics by putting together several tools namely WordNet and DBpedia in order to reduce the number of non-interpretable terms.

REFERENCES

[1] C. Lucchese, S. Orlando, R. Per, and F. Silvestri, "Mining query logs to optimize index partitioning in parallel web search engines," in Proceedings of the 2nd international conference on Scalable information systems, Suzhou, China, 2007, pp. 1-9.

[2] J. R. Wen, J. Y. Nie, and H. J. Zhang, "Query Clustering Using User Logs," ACM Transactions on Information Systems (TOIS), vol. 20, no. 1, pp. 59-81, 2002.

[3] Z. Zhang and O. Nasraoui, "Mining search engine query logs for social filtering-based query recommendation," Applied Soft Computing, pp. 1326-1334, 2008.

[4] R. Baeza-Yates and C. Castillo, "Relating web characteristics with link based web page ranking," String Processing and Information Retrieval, pp. 21-32, 2001.

[5] M. Chau, X. Fang, and O. R. Sheng, "Analysis of the query logs of a web site search engine," Journal of the American Society for Information Science and Technology, pp. 1363-1376, 2006.

[6] Bernard J. Jansen, Amanda Spink, and Isak Taksa, Handbook of Research on Web Log Analysis. Hershey, PA, USA: IGI Global, 2008.

[7] C. Silverstein, H. Marais, M. Henzinger, and M Moricz, "Analysis of a very large web search engine query log," in SIGIR Forum, 1999, pp. 6-12.

[8] Bernard J. Jansen, "Search log analysis: What it is, what's been done, how to do it," Library & Information Science Research, p. 407–432, 2006.

[9] D. Blecic et al., "Using transaction log analysis to improve OPAC retrieval results," College and Research Libraries, pp. 39-50, 1998.

[10] M. Kurth, "The limits and limitations of transaction log analysis," Library Hi Tech, p. 98−104, 1993.

[11] A. Phippen, L. Sheppard, and S. Furnell, "A practical evaluation of Web analytics," Internet Research, pp. 284-293, 2004.

[12] S Stamou and A Ntoulas, "Search personalization through query and page topical analysis," in User Modeling and User-Adapted Interaction, 2009, pp. 5-33.

[13] S. L. Chuang and L. F. Chien, "Towards automatic generation of query taxonomy: a hierarchical query clustering approach," in International Conference on Data Mining (ICDM), 2002, pp. 75-82.

[14] C. Fellbaum, WordNet:An Electronic Lexical Database, The MIT Press, Ed. Cambridge MA, London: Fellbaum C., 1998.

[15] (2010) DBpedia. [Online]. http://wiki.dbpedia.org

[16] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and Application of a Metric on Semantic Nets," IEEE Transactions on Systems Management and Cybernetics, pp. 17-30, 1989.

[17] J. Lee, M. Kim, and Y. Lee, "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies," Journal of Documentation, pp. 188-207, 1993.

[18] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," in 14th International Joint Conference on Artificial Intelligence (IJCA), 1995, pp. 448-453.

[19] Giuseppe Pirró and Nuno Seco, "Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content," in OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems, Monterrey, Mexico, 2008, pp. 1271-1288.

[20] G. Pass, A. Chowdhury, and C. Torgeson, "A Picture of search," in 1st international conference on Scalable information systems, Hong Kong, 2006.

[21] (2009) Lucene 2.4.1 API. [Online]. http://lucene.apache.org/java/2_4_1/api/index.html

[22] JAWS. (2009) Java API for WordNet Searching. [Online]. http://lyle.smu.edu/~tspell/jaws/doc-1.2/overview-summary.html

[23] L. Limam, L. Brunie, D. Coquil, and H. Kosch, "Query Log Analysis for User-Centric Multimedia," in International Conference on New Media Technology, Graz, Austria, 2008, pp. 3-5.