Vectorial Information Structuring for Documents Filtering and Diffusion

Omar Nouali and Abdelghani Krinah
Basic Software Laboratory, CERIST, Algeria

Abstract**:** *Information retrieval tries to identify relevant documents for an information need. The problems that an IR system should deal with include document indexing (which tries to extract important content from a document), user needs analysis (similar to document indexing but applied to a query), and their internal representation which makes them suitable for being explicitly manipulated by the corresponding algorithms (i.e., matching the query with the documents). This paper describes a vectorial approach for information organization, and its application to search/retrieval systems from a vast amount of textual data.*

Keywords**:** *Vectorial information structuring, information filtering, terms classification, documents diffusion.*

Received March 7, 2006; accepted June 6, 2006

1. Introduction

Information retrieval system objective is to offer information meeting at best the user's requirements, while having less possible interaction with him? In other terms, it's about solving the duality "results relevance - research cost", that is being in center of the problems of information retrieval on Internet.

Indeed, the tendency of documentary computing today is to be centered on the user's status to filter, adapt, personalize its research all while discharging him from responsibility to guide it, so he can concentrate its efforts, either on access or acquirement of information, but especially on consumption or utilization of this one, that was to be his starting objective. That is called 'passive approach' adopted by tools such as mailing lists.

However, transmitted information is sometimes less relevant than those of classical search engines where the need of powerful filtering taking into account the specificities of the final recipient. The question is to find a documents structuring shape which allows, on one hand to preserve at best their informational content and on other hand to make them usable by the various treatments to be applied on.

This study presents a vectorial data structuring of both content (documents corpus) and needs (users profiles), and how it will improve the quality of information filtering carrying by our retrieval system [13] with the aim of diffusing only relevant content.

2. State of Art

Document filtering, also known as Selective Dissemination of Information (SDI) has a long history, most of it based on the unranked Boolean retrieval model [14]. A user's information need is expressed by a query. Queries are expressed with Boolean logic.

A query either matches or does not match a document. There is no ability to partially satisfy a query (for example, see LMDS system [19]). This model adopted by current search engines brought only partial solution to this problem; it especially succeeded in reducing the research index to a list of results classified according to certain criteria of relevance. What already constitutes a considerable progress taking into account initial size of the documentary base which is whole Internet.

Nevertheless, "works of Jones [9], Spink [17] and Bruza [3], on the use of search engines showed that the system resources are under-utilized and that the tools offered to the final user to explore the high number of answers are insufficient and unsuited" [1]. Therefore, it is generally accepted that statistical systems provide better performances for document retrieval than do unranked Boolean systems [4]. The growing power of computer hardware has made statistical systems increasingly practical for even large scale document filtering environments. A common approach has been to simulate document filtering with an existing vector-space or probabilistic document retrieval system on a collection of new or recent documents (e.g., most TREC systems [8]). This approach is simple, effective, and has the advantage of a corpus from which to gather statistics like *idf* [4].

3. Information Organization

The underlying model for our approach is the vector space. This model has been developed in the 1980's to enhance electronic information retrieval, see [15]. Documents are given an extensional, vectorial

representation, in which dimensions of the vector representing a document are the terms occurring in. In the vector space model, a corpus of texts (documents) is transformed into a term-document matrix, displaying for each term its occurrence frequency in each document. Hereby, each term can be defined as a vector of its occurrences in the document collection [18]. Documents are given an extensional, vectorial representation, in which dimensions of the vector representing a document are the terms occurring in.

First, a preprocessing is necessary to prepare the documents to following filtering steps. It consists in eliminating common words (articles, prepositions, etc.) using a list of "stop words". Then, it is about reducing morphological variants to a common form (often called *lemma*). For that, a set of lemmatization rules are applied on the various forms to return verbs into infinitive, remove plural forms, etc. During this phase, the recognition of multi words (i.e., terms consisting of more than one word) is also performed [12]. Following this preprocessing step, document is given to an analyzer, that aim is to identify and extract words or terms that best characterize the contents of each document using a statistical weighting process. Therefore, the distance function so far has been largely defined and used ad hoc, usually by a *tf.idf* weighting scheme [16], where two intuitions are at play [10]:

- The more frequently a term T_k occurs in a document D_j, the more important for D_j it is (the term *frequency assumption*).
- The more documents a term T_k occurs in, the smaller its contribution is in characterizing the semantics of a document in which it occurs (the inverse document frequency assumption).

