PERGAMON

Information Processing and Management 37 (2001) 623–637

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Document ranking based upon Markov chains

Czesław Daniłowicz *, Jarosław Baliński

Department of Information Systems, Wrocław University of Technology, Wybrzeże S. Wyspianskiego 27, 50-370 Wrocław, Poland

Received 21 March 2000; accepted 26 June 2000

Abstract

One of the most important problems in information retrieval is determining the order of documents in the answer returned to the user. Many methods and algorithms for document ordering have been proposed. The method introduced in this paper differs from them especially in that it uses a probabilistic model of a document set. In this model documents are regarded as states of a Markov chain, where transition probabilities are directly proportional to similarities between documents. Steady-state probabilities reflect similarities of particular documents to the whole answer set. If documents are ordered according to these probabilities, at the top of a list there will be documents that are the best representatives of the set, and at the bottom those which are the worst representatives. The method was tested against databases INSPEC and Networked Computer Science Technical Reference Library (NCSTRL). Test results are positive. Values of the Kendall rank correlation coefficient indicate high similarity between rankings generated by the proposed method and rankings produced by experts. Results are comparable with rankings generated by the vector model using standard weighting schema $tf \cdot idf$.

Keywords: Document similarity; Document ranking; Markov chain

1. Introduction

Interests for document ranking are connected with the spread of on-line access systems in the 70's. With such systems grew a new group of users – end-users that performed searching without help of intermediary searchers. In general, end-users did not have the knowledge to formulate their needs in systems based on a Boolean model (Sewell & Teitelbaum, 1986), which was then the most popular one. We must emphasize that division of documents into two groups, relevant and

* Corresponding author. Tel.: +48-071-3203258; fax: +48-071-3203453.
E-mail addresses: danilowicz@zsi.pwr.wroc.pl (C. Daniłowicz), balinski@zsi.pwr.wroc.pl (J. Baliński).

0306-4573/01/$ - see front matter
PII: S0306-4573(00)00038-8

non-relevant ones with respect to the query, does not coincide with users' expectations. Since, in every set of documents users can distinguish documents more or less satisfying their information need expressed in the query. Moreover, they usually start reading from the most relevant documents.

There have been many proposed methods and strategies for ranking documents. It is not the goal of this paper to analyze them. We present here only important approaches.

In a vector model one assumes that terms (descriptors) occurring both in a document description and in a query may be of different importance. It is expressed using non-negative real numbers called weights. In such a case, a document and a query can be represented by vectors in the n-dimensional space, where n is equal to the number of all distinguished terms. After a similarity (or distance) measure has been defined in the vector space, documents may be ranked. Ordering function is just similarity calculated for a query vector and a document vector (Salton, 1971).

In a basis fuzzy model a query is a Boolean expression (in which term weights are identical), and documents are represented by vectors (Tahani, 1976). The so-called membership function is used for ranking, which determines the degree of conformity for a query and a document. Its values are defined recursively: for an AND operator based on appropriate t-norm (most frequently MAX), for an OR operator – based on appropriate t-norm (most frequently MIN), and for a NOT operator – as a complement to the maximal value (Gupta & Qi, 1991). The fuzzy model reduces itself to a Boolean model when all term weights in all documents are identical.

There are also a few further-reaching generalizations based upon the theory of fuzzy sets. From the theoretical point of view the most interesting one seems to be a model proposed by Salton, Fox and Wu (1983a), which is a generalization of both the Boolean and vector models. Documents are represented by vectors and queries are extended Boolean expressions, in which importance of terms is differentiated using weights. For document ranking, similarity between a query and a document is used, normalized by L_p vector norm. Generality of model follows from both a query form and the way similarity is calculated. Using parameter p one can decrease selectivity of the AND operator and increase selectivity of the OR operator. For margin values of p this model is compatible with the vector or fuzzy model.

For most models of ranking systems it is necessary to know weights of terms representing documents. There are many ways of calculating weights. In the simplest, document-oriented case a term weight depends exclusively on the characteristic of a given document (term frequency, document length). More complex are collection-oriented algorithms, which take into account not only features of indexed documents, but also features of other documents in the set. The importance of document collection in determining index terms and their weights was examined by Sparck Jones (1973), Salton and his co-workers (1971, 1983a,b, 1988), and others. There is also a known approach oriented towards system users (Choroś & Daniłowicz, 1982; Daniłowicz, 1983), which assumes that term weights should depend not only on features of individual documents and of the whole collection, but also on users' preferences.

A separate problem is determining term weights in queries. In the SMART system the same procedure for determining weights was to be used for documents and queries (Salton, 1971). No discrimination was obtained, however, since natural language query is usually an expression or simple sentence, rarely a complex sentence and very rarely text consisting of more than one sentence. Automatic indexing procedures used in SMART calculated term weights based on term

frequency in the text (uniformly for documents and queries). Because particular terms occurred in queries only once, their weights in the vector representation were all the same. In order to differentiate them, experts manually increased frequencies for the terms, which they found the most important.

