A Tool to Personalize the Ranking of the Documents Returned by an Internet Search Engine

Wadee S. Al halabi, Miroslav Kubat, Moiez Tapia
Department of Electrical and Computer Engineering, University of Miami
w.alhalabi@umiami.edu, mkubat@miami.edu, mtapia@miami.edu

Abstract

Internet search engines identify web pages that contain user-specified keywords, and then rank these pages according to their (heuristically assessed) relevance to the user's query. In this paper, we investigated the possibility of evaluating this relevance by the similarity of the returned web page to web pages previously visited by the same user: these previously visited pages thus serve as positive training examples from which a machine-learning program induces an internal model of the user's interests and preferences. We describe two different ways to represent this model. Our experiments indicate that this approach can indeed improve the ranking.

1. Introduction

Upon receiving a user's query in the form of a list of keywords, a typical Internet search engine identifies those web pages (documents) that contain the query keywords, and then estimates the relevance of these documents to the user's query. The engine then offers the user a list of hyperlinks pointing to these documents, ordered according to the relevance to the query. Historically, a document's relevance to the user's needs has been calculated from (1) the frequencies of the keywords in different parts of the documents, (2) the popularity of these documents (measured by the time spent on them by an average user), and (3) the structure of the links to and from related web pages.

In the research reported here, we build on the obvious fact that each user has somewhat different interests that are to a great degree reflected by the contents of the web pages he or she has visited in the past. Alternatively, the user may want to indicate these preferences by one or more web sites that he or she deems relevant. Building on this assumption, we have developed a tool that accepts the hyperlinks obtained by submitting the user's query to an off-the-shelf search engine, downloads the corresponding documents, and then uses the knowledge induced from documents previously visited by the same user to calculate the weights to be assigned to the returned documents.

Finally, the system re-orders the returned documents according to these weights, and returns to the user the new ranking instead of the one recommended by the original search engine.

To justify the hypothesis that the new ranking better reflects the user's needs, we have conducted several experiments with two alternative ways to exploit the user-supplied reference documents. From these, our own LVA algorithm appears to outperform the classical VSA technique known from the discipline of information retrieval both in the quality of ranking and in the computation expenses.

2. Previous Research

In the past few years, several research groups have studied various methods to personalize the outputs of search engines. Some of these groups employed machine learning techniques. To establish the context for our own research, let us briefly summarize this earlier work.

Fan et al. [1] developed a system that personalizes its output, and demonstrated that the personalization improved its utility. Following this idea, literature reports several attempts to improve the behavior of information-retrieval systems by the use of machine-learning techniques. One possibility is reported by Boyan et al. in [2] who use an induction algorithm to find a way to assign weights to the returned documents. These weights are based on the location of the keywords in the document and are used for the re-ordering of the documents. Cui et al. [12] improve the quality of query processing by analyzing the search engine's logs of previous queries (submitted by the same user). Likewise, Haveliwala et al. [3] investigate the possibility of finding a Web page relevant to a reference Web page. In their approach, the reference document represented a positive example (they did not use negative examples).

Poincot et al. in [4] introduced a new approach to compare documents and calculate their similarity by way of Kohonen's neural. Chakrabarti et al. [5] present a system that mines the web by the hub-and-authority's technique. Ahonen et al. [7] experiment with co-occurring text phrases and report promising experimental results.

Liu et al. [9] implemented a machine learning algorithm that learns how to represents the user's interests. Every time the user connects to a URL, the system categorizes this URL. Some more evidence that machine learning can improve web mining and information retrieval undertakings is provided by [1, 2, 4, 5, 6]. Interesting work was reported by Müller et al. [11] who also use a multi-agent machine learning approach to learn how to filter out irrelevant Web pages. Lin et al. [15] explore the documents before the searching process starts; their algorithm then categorizes the documents according to the knowledge induced from these documents. Focusing on the use of multi-agent systems, Perez et al. [10] discuss the advantages of the parallelism in information retrieval system and explore ways to optimize the cooperation and coordination among agents.

