ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Data Mining Techniques in User Profile Personalization

Robin Singh Bhadoria*
M.Tech (Software System)
Samrat Ashok Technological Institute (Govt. Autonomous)
SATI, Vidisha (MP)
ssr_robin@yahoo.co.in

Deepak Sain
M.Tech (Software System)
Samrat Ashok Technological Institute (Govt. Autonomous)
SATI,, Vidisha (MP)
dsain30@yahoo.in

Prof. Satendra Kumar Jain
Asstt. Professor,
Dept. of Computer Application,
Samrat Ashok Technological Institute (Govt. Autonomous)
SATI, Vidisha (MP)

Abstract: A particular user's expectation from his/her desktop or digital device is quite different from the expectations of a group of users for the same device. This implies that the behaviour of services and processes on the user's personal computing device to suit a user's preferences will not just ease the usage of these devices but also make them more sensitive to an individual's preferences giving an intuitive and natural interactive experience. In this work we look at two such areas: web search personalization and personalized organization of downloaded files on these personal devices. We use the documents of interest to the user and the user's download history to build a user profile. This profile is then used to re-rank web search results and identify preferred download locations. The results from experiments done in both areas are encouraging.

I. INTRODUCTION

The quantum of information is growing exponentially. This applies not just to the World Wide Web but also desktops, laptops, hand-helds and other equivalent devices whose storage capacities are growing at the same rate. Consequently, if users are to have the same degree of control over storage, organization and retrieval of information they need tools (assistants or bots) that help them by automating or simplifying the interaction of users with data. We must remember that the capacity of users remains roughly constant or at best increases very slowly. So, if information grows exponentially the tool must be able to store, organize, retrieve and process an exponentially growing entity to give the end user the same quality of experience.

One way to do this is to build a tool that uses all the historical usage information associated with a user to help a user to store, organize and retrieve information or data with much less actual interaction with the data. The generic name for this ability is personalization. Currently, personalization has been applied mostly to search on the web and to a few aspects of 'look and feel' which we shall refer to as surface personalization. There is clearly a need for deep personalization that responds intuitively and appropriately to a user's needs when the user interacts with data. This interaction, in general, is not just limited to search but includes storage and organization, though search is the dominant need. The personalization tool should be able to 'read a user's mind' to be able to do the right thing.

Personalization can be extended to recommend news/blogs related to a user's interests; augment the recent document list in document readers using favorite document lists; identify users with common interests for collaborative recommendation; use geographical location and information from the web to suggest events in the city that match a user's interests etc. To do all this and more, identification of a user's interest profile is necessary. Identifying the user's interest has other benefits as well.

A. Desktop Personalization

Desktop personalization involves customization of processes on one's personal device in order to enhance user experience. The idea is that a personal device should mirror the preferences and interests of the user. Currently, users customize their desktop in various ways. They change the user interface (desktop background, screen savers etc.) as per their liking, choose short cut icons as per their interest and convenience, add gadgets as per their requirements etc. These changes are superficial in nature as the real working of the device has not been personalized. Also, these changes require an effort on part of the user. What we are interested in is automatic personalization of some processes. Web search personalization is one of those. The way search results are ranked on one user's desktop should be different from the way they are ranked on another user's desktop if the two users in question have different interest profiles. In fact the way results are ranked on someone's desktop he/she uses in office should be different from what he/she gets on his/her desktop at home assuming different preferences at the two places. Similarly, to the best of our knowledge download location of a file is either fixed in a browser (set by user or a default location) or is given by the user every time he/she downloads a file. Some users do not organize their personal documents properly in directories. Desktop personalization aims at correcting all these by automatically learning a user's interests without significant effort from the user.

II. USER PROFILE

Any personalization system whose objective is to customize services for a user has to build a user profile. The

user profile has information that can distinguish one user from a multitude of other users.. Profiles normally include topics of interests but may also include topics of disinterest by taking into account relevant and non-relevant documents [6].

User profiles are generally built by giving appropriate weights to keywords that are deemed to represent a user's interests or by using weighted concepts from an existing ontology. On the other hand, a Weighted Concepts profile contains vocabulary which is large enough to represent a user's current and future interests. This is because first nodes from an existing concept hierarchy [7].are identified as user's interest based on some feedback which may be explicit or implicit. A proper subset of pages from those nodes (which have been manually classified earlier) are then fetched which results in a large vocabulary. However, the ontology should be built properly as it should represent correct relations between various concepts [6]. An example of concept hierarchy is shown in figure 2.1.

