An Outranking Approach for Rank Aggregation in Information Retrieval

Mohamed Farah
Lamsade, Paris Dauphine University
Place du Mal de Lattre de Tassigny
75775 Paris Cedex 16, France
farah@lamsade.dauphine.fr

Daniel Vanderpooten
Lamsade, Paris Dauphine University
Place du Mal de Lattre de Tassigny
75775 Paris Cedex 16, France
vdp@lamsade.dauphine.fr

ABSTRACT

Research in Information Retrieval usually shows performance improvement when many sources of evidence are combined to produce a ranking of documents (e.g., texts, pictures, sounds, etc.). In this paper, we focus on the *rank aggregation problem*, also called *data fusion problem*, where rankings of documents, searched into the same collection and provided by multiple methods, are combined in order to produce a new ranking. In this context, we propose a rank aggregation method within a multiple criteria framework using aggregation mechanisms based on decision rules identifying positive and negative reasons for judging whether a document should get a better rank than another. We show that the proposed method deals well with the Information Retrieval distinctive features. Experimental results are reported showing that the suggested method performs better than the well-known CombSUM and CombMNZ operators.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval – *Retrieval models.*

General Terms: Algorithms, Measurement, Experimentation, Performance, Theory.

Keywords: Data fusion, Metasearch Engine, Multiple Criteria Approach, Outranking Methods, Rank Aggregation.

1. INTRODUCTION

A wide range of current Information Retrieval (IR) approaches are based on various search models (Boolean, Vector Space, Probabilistic, Language, etc. [2]) in order to retrieve relevant documents in response to a user request. The *result lists* produced by these approaches depend on the exact definition of the relevance concept.

Rank aggregation approaches, also called *data fusion* approaches, consist in combining these result lists in order to produce a new and hopefully better ranking. Such approaches give rise to metasearch engines in the Web context. We consider, in the following, cases where only ranks are available and no other additional information is provided such as the relevance scores. This corresponds indeed to the reality, where only ordinal information is available.

Data fusion is also relevant in other contexts, such as when the user writes several queries of his/her information need (e.g., a boolean query and a natural language query) [4], or when many document surrogates are available [16].

Several studies argued that rank aggregation has the potential of combining effectively all the various sources of evidence considered in various input methods. For instance, experiments carried out in [16], [30], [4] and [19] showed that documents which appear in the lists of the majority of the input methods are more likely to be relevant. Moreover, Lee [19] and Vogt and Cottrell [31] found that various retrieval approaches often return very different irrelevant documents, but many of the same relevant documents. Bartell *et al.* [3] also found that rank aggregation methods improve the performances w.r.t. those of the input methods, even when some of them have weak individual performances. These methods also tend to smooth out biases of the input methods according to Montague and Aslam [22]. Data fusion has recently been proved to improve performances for both the ad hoc retrieval and categorization tasks within the TREC genomics track in 2005 [1].

The rank aggregation problem was addressed in various fields such as *i)* in social choice theory which studies voting algorithms which specify winners of elections or winners of competitions in tournaments [29], *ii)* in statistics when studying correlation between rankings, *iii)* in distributed databases when results from different databases must be combined [12], and *iv)* in collaborative filtering [23].

Most current rank aggregation methods consider each input ranking as a permutation over the same set of items. They also give rigid interpretation to the exact ranking of the items. Both of these assumptions are rather not valid in the IR context, as will be shown in the following sections.

The remaining of the paper is organized as follows. We first review current rank aggregation methods in Section 2. Then we outline the specificities of the data fusion problem in the IR context (Section 3). In Section 4, we present a new aggregation method which is proven to best fit the IR context. Experimental results are presented in Section 5 and conclusions are provided in a final section.

2. RELATED WORK

As pointed out by Riker [25], we can distinguish two families of rank aggregation methods: *positional methods* which assign scores to items to be ranked according to the ranks

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

they receive and *majoritarian methods* which are based on pairwise comparisons of items to be ranked. These two families of methods find their roots in the pioneering works of Borda [5] and Condorcet [7], respectively, in the social choice literature.

2.1 Preliminaries

We first introduce some basic notations to present the rank aggregation methods in a uniform way. Let $D = \{d_1, d_2, \ldots, d_{n_d}\}$ be a set of n_d documents. A *list* or a *ranking* $\succ_j$ is an ordering defined on $D_j \subseteq D$ $(j = 1, \ldots, n)$. Thus, $d_i \succ_j d_{i'}$ means d_i '*is ranked better than*' $d_{i'}$ in $\succ_j$. When $D_j = D$, $\succ_j$ is said to be a *full list*. Otherwise, it is a *partial list*. If d_i belongs to D_j, r_i^j denotes the *rank* or *position* of d_i in $\succ_j$. We assume that the best answer (document) is assigned the position 1 and the worst one is assigned the position $|D_j|$. Let $\Re_D$ be the set of all permutations on D or all subsets of D. A *profile* is a n-tuple of rankings $PR = (\succ_1, \succ_2, \ldots, \succ_n)$. Restricting PR to the rankings containing document d_i defines PR_i. We also call the number of rankings which contain document d_i the *rank hits* of d_i [19].

The rank aggregation or data fusion problem consists of finding a ranking function or mechanism Ψ (also called a *social welfare function* in the social choice theory terminology) defined by:

$$\Psi : \begin{array}{lcl} \Re_D^n & \rightarrow & \Re_D \\ PR = (\succ_1, \succ_2, \ldots, \succ_n) & \rightarrow & \sigma = \Psi(PR) \end{array}$$

where σ is called a *consensus ranking*.

