Competence-Conscious Associative Rank Aggregation

Adriano Veloso, Marcos André Gonçalves and Wagner Meira Jr.

Computer Science Department – Universidade Federal de Minas Gerais
adrianov@dcc.ufmg.br
mgoncalv@dcc.ufmg.br
meira@dcc.ufmg.br

Abstract. The ultimate goal of ranking methods is to achieve the best possible ranking performance for the problem at hand. Recently, a body of empirical evidence has emerged suggesting that methods that learn to rank offer substantial improvements in enough situations to be regarded as a relevant advance for applications that depend on ranking. Previous studies have shown that different (learning to rank) methods may produce conflicting ranked lists. Rank aggregation is based on the idea that combining such lists may provide complementary information that can be used to improve ranking performance. In this article we investigate learning to rank methods that uncover, from the training data, associations between document features and relevance levels in order to estimate the relevance of documents with regard to a given query. There is a variety of statistic measures or metrics that provide a different interpretation for an association. Interestingly, we observed that each association metric has a specific domain for which it is most competent (that is, there is a specific set of documents for which a specific metric consistently produces better ranked lists). We employ a second-stage meta-learning approach, which describes the domain of competence of each metric, enabling a more sensible aggregation of the ranked lists produced by different metrics. We call this new aggregation paradigm *competence-conscious associative rank aggregation*. We conducted a systematic evaluation of competence-conscious aggregation methods using the LETOR 3.0 benchmark collections. We demonstrate that the proposed aggregation methods outperform the constituent learning to rank methods not only when they are considered in isolation, but also when they are combined using existing aggregation approaches.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: Learning to Rank

Keywords: Competence, Ranking, Meta-Learning

1. INTRODUCTION

Ranking has emerged as a new class of statistical learning problem, which is distinct from classic ones (i.e., classification and regression). Ranking problems arise in a wide variety of domains. In Information Retrieval, ranking becomes particularly important because accurate ordering documents that are retrieved by search engines is paramount to effective search. Many features may affect the ordering of these documents, and thus, it is difficult to adapt ranking functions manually. Hence, there is a growing interest in training search engines to sort documents automatically using machine learning algorithms [Almeida et al. 2007; Cao et al. 2007; Yue et al. 2007]. The conventional approach to this learning task assumes the availability of examples (i.e., a training data which typically consists of document features and the corresponding relevance to specific queries), from which a ranking function can be learned. When a new query is given, the documents associated with this query are ranked according to the learned function, which gives a score to each document indicating its relevance to the query.

There are countless strategies for devising learning to rank methods. Such methods usually rely on techniques such as neural networks [Burges et al. 2005], genetic programming [Almeida et al.

This research was partially supported by CNPq, CAPES, FINEP, FAPEMIG, and INWeb.

2007] and support vector machines [Yue et al. 2007]. A more recent strategy is to directly explore associations between document features and relevance levels. Such associations are usually hidden in the training examples, and when uncovered, they may reveal important aspects concerning the underlying phenomenon that generated these examples. These aspects can be exploited for the sake of learning ranking functions. This strategy has led to a new family of learning to rank methods which are hereafter referred to as *associative methods* [Veloso et al. 2008]. The ranking function produced by an associative method is essentially a set of rules $\mathcal{X} \xrightarrow{\theta} r$, where each rule indicates an association between a set of features $\mathcal{X}$ and a relevance level r. A statistic metric θ gives the strength of this association, and is used to estimate the relevance of documents. Differently from [Veloso et al. 2008], where the only association metric used was the well-known confidence [Agrawal et al. 1993], in this article we propose to combine multiple association metrics in order to provide improved ranking performance.