Of course, one cannot recommend to end-users such a way of query formulating, and expect them to weight query terms by numbers from interval $(0, 1]$. These extensions of a Boolean model that enable users to formulate their information needs in a natural way can gain more acceptance. From this perspective linguistic approach is noteworthy. Bordogna and Pasi (1993) introduced a linguistic variable whose values (*important*, *very important*, *fairly important*, ...) are associated with each term appearing in the query and characterize the contents of the desired documents.

Probabilistic models are another group of information retrieval models. In the standard model one makes an assumption that term weights are binary (0 or 1), and a query is a subset of index terms. Order of documents in the answer is determined by the probability of a document being relevant (Robertson & Sparck Jones, 1976). Probabilities of being relevant are calculated based on probabilities that particular terms occurred in relevant and non-relevant documents. An extensive survey of related models can be found in the paper of Crestani, Lalmas, van Rijsbergen and Campbell (1998).

New impulses for research of document ranking arose, when the WWW information system came into being. Diversity of data, irregular document structures and huge sizes of WWW resources are the main reasons for problems with information retrieval on the Web. The sources of information for document retrieval and ranking are hyperlinks and document (page) contents. There are measures based only on hyperlinks (Carriere & Kazman, 1997; Kleinberg, 1998), they can be combined with traditional measures for document ordering. Proposed were also methods integrating information included in both hyperlinks, and the contents of HTML document (Croft & Turtle, 1993; Marchiori, 1997).

We mentioned here only a few methods from many presented in the literature. The method introduced in this paper is different from others especially in that it uses a probabilistic model of a document set for document ranking.

In the model we propose, documents are treated as states of an ergodic Markov chain (Feller, 1961), in which transition probabilities are proportional to similarities between documents. Steady-state probabilities then reflect similarities of particular documents to the whole set of documents in the answer. Arranging documents according to non-increasing steady-state probabilities, at the top of the list we obtain documents that best represent the content of the collection and at the bottom those that are the least representative.

2. Model of ranking process

2.1. Idea of ranking

One basis for document ranking is an information system in which the answer is a set of objects returned in random order. Each document retrieval system based on a Boolean model has such a feature. If one assumes that queries do not include an OR operator or this operator joins terms semantically similar, and the system correctly classifies documents as relevant and non-relevant,

then documents in the answer are similar one to another. Degrees of similarity are different, but we can rule out documents with similarity equal to zero. The practical problem we would like to solve is determining the order of similar documents.

We make an assumption that in the first rank will be the document whose contents comprise the most information in the set, i.e., the document best representing the set. It has the biggest chance to fulfil the information need specified by the user. The last document returned should include the least information specific to the whole set (i.e. the worst representative of the set). The degree to which the information in the collection overlaps with the information included in a document can we express the similarity between the document and the collection. We measure this similarity using a discrete time Markov chain. Interdocument similarities are used as estimates of transition probabilities in the ergodic Markov chain, and calculated steady-state probabilities are used for ranking documents. The idea of using a Markov chain to determine how representative a document is for the whole set can be motivated as follows.

Let us assume that the time documents from the set D are presented to the user, divided in intervals (moments) of the same length. The first document presented is chosen randomly. Let it be the document d_i. After the first moment has elapsed, we can go on presenting the document d_i or start presenting another document. The choice is based on preferences, which are determined by similarities.

Random choice of a document from the set D is made with the probability proportional to similarity of a document d_i to that document. So the greatest chance to be presented has that document to which d_i is the most similar, and the least chance has that one to which d_i is the least similar. After we have chosen the document to be presented in the second moment we choose the document to be presented in the third moment in the same way – and so on. If we continue the presentation then relative frequencies of particular documents will go to the fixed values, independent of the document presented first. When the similarity matrix reflects interdocument similarities with regard to the subject of the query (based on which the set D was built), relative frequencies are a good measure of representativeness for the document.

2.2. Model description

Let $D = \{d_1, d_2, \ldots, d_n\}$ be the answer set for the user query. Let us assume that it contains states of a discrete time Markov chain, and *transition probabilities* $p(d_j/d_i)$ are directly proportional to similarities $s(d_i, d_j)$ between documents:

$$p(d_j/d_i) = s(d_i, d_j)/c, \quad \text{where } c = \text{const for } i, j = 1, 2, \ldots, n. \tag{1}$$

Let $\mathbf{P} = [p_{i,j}]$ be the transition matrix, so $p_{i,j} = p(d_j/d_i)$, and $\mathbf{p} = [p_1, p_2, \ldots, p_n]$ be the vector of steady-state probabilities.

Steady-state probabilities are unconditional probabilities of reaching particular states, when the time (number of transitions) goes to infinity. They do not depend on prior probabilities of starting from a particular state. These probabilities involve all the ways a particular state can be arrived at. If transition probabilities $p(d_j/d_i)$ satisfy condition (1), then steady-state probabilities p_i, involve all direct and indirect similarities between documents in D and the particular document d_i. Documents in the answer set D are ranked according to non-increasing steady-state probabilities p_i.