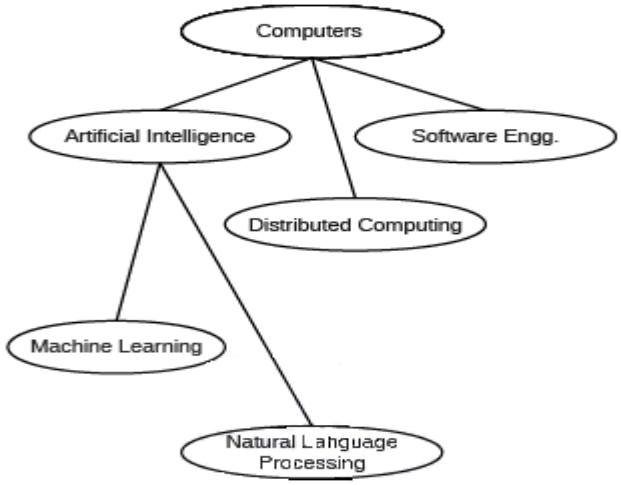

Figure 2.1: Subset of a Concept Hierarchy

Concept based profiles differ from keyword based profiles in that the features in the feature vector are concepts instead of keywords. For a system to identify general interests of user, a hierarchical profile is preferred to a flat one [12]. Here the feature, that is the concepts are given weights and these represent the user's interest in a particular concept. The levels in the reference ontology can be static or dynamic depending on a user's interests. Concept hierarchies are normally constructed from web directories like ODP [7], Yahoo [13] etc. Web directories were initially used for categorizing web pages and for help in navigating to the relevant page of interest.

Before the user profile construction a system needs to identify the interests of users. Our system uses the implicit feedback approach to decipher the interests of a user. The information sources currently used are:

a. The documents on the user's machine are taken as being of interest to him/her. The assumption is that if a user keeps a document on his/her machine there is a strong possibility that the user is interested in those documents.
b. The browsing history is another source of information. Although not all web pages browsed are of interest to the user but pages that are frequently visited and those on which more time is spent can be taken as useful documents for the user's interest.
c. The bookmarked pages are definitely interesting for a user.
d. The download history can be used as a starting point for deciding favorite directories for download. This would be more useful in case of those users who organize files on their machines properly.

III. USER PROFILE CONSTRUCTION TECHNIQUES

Information retrieval from text documents involves as a first step the pre-processing of text. These pre-processed documents then form the input for the various algorithms for information extraction.

These are the sources we have used in constructing a user's profile. Other sources of information which can be used are:

a. Emails read or created.
b. Bookmarks from a social bookmarking site.
c. Web communities.

- Social networking service.
- Web fore.
- Blogs of interest.

In this section we discuss the ***Vector Space Model*** which has been used in our system.

A. Vector Space Model

In this model the documents are represented as feature vectors where features are the keywords extracted from a given set of the documents. Each feature corresponds to an axis in this space. An example is shown in table 3.1. So, if we have an n- term document set (that is, there are n distinct terms in the vocabulary of that set of documents), the dimensionality can be pretty high (n) for even small document sets. If a term is more frequent in a document it is expected to be given more weight than a less frequent word as it represents the document better. In the term frequency model the value of a feature is term frequency divided by the total number of words in the document.

Table 3.1: An example feature vector from a document using Boolean Vector Space model. The feature space is <Murali, Warne, best, bowler, ever, wicket, player>,The feature vector is {1,0,0,1,1,1,0 }

Document	Murali	Warne	Wicket	Best	Bowler	Ever	Played
Murali is the best bowler ever	1	0	0	1	1	1	0

This normalizes the frequency for different document lengths. However, if some features occur throughout the set of documents they are not informative for the purposes of retrieval since their presence does not distinguish one document from another.

The solution to this problem is assigning tf-idf[1] scores to features. In the TF-IDF model [2], a feature is assigned a score which is a product of its term frequency and inverse document frequency of the same feature. This implies that the terms which represent the documents better get more weight-age and hence help to distinguish documents.

