Improving Document Ranking in Information Retrieval Using Ordered Weighted Aggregation and Leximin Refinement

Mohand Boughanem **Yannick Loiseau** **Henri Prade**
IRIT, 118 route de Narbonne, 31062 Toulouse cedex 4
{bougha,loiseau,prade}@irit.fr

Abstract

Classical information retrieval (IR) methods often lose valuable information when aggregating weights, which may diminish the discriminating power between documents. To cope with this problem, the paper presents an approach for ranking documents in IR, based on a vector-based ordering technique already considered in fuzzy logic for multiple criteria analysis purpose. Moreover, the proposed approach uses a possibilistic framework for encoding the retrieval status values. The approach, applied to a benchmark collection, has been shown to improve IR precision w.r.t. classical approaches.

1 Introduction

In classical information retrieval systems, documents and queries are usually represented by sets of weighted terms. Weights on document terms reflect statistics on the presence of the terms in the document and in the collection, while weights in the query terms express preferences. To evaluate to what extent a document is relevant to a query, a *retrieval status value* (rsv) is computed by aggregating the above weights for the terms present in the query, in a way that reflects the query structure (expressing disjunction or conjunction). Then documents are ranked on the basis of the *rsv*'s. Different kinds of aggregation functions can be used for combining the weights of the terms (pertaining to the same document) that are present in the considered query (assumed in this paper to be without any user's preference weighting). Candidate operators for aggregation that are found in the literature are average, similarity-based evaluation, p-norms [9], fuzzy logic conjunctive or disjunctive operations. However, this type of approach leads to a loss of information, since individual keyword values are fused together. A consequence is that it is impossible to discriminate documents having the same global relevance value. As an example, let us consider a three-terms query, aggregated by the average. This is only an example, and remarks similar to the ones below apply to other aggregation operators, including *min* and other fuzzy logic connectives. Let us suppose that the evaluation of the query $q = t_1 \wedge t_2 \wedge t_3$ on two documents d_1 and d_2 gives the following results (using normalized weights):

$$\begin{aligned} rsv(q, d_1) &= (w(t_1, d_1) + w(t_2, d_1) + w(t_3, d_1))/3 \\ &= (0.1 + 0.7 + 0.7)/3 = 0.5 \\ rsv(q, d_2) &= (w(t_1, d_2) + w(t_2, d_2) + w(t_3, d_2))/3 \\ &= (0.5 + 0.5 + 0.5)/3 = 0.5 \end{aligned}$$

The issue is to know whether the user prefers a document with a medium relevance for all her criteria, or having a high relevance for most of them. This example not only raises an ambiguity problem between documents having apparently the same relevance, but more generally points out the problem of the impact of terms having weights much higher than others. If we want to privilege d_1 over d_2, this problem can be dealt with by using operators such as Ordered Weighted Average [10], which focus on the weights with high values and model quantifiers such as *most of* [5], provided that such a quantifier is specified in the query. But this does not give a way of preferring d_2 to d_1 if we consider that one low weight can be a serious reason for discounting a document.

In this paper, we try another road. We no longer plan to aggregate the weights, but rather to rank-order the documents directly on the basis of the vectors of the weights of the terms present in the query, using decision making ideas that handle multiple criteria values (here replaced by the relevance value of each query term). This alternative method is described in section 2. Besides, the possibilistic framework used in the indexation of documents [7], is briefly recalled in section 3. The section 4 presents the results of a large scale evaluation benchmark.

2 Multicriteria ranking

At least two approaches can be used to compare objects according to multiple criteria. The first one is to aggregate these criteria, then to compare the obtained values. This corresponds to the classical information retrieval approach, considering each query term relevance as a criterion to fulfil. The second method amounts to compare the criteria evaluation vectors directly by using a refinement of Pareto ordering. This later method is discussed in this paper. We briefly discuss the aggregation approach first.

2.1 Ordered weighted minimum

Query terms are usually weighted in order to allow the user to express her preferences and assess the importance of each term. Therefore, the result of the evaluation of a query on a document is a vector of the weights of the terms of the document present in the query, usually modified for taking into account preferences about the importance of the terms in the query. This is why classical IR aggregation methods use weighted conjunction (or disjunction) operators. In conjunctive queries, these operators can be weighted average or weighted minimum. Similar ideas applies to disjunctions as well. However, this kind of aggregation is too restrictive. To relax the conjunction, ordered weighted operators, such as average (OWA [10]) or minimum (OWmin [4]) have been introduced. The idea underlying this type of aggregation is to give low importance to the smallest weights in the evaluation vector, thus minimizing the impact of small terms, which amounts to model a *most of* quantifier.