The version of *tf.idf* used in this study is a combination of both term frequency and inverse document frequency assumptions; it is given by the formula below:

$$Tf.idf\ (t_k, d_j) = tf\ (t_k, d_j).\ (log\ |C| / \#c(t_k)\ ) \quad (1)$$

where $\#c(t_k)$ denotes the number of documents in the collection $|C|$ in which term t_k occurs at least once.
As a result of this preprocessing, the document is conceptually represented by a K dimensions vector space:

$$M = \{(T_1, W_1), (T_2, W_2, \ldots, (T_k, W_k)\} \quad (2)$$

Where T_i represents the i^{th} term, W_i the weight (calculated by *tf.idf* formula) and k the terms space.
As shown by Figure 1, the matrix structure of corpus is obtained by gathering all documents which constitute it.

Profiles are also organized in a vectorial way. User modeling is first based on the needs he expresses himself during inscription step. Unlike most current mailing lists that impose their subscribers to select among a list of given topics, here the user has a greater liberty in choice of the keywords that interest him as he would do with common search engines. Terms subjected (simple and compounded ones) are then used to build the profile vector. Moreover, we use a thesaurus that allows improving user's representation by implying other words; even if they did not appear explicitly in the original submitted keywords list.

Corpus	Java	Oracle	Base de donee	SQL Server	MYSQL	Oracle Light
Doc1	0	0	0	0.03571429	0	0
Doc2	0	0	0	0.03571429	0.25	0
Doc3	0	0	0	0.03571429	0.25	0
Doc4	0.2	0	0	0	0	0
Doc5	0	0	0	0.03571429	0	0
Doc6	1	0.42857143	0	0	0	0
Doc7	0	0	0.666666667	0	0	0
Doc8	0.2	0	0	0.32142857	0	0.25
Doc9	0	0.14285714	0	0	0	0
Doc10	0	0	0	0	0	0
Doc11	0.4	0.42857143	0	0	0	0
Doc12	0	0	0	0.03571429	0	0
Doc13	0.2	0	0	0.25	0	0

Figure 1. Matrix corpus representation.

Having an already existing term T_1 (in profile), with a P_1 weight, and having the following entry in the thesaurus (T_1, T_2, S_{12}) where T_1 and T_2 are two semantically close terms whereas S_{12} represents their degrees of similarity. So T_2 will be added to the profile and its P_2 weight will be the multiplication of P_1 and S_{12}. This with an aim of giving the most importance to the keywords chosen by the user, then those which are directly linked to them, and so on, since the multiplication of a number by an index ranging between '0' and '1' (which is case of the various existing weights) will decrease its value. The following example will illustrate the process in more explicit way:

A user describes him as being interested by the Databases Management System (DBMS). Consequently this keyword will be seen allotting a weight equal to '1' in the profile, which will contain only one term {(DBMS, 1)}. Considering the thesaurus, one finds that the keyword 'DBMS' is related to the term 'database' with '0.85' as degrees of similarity. Therefore, the profile will be extended by keyword 'database' and weight value's 1 X 0.85 = '0.85', to become {(DBMS, 1), (database, 0.85)}. Moreover, 'database' co-occurs in the thesaurus with the term 'SQL' with '0.40' as index of similarity. So, keyword 'SQL' will be added to the profile and its weight will have the value '0.85' X '0.40' = '0.34'. Finally, the profile will be represented by the following vector:

$$P = \{(DBMS, 1), (database, 0.85), (SQL, 0.34)\} \quad (3)$$

In the same way as corpus, profile matrix is made by assembling all existing user's vectors, as shown in Figure 2.