The term frequency tf_{ij} of a term t_i in a document d_j is defined as:

$$tf_{ij} = \frac{number\ of\ occurences\ of\ t_i\ in\ d_j}{Sum\ of\ number\ of\ occurrences\ of\ all\ terms\ in\ d_j}$$

The inverse document frequency of a term t_i in a set of documents D is defined as:

$$idf_i = \log\left(\frac{number\ of\ documents\ in\ \mathcal{D}}{number\ of\ documents\ \ in\ \mathcal{D}\ containing\ t_i}\right)$$

A TF-IDF score wij is the product of tfij and idfi. The weightage given is:

$$w_{ij} = tf_{ij} \times idf_i$$

These weights are then normalized to account for the length of documents:

$$wn_{ij} = \frac{w_{ij}}{\sqrt{\sum_{t_i \in d_j} w_{ij}^2}}$$

B. Clustering

The objective of clustering is to find common patterns, group similar objects, or to organize them in hierarchies. The grouping of similar objects should be such that members within a group are closer to each other than to members of a different group. In other words, the intra-cluster distance should be less than the inter-cluster distance. For personalization the objects to be organized are documents present on a user's personal device and the web documents obtained from a user's browsing history. Clustering algorithms can be flat or hierarchical depending on data description [3].We says that a data description is flat if the clusters are disjoint. But in many problems the clusters contain sub-clusters recursively.

The PDDP algorithm partitions the cluster C based on the principal direction vector which is the first left singular vector of the singular value decomposition (SVD) of the term-document matrix corresponding to that cluster.

Let $D_m = (d_1, d_2, ...d_m)$ be an n × m matrix where each d_i is the feature vector representing a document. Let $\mu_c = D_m e/m$ be the mean of $d_1, d_2, ...d_m$. Let u_c be the principal direction vector. Each document is projected onto u_c. The document is then assigned to one of the two clusters based on sign of $u^T (d_i - \mu_c)$. The PDDP algorithm starts with a single cluster and then partitions recursively based on the direction of projection on the principal direction vector corresponding to that cluster. This yields a binary tree with leaves corresponding to final clusters. At each stage the cluster to be partitioned is chosen on the basis of its 'scatter value' which is the measure of cohesiveness of the cluster. The 'scatter value' corresponds to the distance between each document in the cluster and the mean of the cluster. The cluster with the largest scatter value is chosen for splitting. The stopping criterion can be set based on scatter values or the desired number of clusters. One stopping criterion could be stopping when the maximum of the scatter values of leaf clusters (that is, the clusters yet to be split) falls below the scatter value of the cluster constructed by mean vectors of leaf clusters obtained so far.

a. Creation of the Profile

The Vector Space Model [2] is used to form clusters from the following sources:

i. Documents on a user's machine with textual content.
ii. Web pages from browsing history.
iii. Bookmarked web pages.

The documents used as input to the clustering module are of different types (pdf, doc, ppt, html etc). They are first converted to text and then preprocessed. The preprocessing step involves stop word removal and stemming. The Porter stemmer [22] has been used. These are then converted to feature vectors where the features are the terms in the documents after the preprocessing step.

The documents are then clustered divisively with the PDDP (Principal Directions Divisive Partitioning) algorithm [4], using the cosine similarity metric.