The *OWmin* operator uses an auxiliary vector of levels of importance in order to minimize the impact of low weighted terms on the final relevance value. Thus, as for OWA, the term weights vectors are ordered and discounted by importance levels, using the minimum instead of the average for computing the global evaluation. Two weighting methods are considered, based on Dienes implication and on Gödel implication respectively (e.g. [4]). For a vector $T = t_1, \ldots, t_n$ representing the indexing weights for a document, t_i is the relevance degree between the i^{th} query term and the document. The vector is assumed to be decreasingly ordered (i.e. $t_i \geq t_{i+1}$). Let $W = (w_1, \ldots, w_n)$ be the level of importance vector, also assumed to be decreasingly ordered, i.e. $w_i \geq w_{i+1}$, with $w_1 = 1$. The idea is to give more importance (w_i high) to the terms with a high relevance degree. The $OWmin$ aggregation using Dienes implication will be: $OWmin_D(T, W) = \min_i(\max(t_i, 1-w_i))$, while the Gödel implication is defined by $w_i \rightarrow t_i = 1$ if $w_i \leq t_i$ and $w_i \rightarrow t_i = t_i$ otherwise, which gives: $OWmin_G(T, W) = \min_i(w_i \rightarrow t_i)$. In both cases, if the smallest w_i's are zero, these weighted aggregations amount in practice to restrict the minimum to the t_i's with high values (since small t_i's will be replaced by 1 in the aggregation, and the high values of t_i's, corresponding to values of w_i's equal or close to 1, will remain unchanged).

However, as already said, we want to rank-order documents by taking advantage of the full weights vector associated with each document, rather than using an aggregated value. This means that we keep the idea of using weights for modifying the indexing weights and restricting the focus of the evaluation, but we no longer compute an aggregated value (taken above as the minimum). In order to compare vectors, the classical Pareto partial ordering has to be refined, since no pairs of documents should remain incomparable. In the following, we use refinements of the *min* operation, which do refine the Pareto ordering.

2.2 Refining the minimum aggregation

Two refinements are considered in this paper, called *discrimin* and *leximin*, see e.g. [3]. They allow to distinguish between vectors having the same minimal value.

Discrimin: Two evaluation vectors are compared using only their distinct components. Thus, identical values having the same place in both vectors are dropped before aggregating the remaining values with minimum. Thus, only discriminating term weights are considered. In the context of information retrieval, given two vectors representing the weights of terms in query q for documents d_1 and d_2, expressing term-document relevance. For instance, $r\vec{s}v(q, d_1) = (1, 0.5, 0.1, 0.3)$ and $r\vec{s}v(q, d_2) = (0.2, 0.7, 0.1, 1)$. Using *min* as an aggregation, these two vectors would get the same score. The *discrimin* procedure "drops" the third term, giving $rsv(q, d_1) = 0.3$ and $rsv(q, d_2) = 0.2$ and allowing to rank these documents.

Leximin: It is a *discrimin* applied on vectors with increasingly re-ordered components. Considering two vectors $r\vec{s}v(q, d_1) = (1, 0.5, 0.1, 0.2)$ and $r\vec{s}v(q, d_2) = (0.2, 0.7, 0.1, 1)$, the discrimin leads to the same evaluation. Since the leximin sorts the values before comparing them, the 0.2 values are also dropped, giving $rsv(q, d_2) = 0.7$ and $rsv(q, d_1) = 0.5$, thus ranking d_2 before d_1.

3 Possibilistic indexing and evaluation

In this paper, we will use the possibilistic model suggested in [7] and used in [2]. In this approach, the document relevance for the query is no longer given by the statistical index weight, but by a pair of possibility and necessity degrees computed from it. The retrieval status value is then a pair: $rsv(q, d) = (\Pi(q, d), N(q, d))$. The idea is to distinguish between two aspects of the relevance. If the weight of a term in a document is high enough, then this term is considered to be more or less certainly (or necessarily) representative of the content of the document. If the weight is not sufficiently high, then this term is considered only as possibly representative of the document. Thus, $rsv(q, d)$ represents to what extent it is certain and possible that d relevant with respect to q.