It is to be specified that profiles extension is made in a dynamic way, during the filtering operation which is done on copies dynamically created.

Noms	Java	Base de donnes	Oracle	Servlets	SQL	Object Oriented
User1	1	0	0	0.58	0	0.33
User2	0	1	0.38	0	0.73	0
User3	1	1	0.38	0.58	0.73	0.33
User4	0	0.38	1	0	0.24	0
User5	0.33	0	0	0	0	1
User6	1	0	0	1	0	0.83
User7	0	1	1	0	0.78	0
User8	0	1	0.38	0	1	0

Figure 2. Matrix profiles organization.

This is done in order to preserve the original contents of profile as described by their owners, whom only are authorized to reach and modify it according to results returned to them, in case they are considered to be not exactly in conformity with their weight.

What follows will confirm the effect of the proposed matrix organization in document retrieval task, especially in term's classification and document's filtering tasks managed by our software.

4. Applications

4.1. Terms Classification

Classification operation consists of grouping terms into specified classes according to their meaning, with an aim of automated thesaurus generation. Basic idea is that, generally, the definite concepts to represent a profile are not inevitably the same ones extracted from the documents. For that, we propose the use of a thesaurus that improves the document representation by taking into account the terms that belong to the document and do not exist in the profile and vice versa; it's about replacing them by semantically closer terms. The underlying metaphor is that term semantics is conveyed by the terms that co-occur with it, i.e., that occur in the same documents [10].

We build a squared term-term matrix having as many rows and columns as terms in our vocabulary. After training on a large corpus of general language text, each of its cells displays the frequencies of co-occurrence of one term with another [18]. This operation is performed using a multidimensional statistical method applied to document's text words, and called analyzes factorial of correspondences [2].

By this technique, each dimension of the corpus matrix (contingency table) makes it possible to define distances (or proximities) between the elements of other dimension. From this table of distances, we obtain a geometrical representation describing the similarities between the lines (documents) or the columns (terms) [7].

Regarding our objective which consists of terms classification, we'll proceed to distance calculation between column vectors which represent the semantic similarity's degree between corresponding terms. Two terms similarity calculation is measured with cosine formula that calculates the cosine of angle between their respective vectors.

$$COS(T_i, T_j) = \sum_n T_{ni} * T_{nj} / \sqrt{\sum_n T_{ni}^2} * \sqrt{\sum_n T_{nj}^2} \quad (4)$$

where:

$COS(T_i, T_j)$ is the cosine or similarity degree between terms T_i and T_j, and T_{nk} is the weight of T_k term in document n.

The closer angle cosine between the two vectors is to 1, the more the vectors are close what implies a greater resemblance between the two terms. As a result, terms and distances separating them are represented in shape of symmetric square matrix $C(k*k)$ where C_{ij} represent resemblance degree between i^{th} and j^{th} terms of the original corpus as illustrated by example, as shown in Figure 3.

Once cosines are calculated, classification operation begins itself, namely gathering into the same class terms that similarity's degree exceeds a certain threshold dynamically fixed. Associations (between terms belonging to the same class) obtained will finally be used to enrich the thesaurus.

4.2. Documents Filtering

Filtering operation consists in comparing each profile with the various corpus documents. The documents' filtering adopts the same principle as the terms classification, namely the use of a textual data analysis statistical method; i.e., euclidean cosine distance function. However, the required goal is not to calculate similarity between tow corpus terms, but between individual from the users-profile and

	Collect	Database	DBMS	Filtering	Internet	Mobile	Protocol	Network	WAP
Collect	1	0	0	0.92	0.47	0	0.12	0.17	0
Database	0	1	0.89	0	0.13	0	0.17	0.31	0.25
DBMS	0	0.89	1	0	0	0.29	0.41	0.43	0.12
Filtering	0.92	0	0	1	0.64	0	0.19	0.38	0
Internet	0.47	0.13	0	0.64	1	0.57	0.83	0.92	0.61
Mobile	0	0	0.29	0	0.57	1	0.32	0	0.87
Protocol	0.12	0.17	0.41	0.19	0.83	0.32	1	0.46	0.77
Network	0.17	0.31	0.43	0.38	0.92	0	0.46	1	0.43
WAP	0	0.25	0.12	0	0.61	0.87	0.77	0.43	1

Figure 3. Term's proximities representation.