2.3. Calculation of steady-state probabilities

Transition probabilities $p(d_j/d_i)$ are calculated based on similarities between documents $s(d_i, d_j)$ after normalization given by the following formula:

$$p_{i,j} = \frac{s(d_i, d_j)}{\max_l\left(\sum_k s(d_l, d_k)\right)}, \quad \text{where } i, j, k, l = 1, 2, \ldots, n. \tag{2}$$

Formula (2) fulfils condition (1), since the denominator (maximal sum of similarities) is constant for a given set of documents D.

The set D with such calculated probabilities does not usually define the Markov chain, because $\sum_{j=1}^{n} p_{i,j} \leqslant 1$, and it is necessary for every state d_i to satisfy the equality:

$$\sum_{j=1}^{n} p_{i,j} = 1. \tag{3}$$

If condition (3) is not satisfied, we introduce an additional state d_{n+1}. Transition probabilities are calculated as follows:

$$p_{i,n+1} = 1 - \sum_{j=1}^{n} p_{i,j} \quad \text{for } i = 1, 2, \ldots, n \tag{4}$$

transition probabilities from state d_{n+1} to state d_i are constant:

$$p_{n+1,i} = p \quad \text{for } i = 1, 2, \ldots, n, \text{ where } 0 < p \leqslant 1/n, \tag{5}$$

moreover, probability of remaining in d_{n+1} is set to be equal

$$p_{n+1,n+1} = 1 - \sum_{i=1}^{n} p_{n+1,i} = 1 - np. \tag{6}$$

After we have introduced the additional state d_{n+1} and calculated transition probabilities according to formulas (3)–(6), we obtain the Markov chain with $n+1$ states: $d_1, d_2, \ldots, d_n, d_{n+1}$. Steady-state probabilities $p_1, p_2, \ldots, p_n, p_{n+1}$ can be calculated, if the Markov chain is ergodic.

In our case, for any documents d_i, $d_j \in D$, similarity $s(d_i, d_j) \neq 0$, hence from (2) follows that also $p(d_j/d_i) \neq 0$. Moreover, from (5) follows that $p(d_{n+1}/d_i) \neq 0$. Concluding, n columns in matrix $\mathbf{P}$ do not have zeros. We have then built the ergodic Markov chain.

Steady-state probabilities (vector $\mathbf{p}$) can be derived from the equation

$$\mathbf{p} = \mathbf{pP} \tag{7}$$

taking into account the following condition:

$$\sum_{i=1}^{n+1} p_i = 1. \tag{8}$$

After we have calculated steady-state vector $\mathbf{p}$, we omit probability p_{n+1}, and use only probabilities $p_1, p_2, \ldots, p_n$ for ranking.

We must emphasize that the additional state d_{n+1}, necessary for the Markov chain to be built, does not have influence on the order of documents (determined by $p_1, p_2, \ldots, p_n$), if transition probabilities from state d_{n+1} to d_i are fixed according to (5). One can show that ratios of ergodic probabilities for every pair of states stay the same independent of the value of p (Appendix A).

2.4. Ranking using relevance

Let us assume now that ranking is based on some relevance function $r(q,d)$, and the ordered answer to query q is represented by the sequence:

$$D_r = (a_1, a_2, \ldots, a_n), \tag{9}$$

where a_i indicates the position of the document d_i in the answer.

Usually values of relevance are calculated as a similarity between a document and a query. Then, ranking (9) similarities $s(d_i,d_j)$ between documents in set $D = \{d_1, d_2, \ldots, d_n\}$ are not taken into account. To take advantage of ranking based on relevance and interdocument similarities, we introduce the modified similarity function. One can distinguish two cases.

In the first one, we assume that the relevance function $r(q,d)$ or its values are known for a query q and documents in the set D. Moreover, $r(q,d)$ and $s(d_i,d_j)$ are normalized ($0 \leqslant r(q,d), s(d_i,d_j) \leqslant 1$). Then similarity $s(d_i,d_j)$ can be modified according to the following formula:

$$s'(d_i,d_j) = (1 - |r(q,d_i) - r(q,d_j)|)s(d_i,d_j). \tag{10}$$

In the second case, we assume that only the order of documents (9) is known, as opposed to values of $r(q,d)$. Then the modified similarity function may have the following form:

$$s''(d_i,d_j) = \left(1 - \frac{|a_i - a_j|}{m}\right)s(d_i,d_j), \tag{11}$$

where m is the number of different values in sequence D_r.

Using the modified similarity function one can calculate transition and steady-state probabilities, in the same manner as in the above outlined procedure, and make a new modified ranking of documents.