2.2 Positional Methods

2.2.1 Borda Count

This method [5] first assigns a score $\sum_{j=1}^{n} r_i^j$ to each document d_i. Documents are then ranked by increasing order of this score, breaking ties, if any, arbitrarily.

2.2.2 Linear Combination Methods

This family of methods basically combine scores of documents. When used for the rank aggregation problem, ranks are assumed to be scores or performances to be combined using aggregation operators such as the weighted sum or some variation of it [3, 31, 17, 28].

For instance, Callan *et al.* [6] used the inference networks model [30] to combine rankings. Fox and Shaw [15] proposed several combination strategies which are CombSUM, CombMIN, CombMAX, CombANZ and CombMNZ. The first three operators correspond to the sum, min and max operators, respectively. CombANZ and CombMNZ respectively divides and multiplies the CombSUM score by the rank hits. It is shown in [19] that the CombSUM and CombMNZ operators perform better than the others. Metasearch engines such as SavvySearch and MetaCrawler use the CombSUM strategy to fuse rankings.

2.2.3 Footrule Optimal Aggregation

In this method, a consensus ranking minimizes the Spearman footrule distance from the input rankings [21]. Formally, given two full lists $\succ_j$ and $\succ_{j'}$, this distance is given by $F(\succ_j, \succ_{j'}) = \sum_{i=1}^{n_d} |r_i^j - r_i^{j'}|$. It extends to several lists as follows. Given a profile PR and a consensus ranking σ, the Spearman footrule distance of σ to PR is given by $F(\sigma, PR) = \sum_{j=1}^{n} F(\sigma, \succ_j)$.

Cook and Kress [8] proposed a similar method which consists in optimizing the distance $D(\succ_j, \succ_{j'}) = \frac{1}{2} \sum_{i,i'=1}^{n_d} |r_{i,i'}^j - r_{i,i'}^{j'}|$, where $r_{i,i'}^j = r_{i'}^j - r_i^j$. This formulation has the advantage that it considers the intensity of preferences.

2.2.4 Probabilistic Methods

This kind of methods assume that the performance of the input methods on a number of training queries is indicative of their future performance. During the training process, probabilities of relevance are calculated. For subsequent queries, documents are ranked based on these probabilities. For instance, in [20], each input ranking $\succ_j$ is divided into a number of segments, and the conditional probability of relevance (R) of each document d_i depending on the segment k it occurs in, is computed, i.e. $prob(R|d_i, k, \succ_j)$. For subsequent queries, the score of each document d_i is given by $\sum_{j=1}^{n} \frac{prob(R|d_i,k,\succ_j)}{k}$. Le Calve and Savoy [18] suggest using a logistic regression approach for combining scores. Training data is needed to infer the model parameters.

2.3 Majoritarian Methods

2.3.1 Condorcet Procedure

The original Condorcet rule [7] specifies that a winner of the election is any item that beats or ties with every other item in a pairwise contest. Formally, let $C(d_i \sigma d_{i'}) = \{\succ_j \in PR : d_i \succ_j d_{i'}\}$ be the *coalition* of rankings that are *concordant* with establishing $d_i \sigma d_{i'}$, i.e. with the proposition d_i '*should be ranked better than*' $d_{i'}$ in the final ranking σ. d_i beats or ties with $d_{i'}$ iff $|C(d_i \sigma d_{i'})| \geq |C(d_{i'} \sigma d_i)|$.

The repetitive application of the Condorcet algorithm can produce a ranking of items in a natural way: select the Condorcet winner, remove it from the lists, and repeat the previous two steps until there are no more documents to rank. Since there is not always Condorcet winners, variations of the Condorcet procedure have been developed within the multiple criteria decision aid theory, with methods such as ELECTRE [26].

2.3.2 Kemeny Optimal Aggregation

As in section 2.2.3, a consensus ranking minimizes a geometric distance from the input rankings, where the Kendall tau distance is used instead of the Spearman footrule distance. Formally, given two full lists $\succ_j$ and $\succ_{j'}$, the Kendall tau distance is given by $K(\succ_j, \succ_{j'}) = |\{(d_i, d_{i'}) : i < i', r_i^j < r_{i'}^j, r_i^{j'} > r_{i'}^{j'}\}|$, i.e. the number of pairwise disagreements between the two lists. It is easy to show that the consensus ranking corresponds to the geometric median of the input rankings and that the Kemeny optimal aggregation problem corresponds to the minimum feedback edge set problem.

2.3.3 Markov Chain Methods

Markov chains (MCs) have been used by Dwork *et al.* [11] as a 'natural' method to obtain a consensus ranking where states correspond to the documents to be ranked and the transition probabilities vary depending on the interpretation of the transition event. In the same reference, the authors proposed four specific MCs and experimental testing had shown that the following MC is the best performing one (see also [24]):

- MC4: move from the current state d_i to the next state $d_{i'}$ by first choosing a document $d_{i'}$ uniformly from D. If for the majority of the rankings, we have $r^j_{i'} \leq r^j_i$, then move to $d_{i'}$, else stay in d_i.

The consensus ranking corresponds to the stationary distribution of MC4.

3. SPECIFICITIES OF THE RANK AGGREGATION PROBLEM IN THE IR CONTEXT

3.1 Limited Significance of the Rankings

The exact positions of documents in one input ranking have limited significance and should not be overemphasized. For instance, having three relevant documents in the first three positions, any perturbation of these three items will have the same value. Indeed, in the IR context, the *complete order* provided by an input method may hide *ties*. In this case, we call such rankings *semi orders*. This was outlined in [13] as the problem of aggregation with ties. It is therefore important to build the consensus ranking based on *robust information*:

- Documents with near positions in $\succ_j$ are more likely to have similar interest or relevance. Thus a slight perturbation of the initial ranking is meaningless.