Conflicting ranked lists can be produced from the same set of rules, depending on the way the association represented by these rules is interpreted (i.e., ranked lists produced by different association metrics may have divergent orderings). By investigating such conflicts we observed that, in fact, the relevance of specific documents is better estimated using specific association metrics (i.e., there is an optimal match between documents and the association metrics that are used to estimate their relevances). Thus, conflicting ranked lists produced by different association metrics may be complementary, in the sense that a better ranking performance may be obtained by aggregating these lists. We found that it is possible to approximate the optimal match between documents and metrics by exploring the domain of competence of metrics (i.e., sets of documents for which a specific metric consistently produces more accurate lists than the ones produced using other metrics). This notion of competence, which is a variation of the one introduced in [Veloso et al. 2009], enables us to devise a powerful aggregation approach, which we call *competence-conscious associative aggregation*. In this new paradigm a second-stage meta-learning approach describes the domain of competence of each metric, so that a metric is only used to estimate the relevance of documents in its domain of competence.

We conducted experiments comparing competence-conscious aggregation against existing learning to rank methods [Freund et al. 2003; Liu 2009; Joachims 2002; Xu and Li 2007; Tsai et al. 2007; Cao et al. 2007; Qin et al. 2007; Wang et al. 2010] as well as against other aggregation methods (CombMNZ [Shaw and Fox 1994], Borda Count [Aslam and Montague 2001], Condorcet [Montague and Aslam 2002], and linear combination [Vogt and Cottrell 1999]). The results, obtained using the LETOR benchmark, show that the proposed aggregation methods provide gains ranging from roughly 1% to 7% in terms of MAP when compared against other representative aggregation methods. The specific contributions of this article are summarized as follows:

—We show that the ranking performance of associative methods is strongly dependent on the association metric used to estimate the relevance of documents. We also show that no metric is consistently superior than all others, in the sense that it can be safely used in isolation. In fact, each metric has a particular domain of competence for which it is able to produce the most accurate ranked list. We introduce the notion of ranking competence and present a comprehensive study of the competence of different association metrics.

—We investigate meta-learning methods, which use the training data to learn the domains of competence of metrics. These domains are then exploited to decide which is the best metric to be applied in order to rank specific documents, resulting in an aggregation of the ranked lists produced by different metrics. This enables a more sensible combination of metrics, improving the quality of the final list.

—We propose two aggregation methods, QC^{3}A and DC^{3}A, based on competence-conscious aggregation. Their difference resides in the way they perform the analysis of the domains of competence. While QC^{3}A performs a query-centric analysis, DC^{3}A performs a finer-grained, document-centric

analysis. We show that the proposed methods provide significant gains in ranking performance.

2. RELATED WORK

Many prior efforts have been devoted to exploit rank aggregation in order to improve ranking performance. Particularly noteworthy contributions include [Shaw and Fox 1994; Vogt and Cottrell 1999; Bartell et al. 1994; Lee 1995; Montague and Aslam 2002; Aslam and Montague 2001; Dwork et al. 2001; Liu et al. 2007]. Fox and Shaw designed a series of aggregation methods including CombSum and Comb-MNZ [Shaw and Fox 1994]. CombSum sets the score of each document in the resulting list to the sum of the scores obtained by each constituent ranked list, while in CombMNZ the score of each document is obtained by multiplying this sum by the number of lists which have non-zero scores for this document. Lee [Lee 1995] performed several experiments with CombSum and CombMNZ methods, showing that for effective aggregation, the constituent lists must present a large overlap of relevant documents. Bartell [Bartell et al. 1994] and Vogt [Vogt and Cottrell 1999] proposed aggregation methods based on optimizing the parameters of the linear combination of the scores given to each document.

Adapted election strategies are widely used to solve aggregation problems. Aslam et al. [Aslam and Montague 2001] developed an aggregation method based on the Borda Count strategy. The Borda Count is the central representative of the class of positional (consensus-based) election procedures. This method determines the most relevant document by giving each document a certain number of points corresponding to the position in which it appears in each of the constituent ranked lists. Once all points have been counted the document with more points is the most relevant one. The Condorcet method, which is the central representative of the class of majoritarian election procedures, was exploited in [Montague and Aslam 2002]. It works similarly to Borda Count, but it specifies that the most relevant document is the one that, when compared with every other document, has more points. Both election-based methods are successively applied in order to rank all documents.