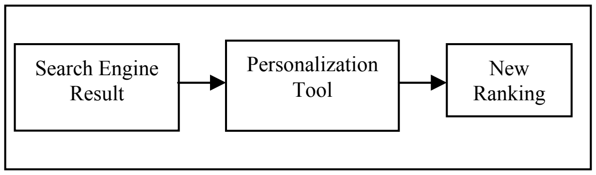

Figure 1. The place of the personalization tool in query processing.

3. Proposed Solution

Let us now describe our own web-search personalization system that we will refer to by the acronym PT (*Personalization Tool*). The input to PT is two-fold: (1) a user-submitted "preferred document" and (2) the output of an off-the-shelf search engine (a set of URLs) obtained in response to the user's query. The hyperlinks (URLs) are then used to download the textual data at the corresponding web pages (images and videos are ignored). The system uses the preferred document to reorder the returned documents.

As indicated, our system uses a simple induction technique to find out how to assess the relevance of a document to the user's needs. The training phase uses positive and negative examples that have been labeled as such by the user that has scanned the output of the search engine. The positive ones (add_files) are those that the user believes have a high degree of similarity to his or her query. The negative examples (remove_files) are those that the user insists are irrelevant to the query.

The Vector Space Algorithm (VSA) [13, 17] is widely used in information retrieval. Let us denote by *W(u,S)* the weight of term *u* in document *S*, and let us denote by *W(u,Q)* the weight of term *u* in query *Q*. Then, the similarity between S and Q is calculated using this following formula:

$$\mathrm{Cos}(S,Q)=\frac{\sum_{u\in S\cap Q} W(u,S).W(u,Q)}{\sqrt{\sum_{u\in S} W(u,S^2).\sum_{u\in Q} W(u,Q^2)}}\,; \qquad (1)$$

The Linear Vector Algorithm (LVA) was introduced [18] as a simplified and much faster alternative to VSA. Let *D* denote the number of occurrences of a given term in a retrieved document, and let *R* denote the number of occurrences of this word in a reference document. The weight of this term is then established using the following formula:

$$\mathrm{Weight}=\frac{D}{R} \quad \text{if} \quad \frac{D}{R}\le 100,$$

$$\mathrm{Weight}=100-\left(\frac{D}{R}-100\right)*0.1$$

$$\text{if } \left(100<\frac{D}{R}<1100\right),$$

$$\text{Otherwise Weight} = 0; \qquad (2)$$

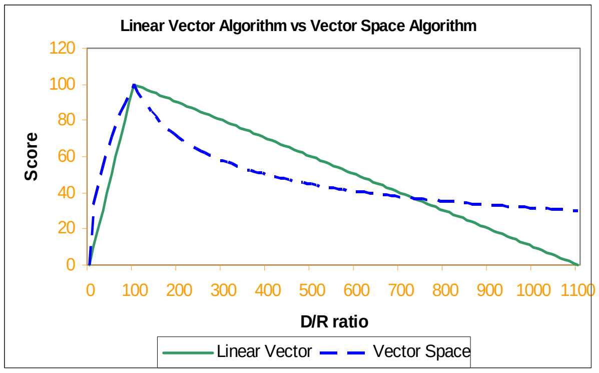

Figure 2. LVA vs. Vector Space behavior

Figure 2 illustrates the difference between the two approaches by showing the weight ("score") they assign to a given term for different values of the *D/R*-ratio. The reader can see why we call our formula "linear," and denote it by LVA. The linearity may be regarded as a crude simplification. On the positive side, it is much faster to calculate, and its lower flexibility makes it less likely to overfit noisy training data.

4. Evaluation methodology

The performance of information retrieval algorithms is usually evaluated by the formulas for *precision* and *recall*. However, these criteria are inadequate for our needs because they fail to reflect the quality of ranking. This is why we have developed an alternative approach [19] that we dubbed a Search Engine Ranking Efficiency Evaluation Tool (SEREET) which was introduced by us [19]. To make this paper self-contained, let us briefly revise this formula.