- Assuming that document d_i is better ranked than document $d_{i'}$ in a ranking $\succ_j$, d_i is more likely to be definitively more relevant than $d_{i'}$ in $\succ_j$ when the number of intermediate positions between d_i and $d_{i'}$ increases.

3.2 Partial Lists

In real world applications, such as metasearch engines, rankings provided by the input methods are often partial lists. This was outlined in [14] as the problem of having to merge top-k results from various input lists. For instance, in the experiments carried out by Dwork *et al.* [11], authors found that among the top 100 best documents of 7 input search engines, 67% of the documents were present in only one search engine, whereas less than two documents were present in all the search engines.

Rank aggregation of partial lists raises four major difficulties which we state hereafter, proposing for each of them various working assumptions:

1. Partial lists can have various lengths, which can favour long lists. We thus consider the following two working hypotheses:
   H^1_k: We only consider the top k best documents from each input ranking.
   H^1_{all}: We consider all the documents from each input ranking.

2. Since there are different documents in the input rankings, we must decide which documents should be kept in the consensus ranking. Two working hypotheses are therefore considered:
   H^2_k: We only consider documents which are present in at least k input rankings ($k > 1$).
   H^2_{all}: We consider all the documents which are ranked in at least one input ranking.
   Hereafter, we call documents which will be retained in the consensus ranking, *candidate documents*, and documents that will be excluded from the consensus ranking, *excluded documents*. We also call a candidate document which is missing in one or more rankings, a *missing document*.

3. Some candidate documents are missing documents in some input rankings. Main reasons for a missing document are that it was not indexed or it was indexed but deemed irrelevant ; usually this information is not available. We consider the following two working hypotheses:
   H^3_{yes}: Each missing document in each $\succ_j$ is assigned a position.
   H^3_{no}: No assumption is made, that is each missing document is considered neither better nor worse than any other document.

4. When assumption H^2_k holds, each input ranking may contain documents which will not be considered in the consensus ranking. Regarding the positions of the candidate documents, we can consider the following working hypotheses:
   H^4_{init}: The initial positions of candidate documents are kept in each input ranking.
   H^4_{new}: Candidate documents receive new positions in each input ranking, after discarding excluded ones.

In the IR context, rank aggregation methods need to decide more or less explicitly which assumptions to retain w.r.t. the above-mentioned difficulties.

4. OUTRANKING APPROACH FOR RANK AGGREGATION

4.1 Presentation

Positional methods consider implicitly that the positions of the documents in the input rankings are *scores* giving thus a cardinal meaning to an ordinal information. This constitutes a strong assumption that is questionable, especially when the input rankings have different lengths. Moreover, for positional methods, assumptions H^3 and H^4, which are often arbitrary, have a strong impact on the results. For instance, let us consider an input ranking of 500 documents out of 1000 candidate documents. Whether we assign to each of the missing documents the position 1, 501, 750 or 1000 -corresponding to variations of H^3_{yes}- will give rise to very contrasted results, especially regarding the top of the consensus ranking.

Majoritarian methods do not suffer from the above-mentioned drawbacks of the positional methods since they build consensus rankings exploiting only *ordinal information* contained in the input rankings. Nevertheless, they suppose that such rankings are complete orders, ignoring that they may hide ties. Therefore, majoritarian methods base consensus rankings on illusory discriminant information rather than less discriminant but more robust information.

Trying to overcome the limits of current rank aggregation methods, we found that *outranking approaches*, which were initially used for multiple criteria aggregation problems [26], can also be used for the rank aggregation purpose, where each ranking plays the role of a criterion. Therefore, in order to decide whether a document d_i should be ranked better than $d_{i'}$ in the consensus ranking σ, the two following conditions should be met:

- a *concordance* condition which ensures that a *majority* of the input rankings are concordant with $d_i\sigma d_{i'}$ (*majority principle*).
- a *discordance* condition which ensures that none of the discordant input rankings strongly refutes $d\sigma d'$ (*respect of minorities principle*).

Formally, the *concordance coalition* with $d_i\sigma d_{i'}$ is

$$C_{s_p}(d_i\sigma d_{i'}) \quad = \quad \{\succ_j \in PR : r_i^j \leq r_{i'}^j - s_p\}$$

where s_p is a *preference threshold* which is the variation of document positions -whether it is absolute or relative to the ranking length- which draws the boundaries between an *indifference* and a *preference situation* between documents.

The *discordance coalition* with $d_i\sigma d_{i'}$ is

$$D_{s_v}(d_i\sigma d_{i'}) \quad = \quad \{\succ_j \in PR : r_i^j \geq r_{i'}^j + s_v\}$$

where s_v is a *veto threshold* which is the variation of document positions -whether it is absolute or relative to the ranking length- which draws the boundaries between a *weak* and a *strong opposition* to $d_i\sigma d_{i'}$.

Depending on the exact definition of the preceding concordance and discordance coalitions leading to the definition of some *decision rules*, several outranking relations can be defined. They can be more or less demanding depending on *i)* the values of the thresholds s_p and s_v, *ii)* the importance or minimal size c_{min} required for the concordance coalition, and *iii)* the importance or maximum size d_{max} of the discordance coalition.