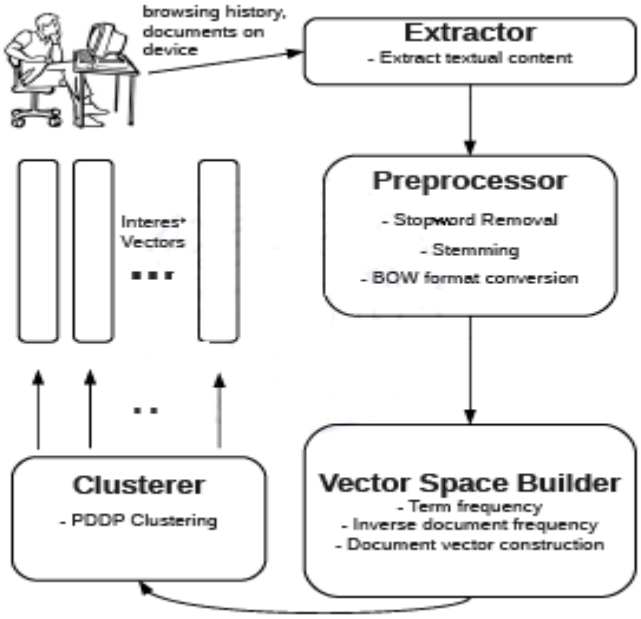

Figure 3.1: User Profile Creation

b. Updating the Profile

The recency windows used in the experiment ensure that a profile is updated as the user's interests change. Every two weeks the documents can be clustered again and the clusters would represent new interests. Between the updates we keep track of changing interests by following a simple intuitive idea. Suppose a user has 10 interests identified after clustering. Of these 10 interests he/she may be more interested in some and less interested in others. The interests which are more dear to a user would mean that the user browses or downloads more of such content in between updates. So if we assign weightages to interest vectors on the basis of documents downloaded and browsed we get a fairer representation of a user's current interest. As far as weightages are concerned they can be assigned proportional to the number of documents assigned to each cluster on the basis of the similarity metric.

IV. EVALUATION OF RESULT

This chapter describes the details of how a user's interests and preferences are used to personalize his/her search and desktop experience. Section 4.1 describes the design, evaluation and implementation details for personalization of web search.

A. Re-Ranking of Search Results

This module uses the user profile construction module (see 3(A)). The feature vectors representing user interests (called the interest vector) were created during profile construction.

During web search the results by the search engine are converted to feature vectors using the same pre-processing techniques that were used for the user profile. Thus each search result is represented by a feature vector. These feature vectors are then passed to Similarity Scorer which assigns them scores

based on their similarity to interest vectors. Each result is assigned a score equal to the maximum of the similarity scores with each interest vector. The results along with their scores are passed to Re-Ranker which sorts the results based on the scores assigned and the modifies the ordering that is ultimately presented to the user.

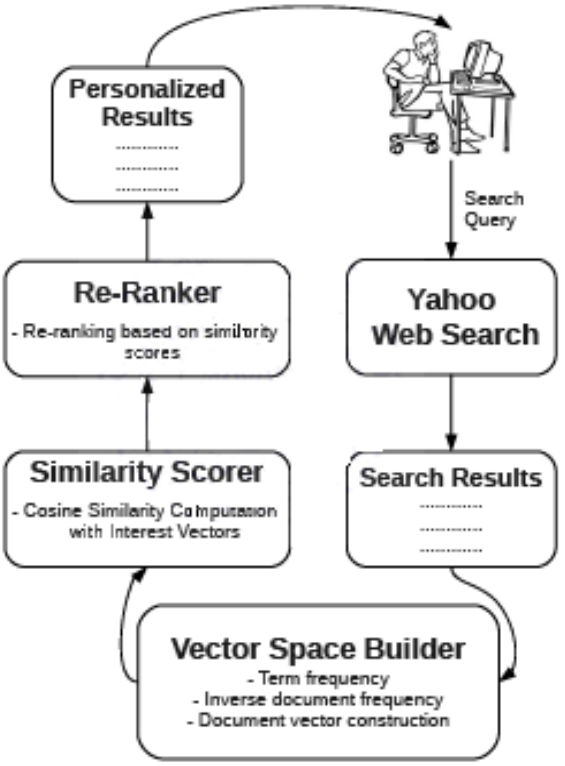

Figure 4.1: Re-ranking of search results

For evaluation of the rankings we used Discounted Cumulative Gain (DCG) [27] and Kendall Tau Distance [28]. The DCG measures the utility of the document in a ranked list based on its position in the list. Each result is associated with a 'gain' that is penalized by its ranking. For a list of length l, the DCG is defined as:

$$DCG(l) = relevance_1 + \sum_{i=2}^{l} \frac{relevance_i}{\log_2 i}$$

Where *relevance*i is the relevance assigned by the user to the ith result in the list.

Not all lists have the same length. Therefore, this value is normalized to obtain what is known as normalized Discounted Cumulative Gain (nDCG). The result list is sorted based on relevance scores and then its DCG is calculated. This is known as ideal Discounted Cumulative Gain iDCG (l). Since this list is partially ordered we used personalization ranks and then web ranks to break ties. The nDCG for a list of length *l* is given by:

nDCG(l) = DCG(l)/iDCG(l)

Kendall Tau distance (kt-Dist) is a metric which counts the number of inversions required to transform one list into another. Two lists are same if kt-Dist = 0 and reverse if it equals l(l – 1)/2 where l is the size of the list. The normalized Kendall Tau Distance (nkt-Dist) is obtained by dividing kt-Dist by l(l – 1)/2. So, nkt-Dist always lies between 0 and 1, both included.