The idea is to evaluate the quality of an ordering based on the knowledge of the correct ordering. Let *Wi* denote the weight of the *i*-th document in the "correct" list and let *n* be the number of documents in the returned list. The value of SEREET is calculated as the follows:

$$\text{SEREET} = \frac{\sum_{i=1}^{n} Wi}{n} * 100\% \qquad (3)$$

When calculating the weights, the algorithm starts at 100 points in the top of the rank and deducts points each time a relevant document is not found in the returned list (is missing). If the returned list of documents contains an irrelevant document, we regard this also as a "miss," while understanding that the terminology is not perfect. In principle, we consider a relevant web page as a hit, and an irrelevant one as a miss. The following formula is used to calculate the weights of the document in the returned list:

$$W_i \begin{cases} = 100 & \text{if top of the rank.} \\ = 100 & \text{if no miss is found} \\ = \text{p.p.} - \dfrac{\# miss}{RankLength} * 100\% & \end{cases} \qquad (4)$$

Where p.p. : previous points.
RankLength: # of links in the rank list

Example: Suppose that we know the ideal ordering is [a,b,d,e,f,g,h,i],, while the search engine returned ordering [a,b,e,g,i,j] The above approach will then return the following weights:

W_a = 100
W_b = 100 - 0
W_e = 100 – 20 = 80
W_g = 80 – 10 = 70
W_i = 70 – 10 = 60
W_j = 60 – 0 = 60

Therefore,

$$\text{SEREET} = \frac{100+100+80+70+60+60}{10} * 100\% = 47\ \%$$

5. Experimental results

We prepared a few test-beds, each defined by a different user query. For the queries, we chose such keywords that could be expected to lead to multiple categories. For instance, by submitting the keyword **jaguar,** a search engine is likely to return documents related to such topics as cats, cars, automobiles, etc. Likewise, the keyword **research fund** may return documents related to different scientific disciplines such as engineering or biology.

To start with, let us look at our system's response to the keyword **research fund.** The user is interested in "cancer research" and "medical field" in general. Therefore, we labeled previously visited web page containing these two words as positive examples. To these, we added some negative examples as documents that fail to contain at least one of these two terms. Having a sufficiently large pool of examples to choose from, we gave preference to those that displayed high degree of relevance/irrelevance.

In the first experiment, we wanted to find out how LVA's ability to predict the user's perceived relevance of the returned documents depends on the number of the training examples submitted to the PT-system. To this end, we ran the system several times, each time on a training set of a different size. To be more specific, we first considered training sets that contained N positive examples and N negative examples for N growing from 1 to 10.

The chart in Figure 3 summarizes the results, quantifying the ranking performance by the error rate as obtained using the SEREET criterion described above. The reader can see that, expectedly, the error rate tends to drop with the increasing size of the training set. When only one pair of training examples (one positive and one negative) was used the accuracy of PT was unimpressive, indicating that larger training set is needed. On the positive side, the number of examples needed for the system's performance to achieve reasonable levels does not appear to be a high is typical for the field of machine learning.

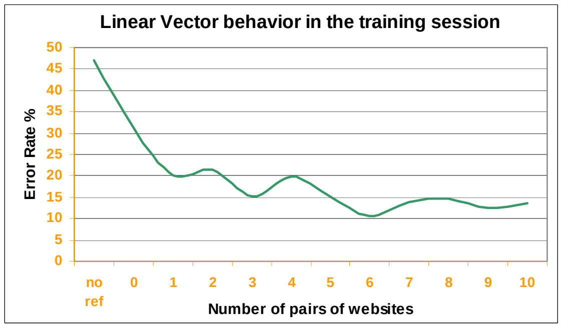

Figure 3. Linear Vector Responses to the Training Process

In our next experiment, we wanted to compare (in the same domain) LVA's performance with that of the more traditional (and more computationally expensive) VSA. Figure 4 summarizes the results obtained using the same training data as in the previous experiment. Again, the error rate was calculated using the SEREET mechanism described above. The reader can see that the LVA approach outperforms. While both algorithms manifest the same error rate in the absence of reference pages, their performance improved with the growing size of the training set. Importantly, the LVA approach appears to be much more accurate in predicting the correct ranking than VSA that has comparably high error rate. The two curves tend to converge only for larger training sets.