A generic outranking relation can thus be defined as follows:

$$\begin{array}{rcl} d_i S_{(s_p,s_v,c_{min},d_{max})} d_{i'} & \Leftrightarrow & |C_{s_p}(d_i\sigma d_{i'})| \geq c_{min} \\ & AND & |D_{s_v}(d_i\sigma d_{i'})| \leq d_{max} \end{array}$$

This expression defines a family of *nested* outranking relations since $S_{(s_p,s_v,c_{min},d_{max})} \subseteq S_{(s'_p,s'_v,c'_{min},d'_{max})}$ when $c_{min} \geq c'_{min}$ and/or $d_{max} \leq d'_{max}$ and/or $s_p \geq s'_p$ and/or $s_v \leq s'_v$. This expression also generalizes the majority rule which corresponds to the particular relation $S_{(0,\infty,\frac{n}{2},n)}$. It also satisfies important properties of rank aggregation methods, called neutrality, Pareto-optimality, Condorcet property and Extended Condorcet property, in the social choice literature [29].

Outranking relations are not necessarily transitive and do not necessarily correspond to rankings since directed cycles may exist. Therefore, we need specific procedures in order to derive a consensus ranking. We propose the following procedure which finds its roots in [27]. It consists in partitioning the set of documents into r ranked *classes*.

Each class C_h contains documents with the same relevance and results from the application of all relations (if possible) to the set of documents remaining after previous classes are computed. Documents within the same equivalence class are ranked arbitrarily.

Formally, let

- R be the set of candidate documents for a query,
- $S^1, S^2, \ldots$ be a family of nested outranking relations,
- $F_k(d_i, E) = |\{d_{i'} \in E : d_i S^k d_{i'}\}|$ be the number of documents in $E (E \subseteq R)$ that could be considered 'worse' than d_i according to relation S^k,
- $f_k(d_i, E) = |\{d_{i'} \in E : d_{i'} S^k d_i\}|$ be the number of documents in E that could be considered 'better' than d_i according to S^k,
- $s_k(d_i, E) = F_k(d_i, E) - f_k(d_i, E)$ be the *qualification* of d_i in E according to S^k.

Each class C_h results from a *distillation process*. It corresponds to the last distillate of a series of sets $E_0 \supseteq E_1 \supseteq \ldots$ where $E_0 = R \setminus (C_1 \cup \ldots \cup C_{h-1})$ and E_k is a reduced subset of E_{k-1} resulting from the application of the following procedure:

1. compute for each $d_i \in E_{k-1}$ its qualification according to S^k, i.e. $s_k(d_i, E_{k-1})$,
2. define $s_{\max} = \max_{d_i \in E_{k-1}}\{s_k(d_i, E_{k-1})\}$, then
3. $E_k = \{d_i \in E_{k-1} : s_k(d_i, E_{k-1}) = s_{\max}\}$

When one outranking relation is used, the distillation process stops after the first application of the previous procedure, i.e., C_h corresponds to distillate E_1. When different outranking relations are used, the distillation process stops when all the pre-defined outranking relations have been used or when $|E_k| = 1$.

4.2 Illustrative Example

This section illustrates the concepts and procedures of section 4.1. Let us consider a set of candidate documents $R = \{d_1, d_2, d_3, d_4, d_5\}$. The following table gives a profile PR of different rankings of the documents of R: $PR = (\succ_1, \succ_2, \succ_3, \succ_4)$.

Table 1: Rankings of documents

r_i^j	$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$
d_1	1	3	1	5
d_2	2	1	3	3
d_3	3	2	2	1
d_4	4	4	5	2
d_5	5	5	4	4

Let us suppose that the preference and veto thresholds are set to values 1 and 4 respectively, and that the concordance and discordance thresholds are set to values 2 and 1 respectively. The following tables give the concordance, discordance and outranking matrices. Each entry $c_{s_p}(d_i, d_{i'})$ ($d_{s_v}(d_i, d_{i'})$) in the concordance (discordance) matrix gives the number of rankings that are concordant (discordant) with $d_i\sigma d_{i'}$, i.e. $c_{s_p}(d_i, d_{i'}) = |C_{s_p}(d_i\sigma d_{i'})|$ and $d_{s_v}(d_i, d_{i'}) = |D_{s_v}(d_i\sigma d_{i'})|$.

Table 2: Computation of the outranking relation

	d_1	d_2	d_3	d_4	d_5
d_1	-	2	2	3	3
d_2	2	-	2	3	4
d_3	2	2	-	4	4
d_4	1	1	0	-	3
d_5	1	0	0	1	-

Concordance Matrix

	d_1	d_2	d_3	d_4	d_5
d_1	-	0	1	0	0
d_2	0	-	0	0	0
d_3	0	0	-	0	0
d_4	1	0	0	-	0
d_5	1	1	0	0	-

Discordance Matrix

	d_1	d_2	d_3	d_4	d_5
d_1	-	1	1	1	1
d_2	1	-	1	1	1
d_3	1	1	-	1	1
d_4	0	0	0	-	1
d_5	0	0	0	0	-

Outranking Matrix (S^1)

For instance, the concordance coalition for the assertion $d_1\sigma d_4$ is $C_1(d_1\sigma d_4) = \{\succ_1, \succ_2, \succ_3\}$ and the discordance coalition for the same assertion is $D_4(d_1\sigma d_4) = \emptyset$. Therefore, $c_1(d_1, d_4) = 3$, $d_4(d_1, d_4) = 0$ and $d_1 S^1 d_4$ holds.