Figure 4.2 shows the comparative normalized DCG values in 5 experiments. Table 4.1 reports the percentage increase in each experiment. The increase is significant ($p < 0.05$) in all experiments except 3. This shows the importance of having a rich user representation in identifying a user's interest(s). The maximum increase is obtained by using recent documents for re-ranking and whole index for constructing the vector space along with url boosting.

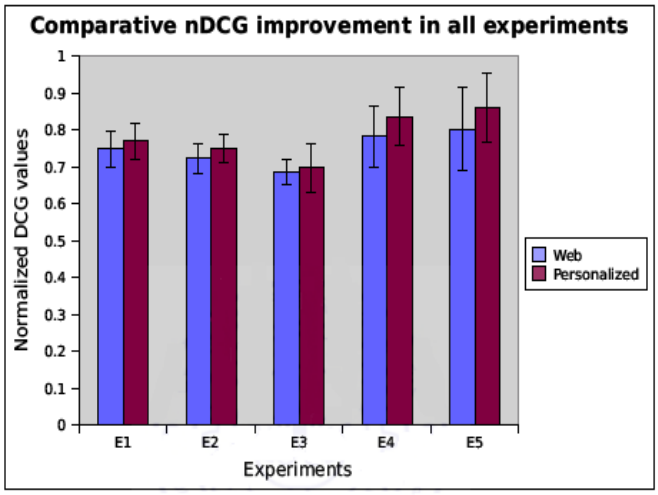

Figure 4.2: Normalized DCG improvement.

Table 4.1: % nDCG improvement in experiments

	nDCG_web	*nDCG_pers*	% improvement
Experiment 1	0.723	0.749	3.60
Experiment 2	0.749	0.771	2.98
Experiment 3	0.686	0.698	1.72
Experiment 4	0.783	0.836	6.79
Experiment 5	.803	0.860	7.07

The figure 4.3 shows average nkt-Dist among personalized search result list, web search result list and ideal ranking of search results. The nkt-Dist between personalized search result list and web search result list (0.520) which was found to be significantly ($p < 0.01$) higher than their individual distances from ideal ranking (0.252,0.266).

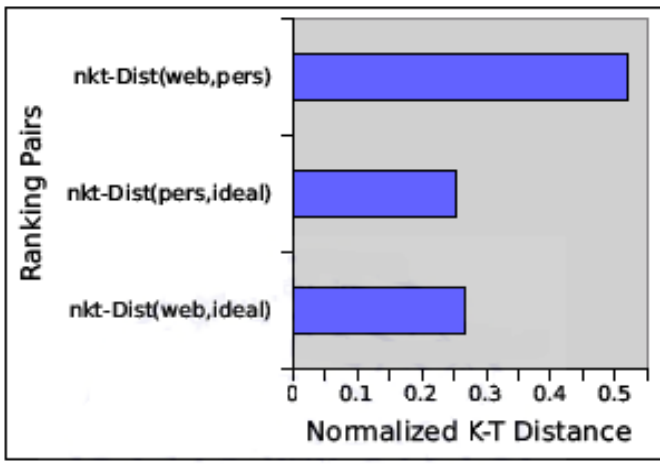

Figure 4.3: Comparison of Normalized KT Distance between rankings.

The Kendall Tau distance [28] between personalized list and web list (nkt-Dist (pers,web)) is significantly ($p < 0.01$) higher than their individual distances from ideal ranking. This reinforces the results from Teevan et al [20] that things which make these two lists (web and personalized) relevant are different. There is large scope for improvement in performance by combining the two rankings properly.

V. CONCLUSION

Two of the most common operations while using a digital device are: search and download. In this thesis we have built a prototype system that uses interest based profiles to personalize the operations of search and download location identification so that they give a more natural and intuitive interactive experience. Initial user based experiments show that at least some ways of personalization work quite well.

For web search personalization we conducted various experiments differing in the way a user's profile was created and search results represented. We found that the best performance was obtained when the short term history was used for disambiguation and long term history was used for constructing the feature space. The Kendall Tau distance [28] between the web search list and personalized search list reinforces the view of Teevan et al. [20] that a lot of improvement can be obtained by properly combining the two rankings.