3. Evaluation framework and results

The main goal of this experiment is checking the degree of agreement for ranking generated by the proposed method and ranking produced by experts. We investigate also correlation between ranking generated by the vector model using standard weighting schema tf · idf and ranking of experts. It lets us compare this standard method with one based on Markov chains. Test results will be shown for INSPEC and Networked Computer Science Technical Reference Library (NCSTRL) databases (in the latter case computed is also correlation between ranking generated in the system with that produced by experts). For each database we examined rankings for 20 queries. Queries have been formulated and results have been verified by experts (research workers in the Department of Information Systems at the University of Technology in Wrocław). For both databases retrieval, ordering and an evaluation of results was performed based on abstracts. Rankings were compared using the Kendall coefficient of rank correlation (Appendix B).

From the answer set for each query a dictionary was built containing all the words which occurred in abstracts, excluding words from a stoplist. The stoplist included prepositions and conjunctions most frequently appearing in English. A document was represented by the vector

$$d_i = [f_{i1}, f_{i2}, \ldots, f_{it}],$$

where t is the number of words in the dictionary.

The component f_{ij} determines the number of times a term labelled with j occurs in the document d_i (i.e. frequency of a term in a document).

The method based on the vector model uses the document-query similarity for document ranking:

$$s_v(q, d_i) = \frac{\sum_{k=1}^{t} w_{qk} w_{ik}}{\sqrt{\sum_{k=1}^{t} w_{qk}^2 \sum_{k=1}^{t} w_{ik}^2}},$$

where

$$w_{qk} = \left(0.5 + \frac{0.5\,\mathrm{freq}_{qk}}{\mathrm{maxfreq}_k}\right)\mathrm{IDF}_k, \quad w_{ik} = f_{ik}\,\mathrm{IDF}_k,$$

$$\mathrm{IDF}_k = \log_2 \frac{\max n}{n_k} + 1,$$

where freq_{qk} is the frequency of term k in the query q, $\mathrm{maxfreq}_q$ the maximal frequency of any term in the query q, $\max n$ the maximal number of documents containing the same term, and n_k is the number of documents containing a term labelled k.

Chosen variant of tf · idf weighting is recommended for most situations by Salton and Buckley (1988). This method, in the sequel referred to as standard method, is a good basis for comparison.

In our method the similarity between *documents*, needed for the transition matrix to be computed, was defined using the cosine function:

$$s(d_i, d_j) = \frac{\sum_{k=1}^{t} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{t} w_{ik}^2 \sum_{k=1}^{t} w_{jk}^2}}.$$

Term weights in documents are identical, as in the case of the vector model, i.e.,

$$w_{ik} = f_{ik}\,\mathrm{IDF}_k, \quad w_{jk} = f_{jk}\,\mathrm{IDF}_k.$$

3.1. Retrieval in INSPEC databases

INSPEC databases were used to test the ranking of Boolean answers. Retrieval covered the time January 1999–June 1999. The rankings obtained using Markov chain analysis and rankings

generated by the standard method were compared with experts' (authors of queries) rankings. All of the queries were term conjunctions.

As an example we discuss the processing of the query:

$$q = \text{AB}\{\text{ONLINE}\}\ \text{AND}\ \text{AB}\{\text{RECOGNITION}\}\ \text{AND}\ \text{AB}\{\text{HANDWRITING}\},$$

where AB{*term*} denotes that *term* should appear in abstract.

The answer set for the query q is shown in Table 1.

After calculating similarities for each pair of documents we obtained the following matrix (see Table 2).

After normalization and adding the state d_{10} the transition matrix for an ergodic Markov chain was calculated (see Table 3).

Then steady-state probabilities were calculated

$$\mathbf{p} = [0.066, 0.088, 0.115, 0.096, 0.079, 0.068, 0.114, 0.073, 0.075, 0.225],$$

which were used to rank documents, as shown in Table 4.

Table 1
Answer to the query q

Document no.	Document title
1	Hybrid Multiplier/Cordic unit for online handwriting recognition
2	Forward search with discontinuous probabilities for online handwriting recognition
3	Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data
4	Advanced state clustering for very large vocabulary HMM-based on-line handwriting recognition
5	Approximate stroke sequence string matching algorithm for character recognition and analysis
6	Recovery of drawing order from scanned images of multi-stroke handwriting
7	A hybrid NN/HMM approach for large vocabulary online handwriting recognition
8	Recognizing online handwritten alphanumeric characters through flexible structural matching
9	Network-based approach to online cursive script recognition

Table 2
Similarity matrix for the query q

Document no.	1	2	3	4	5	6	7	8	9
1	1.000	0.034	0.075	0.034	0.046	0.014	0.047	0.019	0.061
2	0.034	1.000	0.040	0.081	0.079	0.051	0.137	0.059	0.044
3	0.075	0.040	1.000	0.150	0.097	0.076	0.168	0.108	0.090
4	0.034	0.081	0.150	1.000	0.031	0.024	0.171	0.024	0.036
5	0.046	0.079	0.097	0.031	1.000	0.071	0.046	0.043	0.051
6	0.014	0.051	0.076	0.024	0.071	1.000	0.057	0.034	0.013
7	0.047	0.137	0.168	0.171	0.046	0.057	1.000	0.062	0.081
8	0.019	0.059	0.108	0.024	0.043	0.034	0.062	1.000	0.029
9	0.061	0.044	0.090	0.036	0.051	0.013	0.081	0.029	1.000