Notice that $F_k(d_i, R)$ ($f_k(d_i, R)$) is given by summing the values of the i^{th} row (column) of the outranking matrix. The

consensus ranking is obtained as follows: to get the first class C_1, we compute the qualifications of all the documents of $E_0 = R$ with respect to S^1. They are respectively 2, 2, 2, -2 and -4. Therefore s_{max} equals 2 and $C_1 = E_1 = \{d_1, d_2, d_3\}$. Observe that, if we had used a second outranking relation $S_2(\supseteq S_1)$, these three documents could have been possibly discriminated. At this stage, we remove documents of C_1 from the outranking matrix and compute the next class C_2: we compute the new qualifications of the documents of $E_0 = R \setminus C_1 = \{d_4, d_5\}$. They are respectively 1 and -1. So $C_3 = E_1 = \{d_4\}$. The last document d_5 is the only document of the last class C_3. Thus, the consensus ranking is $\{d_1, d_2, d_3\} \rightarrow \{d_4\} \rightarrow \{d_5\}$.

5. EXPERIMENTS AND RESULTS

5.1 Test Setting

To facilitate empirical investigation of the proposed methodology, we developed a prototype metasearch engine that implements a version of our outranking approach for rank aggregation. In this paper, we apply our approach to the Topic Distillation (TD) task of TREC-2004 Web track [10]. In this task, there are 75 topics where only a short description of each is given. For each query, we retained the rankings of the 10 best runs of the TD task which are provided by TREC-2004 participating teams. The performances of these runs are reported in table 3.

Table 3: Performances of the 10 best runs of the TD task of TREC-2004

Run Id	MAP	P@10	S@1	S@5	S@10
uogWebCAU150	17.9%	24.9%	50.7%	77.3%	89.3%
MSRAmixed1	17.8%	25.1%	38.7%	72.0%	88.0%
MSRC04C12	16.5%	23.1%	38.7%	74.7%	80.0%
humW04rdpl	16.3%	23.1%	37.3%	78.7%	90.7%
THUIRmix042	14.7%	20.5%	21.3%	58.7%	74.7%
UAmsT04MWScb	14.6%	20.9%	36.0%	66.7%	76.0%
ICT04CIIS1AT	14.1%	20.8%	33.3%	64.0%	78.7%
SJTUINCMIX5	12.9%	18.9%	29.3%	57.3%	72.0%
MU04web1	11.5%	19.9%	33.3%	64.0%	76.0%
MeijiHILw3	11.5%	15.3%	30.7%	54.7%	64.0%
Average	14.7%	21.2%	34.9%	66.8%	78.94%

For each query, each run provides a ranking of about 1000 documents. The number of documents retrieved by all these runs ranges from 543 to 5769. Their average (median) number is 3340 (3386). It is worth noting that we found similar distributions of the documents among the rankings as in [11].

For evaluation, we used the 'trec_eval' standard tool which is used by the TREC community to calculate the standard measures of system effectiveness which are Mean Average Precision (MAP) and Success@n (S@n) for n=1, 5 and 10.

Our approach effectiveness is compared against some high performing official results from TREC-2004 as well as against some standard rank aggregation algorithms. In the experiments, significance testing is mainly based on the t-student statistic which is computed on the basis of the MAP values of the compared runs. In the tables of the following section, statistically significant differences are marked with an asterisk. Values between brackets of the first column of each table, indicate the parameter value of the corresponding run.

5.2 Results

We carried out several series of runs in order to *i)* study performance variations of the outranking approach when tuning the parameters and working assumptions, *ii)* compare performances of the outranking approach vs standard rank aggregation strategies , and *iii)* check whether rank aggregation performs better than the best input rankings.

We set our basic run mcm with the following parameters. We considered that each input ranking is a complete order ($s_p = 0$) and that an input ranking strongly refutes $d_i \sigma d_{i'}$ when the difference of both document positions is large enough ($s_v = 75\%$). Preference and veto thresholds are computed proportionally to the number of documents retained in each input ranking. They consequently may vary from one ranking to another. In addition, to accept the assertion $d_i \sigma d_{i'}$, we supposed that the majority of the rankings must be concordant ($c_{min} = 50\%$) and that every input ranking can impose its veto ($d_{max} = 0$). Concordance and discordance thresholds are computed for each tuple $(d_i, d_{i'})$ as the percentage of the input rankings of $PR_i \cap PR_{i'}$. Thus, our choice of parameters leads to the definition of the outranking relation $S_{(0,75\%,50\%,0)}$.

To test the run mcm, we had chosen the following assumptions. We retained the top 100 best documents from each input ranking (H^1_{100}), only considered documents which are present in at least half of the input rankings (H^2_5) and assumed H^3_{no} and H^4_{new}. In these conditions, the number of successful documents was about 100 on average, and the computation time per query was less than one second.

Obviously, modifying the working assumptions should have deeper impact on the performances than tuning our model parameters. This was validated by preliminary experiments. Thus, we hereafter begin by studying performance variation when different sets of assumptions are considered. Afterwards, we study the impact of tuning parameters. Finally, we compare our model performances w.r.t. the input rankings as well as some standard data fusion algorithms.

5.2.1 Impact of the Working Assumptions

Table 4 summarizes the performance variation of the outranking approach under different working hypotheses. In this table, we first show that run mcm22, in which missing documents are all put in the same last position of each input ranking, leads to performance drop w.r.t. run mcm. Moreover, S@1 moves from 41.33% to 34.67% (-16.11%). This shows that several relevant documents which were initially put at the first position of the consensus ranking in mcm, lose this first position but remain ranked in the top 5 documents since S@5 did not change. We also conclude that documents which have rather good positions in some input rankings are more likely to be relevant, even though they are missing in some other rankings. Consequently, when they are missing in some rankings, assigning worse ranks to these documents is harmful for performance.