To use this possibilistic model, the possibility and necessity degrees of matching between the document and the query terms must be estimated taking into account the statistical weights of the terms in the document. Each document is therefore considered as a fuzzy set of terms (e.g. [5]) by normalizing the weights between $[0, 1]$. A simple, parametrized, way to assess the possibility and the necessity degrees (resp. Π and N) from the $tf * idf$ weight w_t of term t in document d once normalized is to use the following piecewise linear transformation:

$$\Pi(t, d) = \begin{cases} 0 & \text{if } w_t = 0 \\ 1 & \text{if } w_t \geq \alpha \\ \frac{w_t}{\alpha} & \text{otherwise} \end{cases} \qquad (1)$$

$$N = \begin{cases} 1 & \text{if } w_t = 1 \\ \frac{w_t - \alpha}{1 - \alpha} & \text{if } \alpha < 1 \text{ and } w_t \geq \alpha \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Note that when $\alpha = 0$, $\Pi = 1$ and $N = w_t$ and when $\alpha = 1$, $\Pi = w_t$ and $N = 0$.

Thus, the evaluation of a conjunctive query q involving $t_1, \ldots, t_n$ amounts to compute the pair of vectors $(\Pi(t_1, d), \ldots, \Pi(t_n, d))$ and $(N(t_1, d), \ldots, N(t_n, d))$. Then documents are ordered by applying first the leximin/discrimin ranking procedure on the N-vectors, and in case of ties, the leximin/discrimin is applied to the corresponding Π-vectors to try to refine the ordering.

Experiments on the impact of the possibilistic indexing and parameter α on the system performance are now reported.

4 Experimental results

In this section, we present results of some experiments on a subset of the CLEF2001 (`http://www.clef-campaign.org`) collection, to evaluate the merit of the vector-based ranking of documents. Moreover, the impact of the possibilistic encoding of the term weights in the document is first discussed.

4.1 Description of the experiments

The goal of the experiment is to enhance the global performance of the information retrieval system, and to compare the results that are obtained using several ranking methods with the ones provided by a classical approach. The first experiment therefore compares the use of the possibilistic framework in the matching process with the classical approach, in order to determine the best α value for the possibilistic encoding of the term weights. The second experiment compares results obtained with several conjunction aggregation operators, namely the weighted sum aggregation underlying the classical approach (used in Mercure [1]), the two defined $OWmin$ and the classical minimum, and with the refined leximin/discrimin-based ranking method.

4.1.1 The Mercure IR system

To index the collection, and to compare our results with a classical approach-based system, we used Mercure [1]. In this system, the weight w_t of a term for a document is computed using a formula derived from the *Okapi* system [8]: $w_t = \frac{tf}{0.2+0.7\times\frac{dl}{\Delta_l}+tf} \times (\log(\frac{n_{tot}}{n}))$, where tf is the term frequency in the document, dl is the document length, Δ_l is the average document length in the collection, n_{tot} is the size of the collection and n is the number of documents containing the term. In the collection used, we have $\Delta_l = 184.6$ and $n_{tot} = 113005$. The final similarity degree S_{qd} between a query q and a document d, giving the relevance of the document for the query, is computed as $S_{qd} = \sum_{t\in q} \lambda_t \times w_{td}$, where λ_t is an importance weight for the term in the query (here always 1) and w_{td} is the index term weight for document d, given by the previous equation.

4.1.2 CLEF collection

The collection used in this experimentation is the English part of the CLEF2001 collection, containing 113,005 articles from the 1994 *Los Angeles Times*. During the indexing stage, terms frequencies are computed for each document. These terms are stemmed using the Porter algorithm [6], and stop-words (i.e. words that bring no information) are removed. Together with the collection of documents, a set of topics, which are evaluated on the given documents by human experts, are available. These topics, identified by a number, are described by a title, a short description of the topic, and a narrative part giving precise relevance criteria. They are used as a basis for generating the queries to be evaluated by the IR system. Moreover, the documents estimated to be relevant by experts are provided for each topic.