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7
User1	0.745356	0.0	0.28284273	0.43386093	0.16666667	0.0	0.6499337
User2	0.2860388	0.77459663	0.20519567	0.4	0.28867513	0.0	0.4472136
User3	0.5962848	0.5962848	0.20519567	0.3409972	0.18569534	0.40824828	0.7276069
User4	0.12909944	0.43386093	0.19069251	0.048795003	0.0	0.38490018	0.5
User5	0.16609097	0.4	0.6201737	0.0	0.36380345	0.6324555	0.28867513
User6	0.6	0.3409972	0.0	0.4472136	0.5	0.5735393	0.25
User7	0.32539567	0.048795003	0.06262243	0.18257418	0.0	0.5735393	0.0

Figure 4. Users and documents correspondence.

a document from the corpus to estimate the relevance degree of this one.

As the corpus and the users-profile are represented by two matrices having the same number of columns (terms which don't appear in both matrices will be added and assigned value '0' as weight to obtain the same dimension), the correlation between an individual and a document is done by calculating the cosine between a line vector (a document) of the corpus matrix and a line vector (a user) of the profile matrix using practically the same precedent formula:

$$COS(U_i, D_j) = \sum_n U_{in} * D_{jn} \; / \; \sqrt{\sum_n U_{in}^2 * \sum_n D_{jn}^2} \qquad (5)$$

where:

$COS(U_i, D_j)$ is the cosine or relevance degree of D_j document for user U_i, U_{in} is the weight of T_n term for user U_i, and D_{jn} is the weight of T_n term in D_j document.

The cosine calculated here represents the relevance degree between the document and the user correlated, on the basis of the same preceding principle, namely the closer this cosine is to 1, the more the document is considered to be interesting for user, and thus likely to be conveyed to him. For that, profiles and corresponding documents are organized into a two dimension vector $C(k * l)$ where lines represent users and columns correspond to documents, and across a line and a column the C_{ij} element refers to pertinence rate of j^{th} document for the i^{th} user as shown in Figure 4.

Having the degree of relevance of each document for each individual, does not remain more than fixing a threshold value from that a document is considered meeting the information requirements of a user to only transmit those which reach or exceed that value.

5. Discussion

5.1. Thesaurus Quality

Term classification is carried using automatic treatments based on statistical calculations; this appears in contradiction with semantic aspect which generally characterizes manually built thesaurus. This is compensated by human intelligence assistance in the analysis and consequently, the validation of updates operated on the thesaurus. As shown in Figure 5 this is possible by introducing graphic pre-visualization support, of the various resulting classes and the terms belonging to them, with an aim of allowing an administrator only retaining the most coherent judged results; therefore, voiding encumber thesaurus by inappropriate or incoherent entries.

Indeed, "human work upstream is essential. The all-automatic doesn't work. It is necessary to give a sense to the information which will be handled through manual intervention, in phase with the real user's needs" [6].

Therefore, a judicious choice of similarity's threshold allows the enrichment of the thesaurus by some terms, not only synonymous, but who maintains association relationship (see Table 1). It's clear that the use of this kind of relation which expresses the adherence to the same domain, and the same idea tends to reinforce the semantic value of the thesaurus with an aim of improving recall rate.

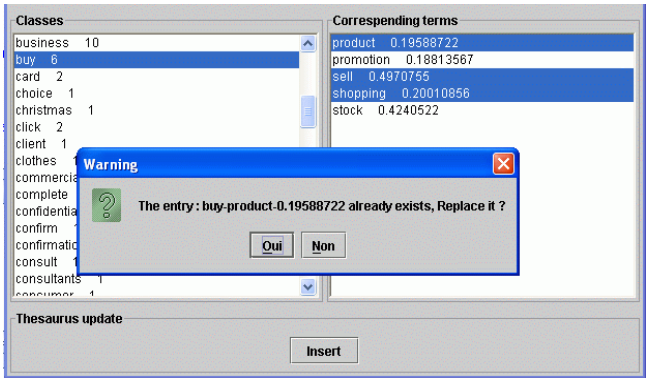

Figure 5. Manual validation of terms classification.