Table 4: Impact of the working assumptions

Run Id	MAP	S@1	S@5	S@10
mcm	18.47%	41.33%	81.33%	86.67%
mcm22 (H^3_{yes})	17.72% (-4.06%)	34.67%	81.33%	86.67%
mcm23 (H^4_{init})	18.26% (-1.14%)	41.33%	81.33%	86.67%
mcm24 (H^1_{all})	20.67% (+11.91%*)	38.66%	80.00%	86.66%
mcm25 (H^2_{all})	21.68% (+17.38%*)	40.00%	78.66%	89.33%

Also, from Table 4, we found that the performances of runs mcm and mcm23 are similar. Therefore, the outranking approach is not sensitive to keeping the initial positions of candidate documents or recomputing them by discarding excluded ones.

From the same Table 4, performance of the outranking approach increases significantly for runs mcm24 and mcm25. Therefore, whether we consider all the documents which are present in half of the rankings (mcm24) or we consider all the documents which are ranked in the first 100 positions in one or more rankings (mcm25), increases performances. This result was predictable since in both cases we have more detailed information on the relative importance of documents. Tables 5 and 6 confirm this evidence. Table 5, where values between brackets of the first column give the number of documents which are retained from each input ranking, shows that selecting more documents from each input ranking leads to performance increase. It is worth mentioning that selecting more than 600 documents from each input ranking does not improve performance.

Table 5: Impact of the number of retained documents

Run Id	MAP	S@1	S@5	S@10
mcm (100)	18.47%	41.33%	81.33%	86.67%
mcm24-1 (200)	19.32% (+4.60%)	42.67%	78.67%	88.00%
mcm24-2 (400)	19.88% (+7.63%*)	37.33%	80.00%	88.00%
mcm24-3 (600)	20.80% (+12.62%*)	40.00%	80.00%	88.00%
mcm24-4 (800)	20.66% (+11.86%*)	40.00%	78.67%	86.67%
mcm24 (1000)	20.67% (+11.91%*)	38.66%	80.00%	86.66%

Table 6 reports runs corresponding to variations of H_k^2. Values between brackets are rank hits. For instance, in the run mcm32, only documents which are present in 3 or more input rankings, were considered successful. This table shows that performance is significantly better when rare documents are considered, whereas it decreases significantly when these documents are discarded. Therefore, we conclude that many of the relevant documents are retrieved by a rather small set of IR models.

Table 6: Performance considering different rank hits

Run Id	MAP	S@1	S@5	S@10
mcm25 (1)	21.68% (+17.38%*)	40.00%	78.67%	89.33%
mcm32 (3)	18.98% (+2.76%)	38.67%	80.00%	85.33%
mcm (5)	18.47%	41.33%	81.33%	86.67%
mcm33 (7)	15.83% (-14.29%*)	37.33%	78.67%	85.33%
mcm34 (9)	10.96% (-40.66%*)	36.11%	66.67%	70.83%
mcm35 (10)	7.42% (-59.83%*)	39.22%	62.75%	64.70%

For both runs mcm24 and mcm25, the number of successful documents was about 1000 and therefore, the computation time per query increased and became around 5 seconds.

5.2.2 Impact of the Variation of the Parameters

Table 7 shows performance variation of the outranking approach when different preference thresholds are considered. We found performance improvement up to threshold values of about 5%, then there is a decrease in the performance which becomes significant for threshold values greater than 10%. Moreover, S@1 improves from 41.33% to 46.67% when preference threshold changes from 0 to 5%. We can thus conclude that the input rankings are semi orders rather than complete orders.

Table 8 shows the evolution of the performance measures w.r.t. the concordance threshold. We can conclude that in order to put document d_i before $d_{i'}$ in the consensus ranking, at least half of the input rankings of $PR_i \cap PR_{i'}$ should be concordant. Performance drops significantly for very low and very high values of the concordance threshold. In fact, for such values, the concordance condition is either fulfilled rather always by too many document pairs or not fulfilled at all, respectively. Therefore, the outranking relation becomes either too weak or too strong respectively.

Table 7: Impact of the variation of the preference threshold from 0 to 12.5%

Run Id	MAP	S@1	S@5	S@10
mcm (0%)	18.47%	41.33%	81.33%	86.67%
mcm1 (1%)	18.57% (+0.54%)	41.33%	81.33%	86.67%
mcm2 (2.5%)	18.63% (+0.87%)	42.67%	78.67%	86.67%
mcm3 (5%)	18.69% (+1.19%)	46.67%	81.33%	86.67%
mcm4 (7.5%)	18.24% (-1.25%)	46.67%	81.33%	86.67%
mcm5 (10%)	17.93% (-2.92%)	40.00%	82.67%	86.67%
mcm5b (12.5%)	17.51% (-5.20%*)	41.33%	80.00%	86.67%

Table 8: Impact of the variation of c_{min}

Run Id	MAP	S@1	S@5	S@10
mcm11 (20%)	17.63% (-4.55%*)	41.33%	76.00%	85.33%
mcm12 (40%)	18.37% (-0.54%)	42.67%	76.00%	86.67%
mcm (50%)	18.47%	41.33%	81.33%	86.67%
mcm13 (60%)	18.42% (-0.27%)	40.00%	78.67%	86.67%
mcm14 (80%)	17.43% (-5.63%*)	40.00%	78.67%	86.67%
mcm15 (100%)	16.12% (-12.72%*)	41.33%	70.67%	85.33%

In the experiments, varying the veto threshold as well as the discordance threshold *within reasonable intervals* does not have significant impact on performance measures. In fact, runs with different veto thresholds ($s_v \in [50\%; 100\%]$) had similar performances even though there is a slight advantage for runs with high threshold values which means that it is better not to allow the input rankings to put their veto easily. Also, tuning the discordance threshold was carried out for values 50% and 75% of the veto threshold. For these runs we did not get any noticeable performance variation, although for low discordance thresholds ($d_{max} < 20\%$), performance slightly decreased.