Dwork et al. [Dwork et al. 2001] showed the benefits of aggregation for Web search applications. They demonstrated that a proposal of Kemeny [Kemeny 1959] leads to an effective aggregation method. The basic idea is to minimize the *Kendall* τ distance between the constituent ranked lists (i.e., it minimizes the number of pairs in which two ranked lists disagree). Liu et al. [Liu et al. 2007] proposed a supervised rank aggregation method, which minimizes the disagreement between the constituent ranked lists and the labeled data.

The work to be presented in this article differs from previous work in the sense that rank aggregation is performed by meta learning. More specifically, the constituent ranked lists to be aggregated are produced by different associative learning to rank methods, and the proposed approach learns how to aggregate these lists. We introduce ranking competence, which is the main artifact exploited during meta learning. The definition of ranking competence is also different from the definition of classification competence, which was introduced in [Veloso et al. 2009]. Specifically, a classifier is competent with regard to a document d, if it correctly predicts the class for d. Ranking competence, on the other hand, measures the discrepancy (or distance) between the estimated relevance of d and the true relevance of d. The meta learning approach used in this article is similar to the meta learning approach used in [Veloso et al. 2009], although the objectives are significantly different − in [Veloso et al. 2009] meta learning was exploited for the sake of classifier delegation, while in this article we exploit meta learning for the sake of rank aggregation.

The rank aggregation methods proposed in this article use associative methods as base components. These associative methods were previously proposed by us in [Veloso et al. 2008]. However, in this article, we propose effective aggregation procedures, and we demonstrate that these aggregation methods are able to provide significant improvements.

Table I. Training data: queries/documents.

	Query	Retrieved Documents				r^d
		id	PageRank	BM25	*tf*	
$\mathcal{D}$	q_1	d_1	[0.85-0.92]	[0.36-0.55]	[0.23-0.27]	1
		d_2	[0.74-0.84]	[0.36-0.55]	[0.46-0.61]	1
		d_3	[0.51-0.64]	[0.56-0.70]	[0.23-0.27]	0
	q_2	d_4	[0.74-0.84]	[0.36-0.55]	[0.28-0.45]	0
		d_5	[0.65-0.73]	[0.56-0.70]	[0.46-0.61]	1
		d_6	[0.93-1.00]	[0.36-0.55]	[0.62-0.76]	0
	q_3	d_7	[0.74-0.84]	[0.22-0.35]	[0.12-0.22]	0
		d_8	[0.65-0.73]	[0.56-0.70]	[0.46-0.61]	0
		d_9	[0.85-0.92]	[0.71-0.80]	[0.46-0.61]	1

3. LEARNING TO RANK USING ASSOCIATION RULES

The task of learning to rank is defined as follows. We have as input the *training data* (referred to as $\mathcal{D}$), which consists of a set of records of the form $<q, d, r^d>$, where q is a query, d is a document (represented as a list of attribute-values or features $\{f_1, \ldots, f_m\}$), and r^d is the *relevance* of d to q. The relevance draws its values from a discrete set of possibilities (e.g., $r_0, \ldots, r_k$). The training data is used to construct a model which relates features of the documents to their corresponding relevance. The *test set* (referred to as $\mathcal{T}$) consists of records $<q, d, ?>$ for which only the query q and the document d are known, while the relevance of d to q is unknown. The model learned from $\mathcal{D}$ is used to estimate the relevance of such documents to the corresponding queries.

Consider Table I, which shows an illustrative example of training data. There are three queries in the training data. For each query three documents are retrieved, and each document is represented by three attributes − PageRank, BM25 and tf (the corresponding features were obtained by discretizing these attributes). In previous work [Veloso et al. 2008] we have presented a new learning to rank method which is based on the utilization of association rules [Agrawal et al. 1993]. The proposed method produces a model, $\mathcal{R}$, composed of rules of the form $f_j \wedge \ldots \wedge f_l \xrightarrow{\theta} r_i$, which describes $\mathcal{D}$ by means of feature-relevance associations. These rules can contain any mixture of the available features in the antecedent and a relevance level in the consequent. The strength of the association between antecedent and consequent is given by a metric, θ.