The identification of preferred download locations worked very well with url and domain based decision schemes. The content based decision scheme while not as good as first two, worked well in some cases where similarity was more obvious. The type based scheme for non-textual content did reasonably well. Overall, the combination of these schemes led to an enjoyable user experience in file downloads.

VI. REFERENCES

[1] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Characterizing the value of personalizing search. In SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 757-758, New York, NY, USA, 2007. ACM.

[2] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Commun. ACM, 18(11):613-620, 1975.

[3] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern classification, new york: John wiley & sons, 2001, pp. xx + 654, isbn: 0-471-05669-3. J. Classif., 24(2):305-307, 2007.

[4] Daniel Boley. Principal direction divisive partitioning. Data Mining Knowledge. Discovery 2(4):325-344, 1998.

[5] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, pages 515-524, New York, NY, USA, 2002. ACM.

[6] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. Pages 54-89, 2007.

[7] The Open Directory Project (ODP). http://dmoz.org.

[8] Alexandros Moukas and Pattie Maes. Amalthaea: An evolving multi-agent in- formation filtering and discovery system for the www. Autonomous Agents and Multi-Agent Systems, 1(1):59-88, 1998.

[9] Matthew Montebello, W. A. Gray, and Stephen Hurley. Evolvable intelligent user interface for www knowledge-based systems. In IDEAS, pages 224-233, 1998.

[10] Dwi H. Widyantoro, Jianwen Yin, Magy Seif El Nasr, Linyu Yang, Anna Zacchi, and John Yen. Alipes: A swift messenger in cyberspace. In Spring Symposium on Intelligent Agents in Cyberspace, pages 62-67, Palo Alto, March 1999.

[11] Extend Ed, H. Sorensen, and M. Mc Elligott. Psun: A profiling system for usenet news.

[12] Eric Bloedorn, Inderjeet Mani, and T. Richard MacMillan. Machine learning of user profiles: Representational issues. CoRR, cmp-lg/9712002, 1997.

[13] Yahoo! Directory. http://dir.yahoo.com.

[14] Alexander Pretschner and Susan Gauch. Ontology based personalized search. In ICTAI '99: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, page 391, Washington, DC, USA, 1999. IEEE Computer Society.

[15] Micro Speretta and Susan Gauch. Personalized search based on user search histories. In WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pages 622-628, Washington, DC, USA, 2005. IEEE Computer Society.

[16] F. Tanudjaja and L. Mui. Persona: A contextualized and personalized web search. In HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3, page 67, Washington, DC, USA, 2002. IEEE Computer Society.

[17] Ahu Sieg, Bamshad Mobasher, and Robin Burke. R.: Inferring users' information context from user profiles and concept hierarchies. In In: Proceedings of the 2004 Meeting of the International Federation of Classification Societies, IFCS 2004, pages 563-574, 2004.

[18] J. M. Gowan. A Multiple Model Approach to Personalised Information Access, 2003. Master Thesis in computer science, Faculty of Science, University College of Dublin.

[19] Susan Gauch, Guijun Wang, and Mario Gomez. Profusion: Intelligent fusion from multiple, distributed search engines. Journal of Universal Computer Science, 2:637-649, 1996.

[20] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 449-456, New York, NY, USA, 2005. ACM.

[21] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. Inf. Process. Manage, 36(6):779-808, 2000.

[22] M. F. Porter. An algorithm for sufx stripping. Pages 313-316, 1997.

[23] Apache PDFBox - Java PDF Library. http://pdfbox. apache.org/.

[24] Apache Tika - Content Analysis Toolkit. http://lucene. apache.org/tika/.

[25] Omniclusterer clustering library. http://www.tcllab.org/ canasai/software/omniclusterer/.

[26] Yahoo! Developer Network. http://developer.yahoo.com/search/.

[27] Kalervo J¨rvelin and Jaana Kek¨l¨inen. Ir evaluation methods for retrieving highly relevant documents. In SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 41-48, New York, NY, USA, 2000. ACM.

[28] M. G. Kendall. A new measure of rank correlation. Biometrika, 30(1/2):81-93, 1938.

[29] Dekang Lin. An information-theoretic definition of similarity. In ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, pages 296-304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.