Table 1. Association relationship between terms belonging to the same class.

Class	Associated Terms
Internet	Network, web, http, ftp, connexion…
database	DBMS, SQL, Oracle…
Computing	Operating system, software, PC…
Economy	Business, market, productivity…
Alimentation	Food, obesity, appetite, weight…

5.2. Filtering Performances and Results Relevance

A classic filtering system operates a quite rigid binary selection decision: all incoming document in the treatment chain is considered pertinent, or no pertinent for a given profile. Our technique adopts more suppleness when selecting relevant documents by assigning them a degree of pertinence, as shown in

Figure 6. Thus, only the documents having the value '0' as relevance degree will be rejected. All remaining documents are considered potentially pertinent. And we can increase or decrease the number of diffused documents by changing the threshold value.

Considering Figure 6 precedent example, fixing the relevance threshold at the '0.3' value allows to recover five results corresponding to this profile (01, 02, 08, 12 and 16).

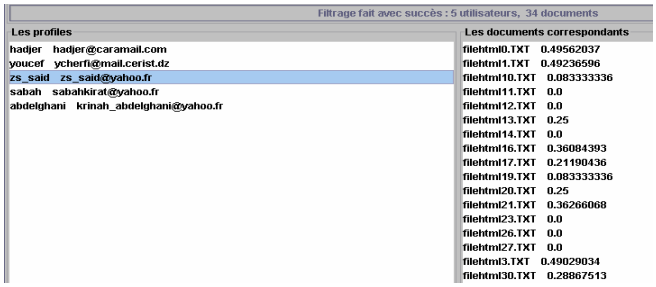

Figure 6. Result of a filtering stage.

By reducing this threshold to '0.2' value, 4 other documents will be returned (06, 09, 11 and 17), and this makes an increase of recall rate by 44%. On the other side, by choosing '0.4' threshold value, 2 results are going to be eliminated (08 and 12), this has effect reducing the noise rate by 40%.

We now move on to the choice of the threshold. A possible route which has been followed in [5] corresponds to setting threshold to a negative value (very low values in our case). The primary effect of this approach is to boost recall at the expanse of precision, resulting in increased net performances when precision and recall are scored the same, (refer to results in [5]).

However, this goes exactly against the TREC evaluations measures which put emphasis on precision [11]. To reduce recall we than have to set threshold to high values. This choice reduces dramatically the number on forwarded documents, pushing precision up. So, we note that a very simple relevance threshold notion, allows influencing, downstream, the quantity and the quality of the selected documents, the threshold value can be given dynamically by taking into account the objectives aimed (avoid the information overload, increase the rate of recall, seek a better precision etc).

6. Conclusion

In this article, we presented a natural language processing which depends on a Vectorial information organization and uses the euclidean (cosine) distance function. We tried to confirm the effect of the proposed metric distance by a document retrieval task. We attempted to show the advantage of allying automatic treatment to manual validation with the aim of warranting a good quality of performances.

References

[1] Balvet A., "Filtrage D'information Par Analyse Partielle Grammaires Locales, Dictionnaires Electroniques et Lexique-Grammaire Pour la Recherche D'information," *TALN 2001*, 2001.

[2] Benzécri J., *Pratique de L'analyse des Données, Linguistique et Lexicologie*, Dunod, Paris, 1981.

[3] Bruza P. and Dennis S., "Query Re-Formulation on the Internet: Empirical Data and the Hyper index Search Engine," *in Proceedings of the Conference: Computer-Assisted Information Searching on Internet (RIAO'97)*, Montreal, pp. 488-499, 1997.

[4] Callan J., "Document Filtering with Inference Network," *in Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Massachusetts, pp. 262-269, 1996.