Table 3
Transition matrix for the query q

State no.	1	2	3	4	5	6	7	8	9	10
1	0.555	0.019	0.041	0.019	0.026	0.008	0.026	0.011	0.034	0.263
2	0.019	0.555	0.022	0.045	0.044	0.028	0.076	0.033	0.024	0.153
3	0.041	0.022	0.555	0.083	0.054	0.042	0.093	0.060	0.050	0.000
4	0.019	0.045	0.083	0.555	0.017	0.013	0.095	0.013	0.020	0.140
5	0.026	0.044	0.054	0.017	0.555	0.039	0.026	0.024	0.028	0.188
6	0.008	0.028	0.042	0.013	0.039	0.555	0.032	0.019	0.007	0.257
7	0.026	0.076	0.093	0.095	0.026	0.032	0.555	0.035	0.045	0.018
8	0.011	0.033	0.060	0.013	0.024	0.019	0.035	0.555	0.016	0.237
9	0.034	0.024	0.050	0.020	0.028	0.007	0.045	0.016	0.555	0.221
10	0.056	0.056	0.056	0.056	0.056	0.056	0.056	0.056	0.056	0.500

Table 4
Rankings for the query q

Document no.	1	2	3	4	5	6	7	8	9
R1. Ranking based on steady-state probabilities	9	4	1	3	5	8	2	7	6
R2. Ranking generated by the standard method	8	5	1	7	4	7	6	2	3
R3. Ranking of experts	6	2	2	2	3	5	1	2	4

In case of the standard method we assumed that the query q was of the form $q = \text{ONLINE RECOGNITION HANDWRITING}$. Next for each document d_i values of query-document similarities $s_v(q, d_i)$ were calculated, and the following vector of similarities was obtained:

$$[0.055, 0.086, 0.229, 0.059, 0.096, 0.059, 0.083, 0.107, 0.098].$$

It was the basis for ranking R2.

For comparison of rankings the Kendall coefficient (Kendall, 1948) was used (Appendix B). It measures the correlation between two rank lists. Calculated values for the list established by experts and the list generated by the introduced method is equal to $\tau(\mathrm{R1}, \mathrm{R3}) = 0.730$. So the correlation is significant. Kendall coefficient for the list generated by experts and the list produced by the standard method is $\tau(\mathrm{R2}, \mathrm{R3}) = 0.278$.

For the remaining 19 queries the number of documents in the answer set was between 6 and 52. The median of Kendall coefficient $\tau(\mathrm{R1}, \mathrm{R3})$ is equal to 0.51, the median of $\tau(\mathrm{R2}, \mathrm{R3})$ is equal to 0.64.

3.2. Retrieval in NTCSRL system

NCSTRL is an international collection of articles and research reports in the area of computer science. It is made available for non-commercial use from 160 scientific institutions and archives. The system is available on WWW (http://www.ncstrl.org) and enables retrieval in document abstracts.

For each of 20 queries the following options were chosen: term conjunction and searching in abstracts. Median values for Kendall coefficients are shown in Table 5.

Table 5
Median values for rank correlation coefficients

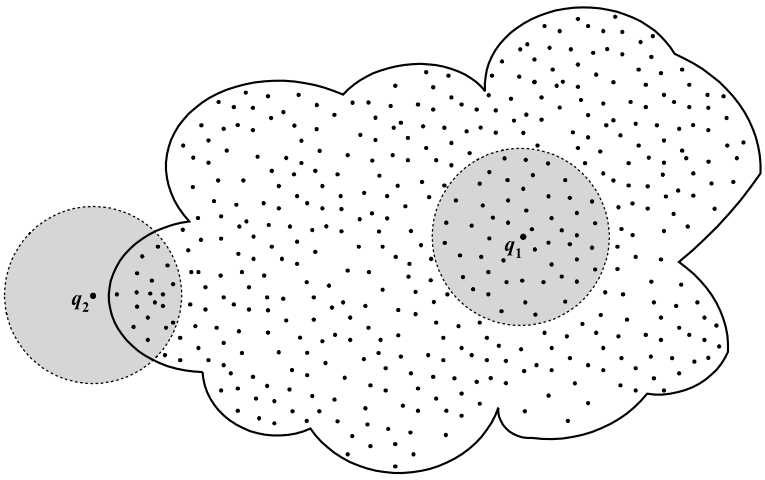
Type of ranking	Median value of correlation with the ranking of experts
Ranking based on steady-state probabilities	0.56
Ranking generated by the standard method	0.75
Ranking of NCSTRL	0.49

4. Conclusions and future work

The method described in this paper can be applied to retrieval (ranking) and classification (calculating class representatives) of any objects, if the similarity or distance measure is given. For document retrieval systems we assumed that in the set being ordered similarity between every pair of documents is positive. It is not a necessary condition. If it is not satisfied, however, problems may arise with the interpretation of ranking.