5.2.3 Impact of the Variation of the Number of Input Rankings

To study performance evolution when different sets of input rankings are considered, we carried three more runs where 2, 4, and 6 of the best performing sets of the input rankings are considered. Results reported in Table 9 are seemingly counter-intuitive and also do not support previous findings regarding rank aggregation research [3]. Nevertheless, this result shows that low performing rankings bring more noise than information to the establishment of the consensus ranking. Therefore, when they are considered, performance decreases.

Table 9: Performance considering different best performing sets of input rankings

Run Id	MAP	S@1	S@5	S@10
mcm (10)	18.47%	41.33%	81.33%	86.67%
mcm27 (6)	18.60% (+0.70%)	41.33%	80.00%	85.33%
mcm28 (4)	19.02% (+2.98%)	40.00%	86.67%	88.00%
mcm29 (2)	18.33% (-0.76%)	44.00%	76.00%	88.00%

5.2.4 Comparison of the Performance of Different Rank Aggregation Methods

In this set of runs, we compare the outranking approach with some standard rank aggregation methods which were

proven to have acceptable performance in previous studies: we considered two positional methods which are the CombSUM and the CombMNZ strategies. We also examined the performance of one majoritarian method which is the Markov chain method (MC4). For the comparisons, we considered a specific outranking relation $S^* = S_{(5\%,50\%,50\%,30\%)}$ which results in good overall performances when tuning all the parameters.

The first row of Table 10 gives performances of the rank aggregation methods w.r.t. a basic assumption set $A_1 = (H^1_{100}, H^2_5, H^4_{new})$: we only consider the 100 first documents from each ranking, then retain documents present in 5 or more rankings and update ranks of successful documents. For positional methods, we place missing documents at the queue of the ranking (H^3_{yes}) whereas for our method as well as for MC4, we retained hypothesis H^3_{no}. The three following rows of Table 10 report performances when changing one element from the basic assumption set: the second row corresponds to the assumption set $A_2 = (H^1_{1000}, H^2_5, H^4_{new})$, i.e. changing the number of retained documents from 100 to 1000. The third row corresponds to the assumption set $A_3 = (H^1_{100}, H^2_{all}, H^4_{new})$, i.e. considering the documents present in at least one ranking. The fourth row corresponds to the assumption set $A_4 = (H^1_{100}, H^2_5, H^4_{init})$, i.e. keeping the original ranks of successful documents.

The fifth row of Table 10, labeled A_5, gives performance when all the 225 queries of the Web track of TREC-2004 are considered. Obviously, performance level cannot be compared with previous lines since the additional queries are different from the TD queries and correspond to other tasks (Home Page and Named Page tasks [10]) of TREC-2004 Web track. This set of runs aims to show whether *relative* performance of the various methods is task-dependent.

The last row of Table 10, labeled A_6, reports performance of the various methods considering the TD task of TREC-2002 instead of TREC-2004: we fused the results of input rankings of the 10 best official runs for each of the 50 TD queries [9] considering the set of assumptions A_1 of the first row. This aims to show whether *relative* performance of the various methods changes from year to year.

Values between brackets of Table 10 are variations of performance of each rank aggregation method w.r.t. performance of the outranking approach.

Table 10: Performance (MAP) of different rank aggregation methods under 3 different test collections

	mcm	combSUM	combMNZ	markov
A_1	18.79%	17.54% (-6.65%*)	17.08% (-9.10%*)	18.63% (-0.85%)
A_2	21.36%	19.18% (-10.21%*)	18.61% (-12.87%*)	21.33% (-0.14%)
A_3	21.92%	21.38% (-2.46%)	20.88% (-4.74%)	19.35% (-11.72%*)
A_4	18.64%	17.58% (-5.69%*)	17.18% (-7.83%*)	18.63% (-0.05%)
A_5	55.39%	52.16% (-5.83%*)	49.70% (-10.27%*)	53.30% (-3.77%)
A_6	16.95%	15.65% (-7.67%*)	14.57% (-14.04%*)	16.39% (-3.30%)

From the analysis of table 10 the following can be established:

- for all the runs, considering all the documents in each input ranking (A_2) significantly improves performance (MAP increases by 11.62% on average). This is predictable since some initially unreported relevant documents would receive better positions in the consensus ranking.
- for all the runs, considering documents even those present in only one input ranking (A_3) significantly improves performance. For mcm, combSUM and combMNZ, performance improvement is more important (MAP increases by 20.27% on average) than for the markov run (MAP increases by 3.86%).
- preserving the initial positions of documents (A_4) or recomputing them (A_1) does not have a noticeable influence on performance for both positional and majoritarian methods.
- considering all the queries of the Web track of TREC-2004 (A_5) as well as the TD queries of the Web track of TREC-2002 (A_6) does not alter the *relative* performance of the different data fusion methods.
- considering the TD queries of the Web track of TREC-2002, performances of all the data fusion methods are lower than that of the best performing input ranking for which the MAP value equals 18.58%. This is because most of the fused input rankings have *very* low performances compared to the best one, which brings more noise to the consensus ranking.
- performances of the data fusion methods mcm and markov are significantly better than that of the best input ranking uogWebCAU150. This remains true for runs combSUM and combMNZ only under assumptions H^1_{all} or H^2_{all}. This shows that majoritarian methods are less sensitive to assumptions than positional methods.
- outranking approach always performs significantly better than positional methods combSUM and combMNZ. It has also better performances than the Markov chain method, especially under assumption H^2_{all} where difference of performances becomes significant.