3.1 Rule Extraction

The search space for rules is huge, and thus, computational cost restrictions must be imposed during rule extraction. Typically, a minimum support threshold (σ_{min}) is employed in order to select frequent rules to compose the ranking model. This strategy, although simple, has some problems. If σ_{min} is set too low, a large number of rules will be extracted from $\mathcal{D}$, and often most of these rules are useless for ranking documents in the test set (a rule $\mathcal{X} \rightarrow r_i$ is only useful to rank document d if the set of features $\mathcal{X} \subseteq d$, otherwise the rule is meaningless to d). On the other hand, if σ_{min} is set too high, some important rules will not be included in $\mathcal{R}$, causing problems if some documents in the test set contain rare features. Usually, there is no optimal value for σ_{min}, that is, there is no single value that ensures that only rules useful for ranking documents in $\mathcal{T}$ are included in $\mathcal{R}$, while at the same time important rules are not missed. The ranking method proposed here deals with this problem by extracting rules on a demand-driven basis [Veloso et al. 2006].

On-demand rule extraction is delayed until a set of documents is considered for a given query. Then, each individual document d in $\mathcal{T}$ is used as a filter to remove irrelevant features and examples from $\mathcal{D}$. This process produces a projected training data, $\mathcal{D}_d$, which is focused only on the useful examples for ranking a specific document, d. Therefore, there is an automatic reduction of the size

and dimensionality of the training data, since examples and features that are meaningless to d are not considered during rule extraction[1]. As a result, for a given value of σ_{min}, important rules that are not frequent in $\mathcal{D}$, may become frequent in $\mathcal{D}_d$, providing a better coverage of the examples.

3.2 Relevance Estimation

In order to estimate the relevance of a document d, it is necessary to combine all rules in $\mathcal{R}_d$. Our strategy is to interpret $\mathcal{R}_d$ as a poll, in which each rule $\mathcal{X} \xrightarrow{\theta} r_i \in \mathcal{R}_d$ is a vote given by a set of features $\mathcal{X}$ for relevance level r_i. Votes have different weights, depending on the strength of the association of the corresponding rules (i.e., θ). The weighted votes for relevance level r_i are summed and then averaged, forming the score associated with relevance r_i for document d, as shown in Equation 1 (where $\theta(t)$ is the value of metric θ assumes for rule t):

$$s(r_i, d) = \frac{\sum_{t \in \mathcal{R}_d} \theta(t)}{\mid \mathcal{R}_d \mid}. \tag{1}$$

Therefore, for a document d, the score associated with relevance r_i is given by the average strength of the rules in $\mathcal{R}_d$ predicting r_i. The likelihood of d having a relevance level r_i is obtained by normalizing the scores:

$$\hat{p}(r_i|d) = \frac{s(r_i, d)}{\sum_{j=0}^{k} s(r_j, d)} \tag{2}$$

Finally, the rank of document d is estimated by a linear combination of the likelihoods associated with each relevance level, as shown in Equation 3:

$$rank(\theta, d) = \sum_{i=0}^{k} r_i \times \hat{p}(r_i|d). \tag{3}$$

The value of $rank$ is an estimation of the true relevance of d (i.e., r^d) using rules in $\mathcal{R}_d$. The rank values are used to produce ranked lists of documents. Thus, both $rank(\theta, d)$ and r^d assume values in the same range. This method was extensively evaluated in [Veloso et al. 2008], and in this article we went further, by exploiting the fact that ranking performance is affected by the association metric that is used in Equation 1.

3.3 Association Metrics

We employed seven different metrics for measuring the strength of association between document features ($\mathcal{X}$) and relevance levels ($r_1, r_2, \ldots, r_k$). These metrics were shown to produce ranked lists that present conflicts [Veloso et al. 2009]:

—Added Value (θ_1) [Hilderman and Hamilton 2001]: This metric measures the gain in accuracy obtained by using rule $\mathcal{X} \rightarrow r_i$ instead of always using r_i, as shown in Eq. 4. Negative values indicate that always predicting r_i is better than using the rule. Its value ranges from -1 to 1.