It will become clear, if we consider the query $q = term\ 1$ OR $term\ 2$, where meanings of *term* 1 and *term* 2 are disjoint. The answer for such a query is usually the sum of two disjoint sets: one corresponding to term 1 and the other corresponding to term 2. But if similarities between some documents from both sets are greater than zero, ordering such a set can be conducted. At the top of the ranked list may occur documents represented by term 1 or documents represented by term 2. In both cases it is doubtful that they are representative for the whole set of documents.

Even if the answer for the query is a set of similar documents, ranking based on the Markov chain method may differ from ranking determined by the expert or the author of the query. It was not the case, as far as our experiments are concerned, but it can not be ruled out. The drawing shows the document collection and two queries, represented by points on the plane, preserving real distances (degrees of dissimilarity). Areas corresponding to answers for the queries q_1 and q_2 are marked grey. One can expect that the ranking produced for the query q_1 will be strongly correlated with the ranking of the expert. Such an expectation is not justified in the case of the query q_2. The document in the first rank will be a good representative of the answer set for the query q_2, but it will not be the most similar to this query.



Results of tests carried out for databases INSPEC and NCSTRL are encouraging. Values of the Kendall coefficient indicate high similarity of rankings generated by the proposed method and rankings produced by experts. Produced results are comparable with ones generated by the standard method. The drawback is a long processing time. The method can then be recommended for further verification in such systems where time is not a critical parameter. These are, for example, agent-based information retrieval systems. An agent or agents play the role of the user's assistant. It returns the answer to the user-specified query immediately, using collected resources according to the profile of user interests (Daniłowicz, 1994), or takes the query for further processing and returns the answer after analysis, possibly in co-operation with other agents. For a long time in many scientific fields theoretical research was performed, and the results formed the basis for such agent ideas (Ferber, 1999). Now many teams are also engaged in experimental research (for example Davies, Weeks, & Revett, 1997; Jieh Hsiang & Hsieh-Chang Tu, 1999; El-Beltagy, De Roure, & Hall, 1999).

Tests conducted allow us also to formulate critical remarks on the similarity function. It appeared that the document ranking is highly dependent on the quality of the similarity function.

We have noticed that some documents not much similar to others have good ranks what they do not deserve. The reason for that phenomenon is first of all the similarity of a document to itself. It is maximal for all documents, also when they are less relevant to the query. In this case the similarity of a document to itself is determined to a high degree by the contents which do not conform to the query, or the set profile. In order to eliminate this source of distortion we plan to use modified functions in further experiments. They will still satisfy condition $s(d_i, d_i) \geqslant (d_i, d_j)$ (Soergel, 1967), but the limitation will be introduced:

$$s(d_i, d_i) = \max_j s(d_i, d_j) \quad \text{for } i = 1, 2, \ldots, n.$$

Besides, symmetric similarity functions do not exactly reflect dependencies between documents, which should determine the ranking. For example let us consider the document d_i, which is a chapter of the book d_j being a collective work. Assumption that the similarity of d_i to d_j is equal to the similarity of d_j to d_i is justified for grouping tasks. It increases chances of both documents to be qualified to the same group, which is not counterintuitive. But the same assumption is a strong oversimplification in case of determining the order of documents. That is why we will use non-symmetric similarity functions in further experiments.

A distinct problem, requiring detailed investigation, is ordering the set when similarity is not given for every pair of documents. Especially interesting is the ordering of WWW pages using hyperlinks. Since links do not reflect all similarities between pages in the set, ranking may require improvement with the help of modified similarity functions (formulas (10) and (11)).

Appendix A

Theorem. *Let* $\mathbf{P}$ *and* $\mathbf{P}'$ *be transition matrices of two Markov chains* M *and* M', *both containing* $n+1$ *states*, *such that*:

$$p_{i,j} = p'_{i,j}, \quad p_{n+1,j} = p, \quad p'_{n+1,j} = p' \quad \text{and} \quad 0 < p, \ p' \leqslant 1/n \quad \text{for } i, j = 1, 2, \ldots, n.$$