6. CONCLUSIONS

In this paper, we address the rank aggregation problem where different, but not disjoint, lists of documents are to be fused. We noticed that the input rankings can hide ties, so they should not be considered as complete orders. Only robust information should be used from each input ranking.

Current rank aggregation methods, and especially positional methods (e.g. combSUM [15]), are not initially designed to work with such rankings. They should be adapted by considering specific working assumptions.

We propose a new outranking method for rank aggregation which is well adapted to the IR context. Indeed, it ranks two documents w.r.t. the intensity of their positions difference in each input ranking and also considering the number of the input rankings that are concordant and discordant in favor of a specific document. There is also no need to make specific assumptions on the positions of the missing documents. This is an important feature since the absence of a document from a ranking should not be necessarily interpreted negatively.

Experimental results show that the outranking method *significantly* out-performs popular classical positional data fusion methods like combSUM and combMNZ strategies. It also out-performs a good performing majoritarian methods which is the Markov chain method. These results are tested against different test collections and queries. From the experiments, we can also conclude that in order to improve the performances, we should fuse result lists of well performing

IR models, and that majoritarian data fusion methods perform better than positional methods.

The proposed method can have a real impact on Web metasearch performances since only ranks are available from most primary search engines, whereas most of the current approaches need scores to merge result lists into one single list.

Further work involves investigating whether the outranking approach performs well in various other contexts, e.g. using the document scores or some combination of document ranks and scores.

Acknowledgments

The authors would like to thank Jacques Savoy for his valuable comments on a preliminary version of this paper.

7. REFERENCES

[1] A. Aronson, D. Demner-Fushman, S. Humphrey, J. Lin, H. Liu, P. Ruch, M. Ruiz, L. Smith, L. Tanabe, and W. Wilbur. Fusion of knowledge-intensive and statistical approaches for retrieving and annotating textual genomics documents. In *Proceedings TREC'2005*. NIST Publication, 2005.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press , 1999.

[3] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic combination of multiple ranked retrieval systems. In *Proceedings ACM-SIGIR'94*, pages 173–181. Springer-Verlag, 1994.

[4] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining evidence of multiple query representations for information retrieval. *IPM*, 31(3):431–448, 1995.

[5] J. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie des Sciences*, 1781.

[6] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings ACM-SIGIR'95*, pages 21–28, 1995.

[7] M. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

[8] W. D. Cook and M. Kress. Ordinal ranking with intensity of preference. *Management Science*, 31(1):26–32, 1985.

[9] N. Craswell and D. Hawking. Overview of the TREC-2002 Web Track. In *Proceedings TREC'2002*. NIST Publication, 2002.

[10] N. Craswell and D. Hawking. Overview of the TREC-2004 Web Track. In *Proceedings of TREC'2004*. NIST Publication, 2004.

[11] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings WWW'2001*, pages 613–622, 2001.

[12] R. Fagin. Combining fuzzy information from multiple systems. *JCSS*, 58(1):83–99, 1999.

[13] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *PODS*, pages 47–58, 2004.

[14] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. on Discrete Mathematics*, 17(1):134–160, 2003.

[15] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proceedings of TREC'3*. NIST Publication, 1994.

[16] J. Katzer, M. McGill, J. Tessier, W. Frakes, and P. DasGupta. A study of the overlap among document representations. *Information Technology: Research and Development*, 1(4):261–274, 1982.

[17] L. S. Larkey, M. E. Connell, and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proceedings ACM-CIKM'2000*, pages 282–289. ACM Press, 2000.

[18] A. Le Calvé and J. Savoy. Database merging strategy based on logistic regression. *IPM*, 36(3):341–359, 2000.

[19] J. H. Lee. Analyses of multiple evidence combination. In *Proceedings ACM-SIGIR'97*, pages 267–276, 1997.

[20] D. Lillis, F. Toolan, R. Collier, and J. Dunnion. Probfuse: a probabilistic approach to data fusion. In *Proceedings ACM-SIGIR'2006*, pages 139–146. ACM Press, 2006.

[21] J. I. Marden. *Analyzing and Modeling Rank Data*. Number 64 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1995.

[22] M. Montague and J. A. Aslam. Metasearch consistency. In *Proceedings ACM-SIGIR'2001*, pages 386–387. ACM Press, 2001.

[23] D. M. Pennock and E. Horvitz. Analysis of the axiomatic foundations of collaborative filtering. In *Workshop on AI for Electronic Commerce at the 16th National Conference on Artificial Intelligence*, 1999.

[24] M. E. Renda and U. Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *Proceedings ACM-SAC'2003*, pages 841–846. ACM Press, 2003.

[25] W. H. Riker. *Liberalism against populism*. Waveland Press, 1982.

[26] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31:49–73, 1991.

[27] B. Roy and J. Hugonnard. Ranking of suburban line extension projects on the Paris metro system by a multicriteria method. *Transportation Research*, 16A(4):301–312, 1982.

[28] L. Si and J. Callan. Using sampled data and regression to merge search engine results. In *Proceedings ACM-SIGIR'2002*, pages 19–26. ACM Press, 2002.

[29] M. Truchon. An extension of the Condorcet criterion and Kemeny orders. Cahier 9813, Centre de Recherche en Economie et Finance Appliquées, Oct. 1998.

[30] H. Turtle and W. B. Croft. Inference networks for document retrieval. In *Proceedings of ACM-SIGIR'90*, pages 1–24. ACM Press, 1990.

[31] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.