As an example, the topic 41 is defined as: ***title:*** *Pesticides in Baby Food;* ***description:*** *Find reports on pesticides in baby food;* ***narrative part:*** *Relevant documents give information on the discovery of pesticides in baby food. They report on different brands, supermarkets, and companies selling baby food which contains pesticides. They also discuss measures against the contamination of baby food by pesticides.*

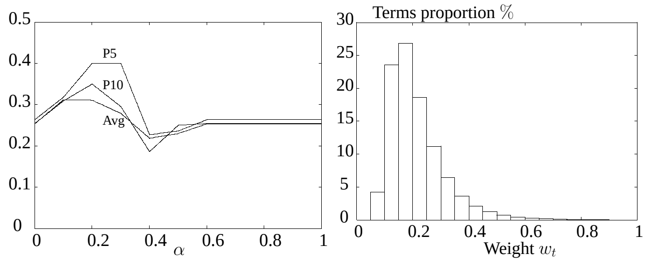
4.2 Evaluations and results

To evaluate the system, we used a set of 25 queries automatically built from the descriptive part of the CLEF topics, considered as keywords conjunctions. To estimate the quality of the information retrieval system, two measures are used. The recall is the ratio of relevant documents retrieved to those relevant in the collection, and the precision is the ratio of relevant documents among the documents retrieved. Since the precision at x, denoted Px, which is the ratio of relevant documents in the x first retrieved documents, is easier to estimate, it is usually used to represent the system performance. Precisions at 5, 10, etc. noted P5, P10, are thus computed. The average precision (AvgPr) is the average of the relevant documents precisions (i.e. Px with x the document's rank). The given values are averages on the evaluated queries results.

4.2.1 Effect of the possibilistic indexing

The first experiment uses the possibilistic degrees to estimate the relevance of the documents for the queries, to compare this approach with the classical one of the Mercure system. The aggregation of individual query terms is done using the *min* operator. Results of the possibilistic approach are shown in figure 1(a), and those of the Mercure system are P5: 0.3909; P10: 0.3682; AvgPr: 0.3827.

First of all, it can be noticed that we obtain the same precision value for $\alpha = 0$ and $\alpha = 1$, since this is equivalent as $\Pi = 1, N = w_t$ and $\Pi = w_t, N = 0$ respectively. The final ranking is therefore done using only the w_t's. However, in this case, the precision is lower than for the classical system. This is not surprising since the aggregation is done using the minimum (without leximin refinement) in place of the sum. This aggregation is indeed too coarse, justifying the use of refined methods, such as $OWmin$ that we will present in the following. However, whereas the average precisions are lower than what is obtained by the classical method, it is worth noticing that P5 is higher for $\alpha = 0.2$ and 0.3. Figure 1(b) shows the number of terms in the index for each value of w_t. Most of the terms have a weight around 0.2. Indeed, the conversion to possibility and necessity degrees has a stronger impact for values of α that share the terms ratio distribution in two approximately equal parts. Then more terms may be discriminated. As there are very few terms with w_t higher than 0.5, for such values of α the terms are discriminated only by Π or only by N, and the precision tends to be the one corresponding to $\alpha = 1$.



(a) Precision variation (b) Terms ratio

Figure 1: Interpretation of α values.

Thus, using a proper tuning of α and the coarse min aggregation, the method based on possibilistic degrees is still comparable to the classical approach based on statistical weights and weight sum. However, this improvement depends on the value of α, whose the optimal value depends on the terms repartition in the collection. Therefore, this value of α is collection dependent. Some other experiments should be done using other collections in order to estimate the collection effect on the performances.

4.2.2 Comparison of ranking methods

We now evaluate the ranking method presented in sections 2 and 3 w.r.t. the one used in Mercure. To apply these ordered weightings, the vectors containing the weights of each query term in the document are decreasingly ordered. As the queries considered here do not introduce further preference levels, the ordered vectors are then weighted using a kind of *most of*-like operator as in section 2.1, based on Dienes or Gödel implications. This type of operator gives more importance to the highest term weights, minimizing the impact of the lowest ones. The weighting vector is computed according to the query length l, $w_i = 1$ if $i \leq l/2$ and a decreasing linear function from 1 to 0 is used when i ranges between $l/2$ and l. Results are then sorted using this (Π, N) values modified by the weight w_i as in 2.1.

Moreover, the numerical precision of term degrees is not meaningful, since resulting from the normalization and the possibilistic transformation, which leads some values to differ only at the 5^{th} decimal. The possibilistic degrees used between terms and documents have therefore been rounded. The discrimin/leximin results depending on this rounding, several precision levels have been tested to estimate the impact of the rounding on the system performances. As in the previous evaluations, we used 25 queries, using different α values, rounding precision (i.e. the number of decimals kept), aggregating and ranking methods, to estimate the document relevance degrees. Table 1 shows the best P5 results, compared with the classical approach using statistical weights and the sum. Results using discrimin do not appear since they are not as good as those obtained with leximin. It should be noted that here, the better results are obtained for values of α different from the previous one. Therefore, the optimal value for this parameter depends not only on the term distribution in the collection, but also on the methods used to aggregate and sort the results. As expected, there is almost no

Table 1: Comparison of multicriteria methods.