[5] Cancedda N., Cesa-Bianchi N., Conconi A., Gentile C., Goutte C., Graepel T., Li Y., Renders J., Shawe-Taylor J., and Vinokourov A., " Kernel Methods for Document Filtering," *in Advances in Neural Information*, MIT Press, 2003.

[6] Cancedda N., Cesa-Bianchi N., Conconi A., Gentile C., Goutte C., Li Y., Renders J., Shawe-Taylor J., and Vinokourov A., "Kernel Methods For Document Filtering," *in Proceedings of The Eleventh Text Retrieval Conference (TREC'2002) Notebook Papers*, pp. 371-380, 2003.

[7] Cesa-Bianchi N., Conconi A., and Gentile C., "Margin-Based Algorithms for Information Filtering," *in Advances in Neural Information*, MIT Press, 2003.

[8] Favier L. and Ihadjadene M., "Vers des Systèmes de Découverte et de Filtrage d'Information Documentaire: Quelle Stratégie Faut-il Mettre en Place, " *in Proceedings of 28th CAIS/ACSI Conférence,* Alberta, pp. 98-127, 2000.

[9] Harman D., "National Institute of Standards and Technology," *in Proceeding of Fourth Text Retrieval Conference (TREC-4),* Gainthersburg, pp. 1-24, 1996.

[10] Jones S. and Cunningham S., *An Analysis of Usage of a Digital Library*, Crete, pp. 261-277, 1998.

[11] Lavelli A., Sebastiani F., and Zanoli R., "An Experimental Comparison of Term Representations for Term Management Applications," *in Proceedings of the SEBD-04*, Italia, pp. 190-201, 2004.

[12] Nouali O. and Krinah A., "Improvement of Retrieval and Filtering Systems by an Automatic Multi Words Extraction Tool," *in Proceedings of*

the CSIT'2006, vol. 2, pp. 386-392, Jordan, 2006.

[13] Nouali O., Zibouche S., and Krinah A., "SYCOFID: A Network Information Broadcasting System," *in Proceedings of (ACIT'2004*), vol. 2, pp. 669-670, Algeria, 2004.

[14] Packer K. and Soergel D., "The Importance of SDI for Current Awareness in Fields with Severe Scatter of Information," *Journal of American Society for Information Science*, vol. 3, no. 30, pp. 125-135, 1979.

[15] Salton. G. and McGill M., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[16] Salton G. and Yang C., "On the specification of Term Values in Automatic Indexing," *Journal of Documentation*, vol. 29, no. 4, pp. 351–372, 1973.

[17] Spink A., Bateman J., and Jansen B., "Searching Heterogeneous Collections on the Web: Behavior of EXCITE Users," *in Proceedings of the National Online Meeting,* New York, vol. 4, pp. 375-386, 1998.

[18] Wandmacher T. and Antoine J., "Exploiting the Context for Automatic Text Prediction: A Vectorial Approach," *Forum de l'école Doctorale Santé, Sciences, Technologies*, Université de Tours, France, 2006.

[19] Yochum. J., "A High-Speed Text Scanning Algorithm Utilizing Least Frequent Trigraphs," *in Proceedings of the IEEE International Symposium on New Directions in Computing*, Norway, pp. 114-121, 1985.

Omar Nouali received his BSc degree in computer science engineering from Houari Boumediene University of Science and Technology (USTHB) in 1988, his Master degree in computer science from Advanced Technology Center, Algeria in 1991, and his PhD in computer science from Houari Boumediene University of Science and Technology (USTHB) in 2004, Algeria. Currently, he is responsible of research in Basic Software Laboratory, CERIST, Algiers. His area of interests includes artificial intelligence, expert systems, natural language processing, information filtering and retrieval, learning, human computer interface, and security.

Abdelghani Krinah received his BSc degree in computer science engineering from Houari Boumediene University of Science and Technology (USTHB) in 2003. Currently, he is a software engineer and organization level II in Basic Software Laboratory, CERIST, Algiers. His area of interests includes artificial intelligence, information filtering and retrieval, and human computer interface.