If both chains M and M′ are ergodic, then respective steady-state probabilities $\mathbf{p} = [p_1, p_2, \ldots, p_{n+1}]$ *and* $\mathbf{p}' = [p'_1, p'_2, \ldots, p'_{n+1}]$ *satisfy the following condition*:

$$\left(\frac{p_i}{p'_i} = const\right) \quad for\ i = 1, 2, \ldots, n.$$

Proof. Steady-state probabilities $\mathbf{p} = [p_1, p_2, \ldots p_n, p_{n+1}]$ for the chain M can be calculated from the following system of equations:

$$\mathbf{pP} = \mathbf{p}, \quad \sum_{i=1}^{n+1} p_i = 1.$$

Transforming the first equation we obtain successively:

$$(\mathbf{pP})^{\mathrm{T}} = \mathbf{p}^{\mathrm{T}},$$
$$\mathbf{P}^{\mathrm{T}}\mathbf{p}^{\mathrm{T}} = \mathbf{p}^{\mathrm{T}},$$
$$(\mathbf{P}^{\mathrm{T}} - \mathbf{I})\mathbf{p}^{\mathrm{T}} = 0,$$

where $\mathbf{I}$ is the identity matrix

Then, replacing the last equation with equation $\sum_{i=1}^{n+1} p_i = 1$, we obtain the following system of equations:

$$\begin{bmatrix} p_{1,1}-1 & p_{2,1} & \ldots & p_{n,1} & p_{n+1,1} \\ p_{1,2} & p_{2,2}-1 & \ldots & p_{n,2} & p_{n+1,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1,n} & p_{2,n} & \ldots & p_{n,n}-1 & p_{n+1,n} \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ p_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

and after substituting $p_{n+1,i} = p$ for $i = 1, 2, \ldots, n$, we have the system of $n + 1$ equations:

$$\begin{bmatrix} p_{1,1}-1 & p_{2,1} & \ldots & p_{n,1} & p \\ p_{1,2} & p_{2,2}-1 & \ldots & p_{n,2} & p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1,n} & p_{2,n} & \ldots & p_{n,n}-1 & p \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ p_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

We solve it using the Gaussian elimination method. After the first step – setting all values beneath the main diagonal to zero – the system has the following form:

$$\begin{bmatrix} w_{1,1} & w_{2,1} & \ldots & w_{n,1} & \alpha_1 p \\ 0 & w_{2,2} & \ldots & w_{n,2} & \alpha_2 p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & w_{n,n} & \alpha_{n+1} p \\ 0 & 0 & \ldots & 0 & 1 + \alpha_{n+1} p \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ p_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

None of the values $w_{i,j}$ and none of the values α_i depend on p, because in the first n columns the value of p does not appear. In the last column values $\alpha_i p$ appear, since any linear combination of terms $k_i p$ also has the form of kp.

In the second step we set to zero all the values over the main diagonal, omitting the last column. After this transformation we obtain a system of $n+1$ equations in the form:

$$\begin{bmatrix} 1 & 0 & \dots & 0 & \beta_1 p \\ 0 & 1 & \dots & 0 & \beta_2 p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \beta_{n+1} p \\ 0 & 0 & \dots & 0 & 1+\beta_{n+1} p \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ p_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Here, none of the values β_i depend on p either. So we obtain the following solution:

$$p_{n+1} = \frac{1}{1+\beta_{n+1} p},$$

$$p_i = -\beta_i p p_{n+1} \quad \text{for } i = 1, 2, \dots, n,$$

whereas values of β_i do not depend on p.

It is the only solution, since the Markov chain has a unique steady-state vector.

Steady-state probabilities for the chain M' can be calculated after making the same operations on coefficients $p'_{i,j}$. Since $p'_{i,j} = p_{i,j}$ for $i, j = 1, 2, \dots, n$, and $p'_{n+1},\ i = p'$, we obtain the solution:

$$p'_{n+1} = \frac{1}{1+\beta_{n+1} p'},$$

$$p'_i = -\beta_i p' p'_{n+1} \quad \text{for } i = 1, 2, \dots, n,$$

whereas values of β_i do not depend on p'.

Hence, for any $i = 1, 2, \dots, n$ we obtain

$$\frac{p_i}{p'_i} = \frac{p}{p'} \frac{p_{n+1}}{p'_{n+1}},$$

and since

$$\frac{p_i}{p'_i} = \text{const} \quad \text{for } i = 1, 2, \dots, n.$$

It also follows that

$$\frac{p_i}{p_j} = \frac{p'_i}{p'_j} = \text{const} \quad \text{for } i, j = 1, 2, \dots, n.$$

Ratios of steady-state probabilities are then constant and, consequently, for each non-increasing sequence of p_i there exists a non-increasing sequence of p'_i with identical sequences of indices.

Appendix B

B.1. Kendall coefficient

As a result of ordering the set of documents $D = \{d_1, d_2, \ldots, d_n\}$ using two methods, the lists Q and R have been obtained. Let us represent them by sequences:

$$Q = (q_1, q_2, \ldots, q_n) \quad \text{and} \quad R = (r_1, r_2, \ldots, r_n),$$

where q_i and r_i denote positions (*ranks*) of object d_i in the list Q and R, respectively. The rank correlation of lists Q and R is defined by the following coefficient:

$$\tau = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}}{\sqrt{\left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2\right)\left(\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^2\right)}},$$

where

$$a_{ij} = \begin{cases} +1 & \text{if } q_i < q_j, \\ -1 & \text{if } q_i > q_j, \\ 0 & \text{if } q_i = q_j, \end{cases} \qquad b_{ij} = \begin{cases} +1 & \text{if } r_i < r_j, \\ -1 & \text{if } r_i > r_j, \\ 0 & \text{if } r_i = r_j. \end{cases}$$

The Kendall coefficient takes values from the interval $[-1, 1]$. If lists are identical, it is equal to 1, if the lists are in reversed order, it is equal to -1. When the lists are not correlated, it is equal to 0.