α	Ranking method	Rounding decimal	P5	P10	AvgPr
0.1	leximin $+OW_D$	1	0.4273	0.3682	0.3572
0.1	leximin $+OW_G$	1	0.4273	0.3636	0.3571
0.6	leximin $+OW_D$	5	0.4273	0.3500	0.3161
0.6	leximin $+OW_D$	4	0.4273	0.3500	0.3158
0.6	leximin $+OW_D$	6	0.4273	0.3500	0.3152
0.8	leximin $+OW_D$	2	0.4273	0.3409	0.3209
0.1	leximin	1	0.4182	0.3545	0.3532
0.2	leximin $+OW_G$	2	0.4182	0.3545	0.3132
0.2	leximin $+OW_D$	2	0.4000	0.3409	0.3216
Mercure	sum		0.3909	0.3682	0.3827

performance difference between the two weighting techniques of section 2.1, denoted OW_G and OW_D in table 1.

Results are rather promising, since they are already better than the ones obtained with the standard ordering method on possibilistic degrees, based on *min*. Moreover, the best results are even better than those of the classical approach, which was our baseline here, improving P5 up to 9.3%. Nevertheless, the average precision is lower. As there is only few relevant documents in the collection for each query (about ten), the ordering method looses its effect for Pn with n rising, since this value is estimated by counting the relevant documents in the n firsts, whatever their position. Thus, the retrieved relevant documents are in the top of the list. The ranking method has a strong effect on the system performances. Moreover, the presented results are averages on the results obtained for 25 queries. The fact that P5 is better than with the classical approach whereas P10 is lower means that some queries are improved while other are degraded, and that the improvement is higher than the degradation. Moreover, in a realistic information retrieval system, such as web search engine, only the first retrieved documents are of interest determinant for the user, as she rarely browse through more than 10 results (which is often the default number of results by page displayed).

5 Conclusion

In this paper, we have presented a new approach to rank documents according to their relevance, using a refined vector-based rank-ordering method. This approach was evaluated on a subset of the CLEF2001 collection. We compared the refined rank-ordering approach (possibly using some ordered weighting method) with the classical approach based on relevance scores aggregated by a weighted sum. These experiments suggest the effectiveness of the refined rank-ordering approach. It outperforms sum or min-based aggregation methods to some extent. These first preliminary results indicate that ranking documents can take advantage of the full weights vector, rather than using an aggregated value. In future works, we plan to evaluate the approach on larger collections, such as TREC collections, and secondly to explore other variants of the approach.

References

[1] M. Boughanem, T. Dkaki, J. Mothe, and C. Soule-Dupuy. Mercure at TREC-7. In *Proc. of TREC-7*, pages 135–141, 1998.

[2] M. Boughanem, Y. Loiseau, and H. Prade. Graded pattern matching in a multilingual context. In *Proc. 7th Meeting Euro Working Group on Fuzzy Sets*, pages 121–126. Eurofuse, Varena, 2002.

[3] D. Dubois, H. Fargier, and H. Prade. Beyond min aggregation in multicriteria decision: (ordered) weighted min, discri-min, leximin. In R.R. Yager and J. Kacprzyk, editors, *The Ordered Weighted Averaging Operators*, pages 181–192. Kluwer, 1997.

[4] D. Dubois and H. Prade. Semantic of quotient operators in fuzzy relational databases. *Fuzzy Sets and Systems*, 78:89–93, 1996.

[5] D. Kraft, G. Bordogna, and G. Pasi. Fuzzy set techniques in information retrieval. In *Fuzzy Sets in Approximate Reasoning and Information Systems*, chapter 8, pages 469–510. Kluwer Academic Publishers, 1999.

[6] M.F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[7] H. Prade and C. Testemale. Application of possibility and necessity measures to documentary information retrieval. *LNCS*, 286:265–275, 1987.

[8] S. E. Robertson and S. Walker. Okapi-keenbow at TREC-8. In *Proc. 8th Text Retrieval Conf.*, pages 60–67. TREC-8, 1999.

[9] G. Salton, E.A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[10] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.