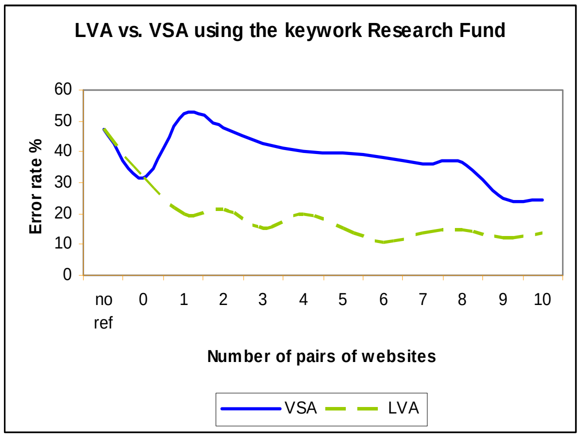

Figure 4. LVA vs. VSA using the keyword research fund for cancer research in the medical field

In another experiment, we used the keyword **jaguar** and regarded as positive those documents that were related to an animal (rather than, say, a car). Figure 5 shows how an increased number of training examples leads to higher ranking performance (along the SEREET criterion). Again, LVA tend to learn faster.

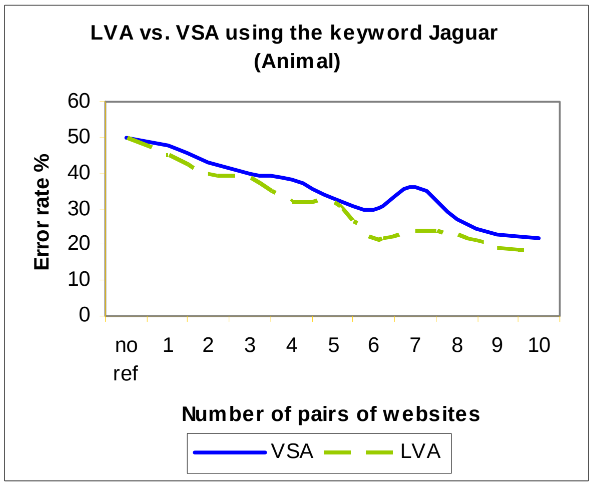

Figure 5. LVA vs. VSA using the keyword Jaguar (Animal)

In the final experiment, we used the keyword **beetle**. Figure 6 illustrates ranking performance. In all experiments, we found that LVA can learn much faster than the VSA.

6. Conclusion

The paper presented, and experimentally evaluated, two simple methods for the personalization of search engine outputs. In particular, we compared our own LVA approach with the more traditional VSA, and showed how their ability to rank documents by their relevance to the user's personal preferences depends on the number of the training examples. From the two techniques, the computationally less expensive LVA approach shows higher ability to rank the documents according to the user's preferences.

We conclude that, when used properly, our system can gradually "learn" to reflect the needs of the user. In the absence of any previous "experience," the system behaves as a traditional search engine. However, with the growing number of positive and negative training examples, the system becomes much more adept at giving preference to those documents that the user is likely to find interesting.