Example. Transformation of lists:

Q	Object	d_7	d_4	d_1	d_2	d_6	d_5	d_8	d_3	d_9
	Position	1	2	2	3	3	4	5	6	7
R	Object	d_4	d_2	d_7	d_1	d_6	d_8	d_9	d_5	d_3
	Position	1	2	3	3	3	3	4	5	5

are sequences $Q = (2, 3, 6, 2, 4, 3, 1, 5, 7)$ and $R = (3, 2, 5, 1, 5, 3, 3, 3, 4)$. The degree of correlation for Q and R (Kendall coefficient) is equal to 0.4777.

References

Bordogna, G., & Pasi, G. (1993). A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation. *Journal of the American Society for Information Science*, *44*(2), 70–82.

Carriere, J., & Kazman, R. (1997). WebQuery: searching and visualizing the web through connectivity. *Computer Networks and ISDN Systems*, *29*(8–13), 1257–1267.

Choroś, K., & Daniłowicz, C. (1982). Relative indexing. Weighted descriptors and relative indexing in a document retrieval system model. *Information Processing and Management*, *18*(4), 207–220.

Crestani, F., Lalmas, M., van Rijsbergen, C. J., & Campbell, I. (1998). "Is this document relevant?...probably": a survey of probabilistic models in information retrieval. *ACM Computing Surveys*, *30*(4), 528–552.

Croft, W. B., & Turtle, H. B. (1993). Retrieval strategies for hypertext. *Information Processing and Management*, *29*(3), 313–324.

Daniłowicz, C. (1983). Relative indexing on the basis of users' profiles. *Information Processing and Management*, *19*(3), 159–163.

Daniłowicz, C. (1994). Modelling of user preferences and needs in Boolean retrieval systems. *Information Processing and Management*, *30*(3), 363–378.

Davies, N. J., Weeks, R., & Revett, M. C. (1997). Information agents for the World Wide Web. In H. S. Nwana, & N. Azarmi, *Software agents and soft computing. Towards enhancing machine intelligence. Concepts and applications* (pp. 81–99). Berlin: Springer.

El-Beltagy, S., De Roure, D., & Hall, W. (1999). A multiagent system for navigation assistance and information finding. In *Proceedings of the fourth international conference on the practical applications of intelligent agents and multi-agent technology* (pp. 281–295). Blackpool: Practical Application Company.

Feller, W. (1961). *An introduction to probability theory and its applications*. New York: Wiley.

Ferber, J. (1999). *Multi-agent systems*. New York: Addison-Wesley Longman Inc.

Gupta, M. M., & Qi, J. (1991). Theory of T-norms and fuzzy inference methods. *Fuzzy Sets and Systems*, *40*, 431–450.

Jieh Hsiang, & Hsieh-Chang Tu, (1999). Personalized Web retrieval: three agents for retrieving Web information. In T. Ishida, *Multiagent platforms. first Pacific Rim international workshop on multi-agents, PRIMA'98. Selected papers* (pp. 118–132). Berlin: Springer.

Kendall, M. G. (1948). *Rank correlation methods*. London: Griffin.

Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the ninth ACM-SIAM symposium on discrete algorithms* (pp. 668–677).

Marchiori, M. (1997). The quest for correct information on the web: hyper search engines. *Computer Networks and ISDN Systems*, *29*(8–13), 1225–1231.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*(3), 129–146.

Salton, G. (1971). *The SMART retrieval system – Experiments in automatic document retrieval*. Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*(5), 513–523.

Salton, G., Fox, E. A., & Wu, H. (1983a). Extended Boolean information retrieval. *Communications of the ACM*, *26*(11), 1022–1036.

Salton, G., Wu, H., & Yu, C. T. (1983b). The measurement of term importance in automatic indexing. *Journal of the American Society for Information Science*, *32*(3), 175–186.

Sewell, W., & Teitelbaum, S. (1986). Observations of end-user online searching behavior over eleven years. *Journal of the American Society for Information Science*, *37*(4), 234–245.

Soergel, D. (1967). Mathematical analysis of documentation systems. An attempt to a theory of classification and search request formulation. *Information Storage and Retrieval*, *3*(3), 129–173.

Sparck Jones, K. (1973). Index term weighting. *Information Storage and Retrieval*, *9*, 619–633.

Tahani, V. (1976). A fuzzy model of document retrieval systems. *Information Processing and Management*, *12*(3), 177–188.