References

[1] W. Fan, M.D. Gordon, P. Pathak, Personalization of search engine services for effective retrieval and knowledge management, in the Proceedings of the 2000 International Conference on Information Systems (ICIS), 2000, Brisbane, Australia.
[2] Boyan, J. A., D. Freitag and T. Joachims. "A Machine learning architecture for optimizing web search engines." *Proceedings of the AAAI workshop on Internet-Based Information Systems*, AAAI Technical Report WS-96-06, 1996.-
[3] T. Haveliwala, A. Gionis, D. Klein, and P. Indyk. "Evaluating strategies for similarity search on the web," In Proceedings of the Eleventh International World Wide Web Conference, May 2002.
[4] P. Poinçot, S. Lesteven, and F. Murtagh. "Comparison of two document similarity search engines," Library and Information Services in Astronomy III, ASP Conference Series, Vol. 153, 1998
[5] S. Chakrabarti, B. Dom, D. Gilbson, J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Mining the link structure of the world wide web," IEEE Computer, 32(8): 60-67, 1999.
[6] D. Mladenic. " Text-learning and related intelligent agents," IEEE Intellegent Systems, 14 (4): 44-54, 1999.
[7] H. Ahonen, O. Heinonen, M. Klemettinen, and A. Verkamo. "Finding co-occurring text phrases by combining sequence and frequent set discovery." In R. Feldman, editor, Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Fundations, Techniques and Applications, page 1-9, 1999.
[8] T. Haveliwala. "Topic-sensitive Pagerank: a context-sensitive ranking algorithm for web search," IEEE Transaction on knowledge and data engineering, Vol 15, No. 4, Page 784-796, Jul-Aug 2003.
[9] F. Liu; Y, C.; W. Meng. "Personalized web search for improving retrieval effectiveness," IEEE Transaction on knowledge and data engineering, Volume: 16, Issue: 1, page 28-40, Jan. 2004.
[10] Perez, M.S. Garcia, R. Carretero, J. "MAPFS-MAS: a model of interaction among information retrieval agents," Proceedings of the Cluster Computing and the Grid 2nd IEEE/ACM International Symposium CCGRID, May 2000, pp.248 - 249
[11] Martin E. Muller, "An Intelligent multi-agent architecture for information retrieval from the Internet."
[12] H. Cui; J.Wen; J. Nie; Wei-Ying Ma, "Query expansion by mining user logs," IEEE Transactions on Knowledge and Data Engineering, page(s): 829- 839, July-Aug. 2003.
[13] R. Baeza-Yates, B. Ribiero-Neto, "Modern information retrieval," Addison-Wesley Pub Co, 1st edition (May 15, 1999).
[14] M.W. Berry, M. Browne, "Understanding search engines: mathematical modeling and text retrieval," Society for Industrial & Applied Mathematics, (July 1999).
[15] S.Lin; M. Chen; J. Ho; Y. Huang, "ACIRD: intelligent Internet document organization and retrieval," IEEE Transactions on Knowledge and Data Engineering, Volume: 14, Issue: 3, Page(s): 599-614, May/Jun 2002.
[16] R. S. Michalski, I. Bratko , M. Kubat, "Machine Learning and Data Mining: Methods and Applications" John Wiley & Sons; (April 9, 1998).
[17] G. Salton, A. Wong and C. S. Yang, "A Vector Space Model for Automatic Indexing," Communication of the ACM, Volume 18 , Issue 11, Pages: 613 – 620, November 1975
[18] W. Al Halabi, M. Kubat, M. Tapia "Search Engine Personalization Tool Using Linear Vector Algorithm" the proceedings of the 4th Saudi Technical Conference. Dec 2006.
[19] W. Al Halabi, M. Kubat, M. Tapia "A numerical efficiency evaluation tool" the proceedings of the 7th IBIMA Italy, Dec 2006.

Wadee Alhalabi, is a PhD candidate at the University of Miami, Head of the Computer Technology Department at the College of Technology-Makkah and a Lecturer in the Department of Computer Science at the University of Umm Alqura,

Miroslav Kubat is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Miami. His research interests are in the field of machine learning, data mining and artificial neural networks.

Moiez Tapia is an Full Professor in the Department of Electrical and Computer Engineering at the University of Miami. His research interests are in the field of Multi-valued logic and calculus, real-time systems, machine learning, fault-tolerant computing.