[1]An example $e \in \mathcal{D}$ is useless for ranking d if $e \cap d = \emptyset$. That is, e does not share any feature with d.

$$\theta_1 = p(r_i|\mathcal{X}) - p(r_i). \tag{4}$$

—Certainty (θ_2) [Lavrac et al. 1999]: This metric measures the increase in accuracy between rule $\mathcal{X}\rightarrow r_i$ and always using r_i, as shown in Eq. 5. It assumes values smaller than 1.

$$\theta_2 = \frac{p(r_i|\mathcal{X}) - p(r_i)}{p(\overline{r_i})} \tag{5}$$

—Confidence (θ_3) [Agrawal et al. 1993]: This metric measures the fraction of examples in $\mathcal{D}_t$ containing $\mathcal{X}$ that belong to r_i. It is the conditional probability of r_i being the correct relevance of document t given that $\mathcal{X} \subseteq t$, as shown in Eq. 6. Its value ranges from 0 to 1.

$$\theta_3 = p(r_i|\mathcal{X}) \tag{6}$$

—Strength Score (θ_4) [Arunasalam and Chawla 2006]: This metric measures the correlation between $\mathcal{X}$ and r_i, but also takes into account how $\mathcal{X}$ is correlated with the complement of r_i, as shown in Equation 7. Its value ranges from 0 to ∞.

$$\theta_4 = \frac{p(\mathcal{X}|r_i)p(r_i|\mathcal{X})}{p(\mathcal{X}|\overline{r_i})} \tag{7}$$

—Yules'Q (θ_5) and Yules'Y (θ_6) [Tan et al. 2002]: These metrics are based on odds value, as shown in Eqs. 8 and 9, respectively. Their values range from -1 to 1. The value 1 implies perfect positive association between $\mathcal{X}$ and r_i, and value -1 implies perfect negative association.

$$\theta_5 = \frac{p(\mathcal{X} \cup r_i)p(\overline{\mathcal{X} \cup r_i}) - p(\mathcal{X} \cup \overline{r_i})p(\overline{\mathcal{X}} \cup r_i)}{p(\mathcal{X} \cup r_i)p(\overline{\mathcal{X} \cup r_i}) + p(\mathcal{X} \cup \overline{r_i})p(\overline{\mathcal{X}} \cup r_i)} \tag{8}$$

$$\theta_6 = \frac{\sqrt{p(\mathcal{X} \cup r_i)p(\overline{\mathcal{X} \cup r_i})} - \sqrt{p(\mathcal{X} \cup \overline{r_i})p(\overline{\mathcal{X}} \cup r_i)}}{\sqrt{p(\mathcal{X} \cup r_i)p(\overline{\mathcal{X} \cup r_i})} + \sqrt{p(\mathcal{X} \cup \overline{r_i})p(\overline{\mathcal{X}} \cup r_i)}} \tag{9}$$

—Relative Confidence (θ_7) [Lavrac et al. 1999]: This metric trades off accuracy and generality, as shown in Equation 10. The first component is the accuracy gain that is obtained by using rule $\mathcal{X}\rightarrow r_i$ instead of always predicting r_i. The second component incorporates generality.

$$\theta_7 = (p(r_i|\mathcal{X}) - p(r_i))p(\mathcal{X}) \tag{10}$$

Example. The entire process of learning to rank using association rules is illustrated using the example shown in Table II. Suppose we want to calculate the rank value for document d_{10}. The original training data shown in Table I is projected according to d_{10}, resulting in $\mathcal{D}_{d_{10}}$={d_1, d_2, d_3, d_4, d_6}. Five rules are extracted from $\mathcal{D}_{d_{10}}$:

(1) PageRank=[0.51-0.64]→ r=0
($\theta_1 = 0.72$, $\theta_2 = 1.00$, $\theta_3 = 1.00$, $\theta_4 = 0.99$, $\theta_5 = 0.99$, $\theta_6 = 888{,}888$, $\theta_7 = 0.44$)
(2) BM25=[0.36-0.55]→ r=1
($\theta_1 = 0.72$, $\theta_2 = 1.00$, $\theta_3 = 1.00$, $\theta_4 = 0.99$, $\theta_5 = 0.99$, $\theta_6 = 888{,}888$, $\theta_7 = 0.44$)
(3) BM25=[0.36-0.55]→ r=0
($\theta_1 = 0.72$, $\theta_2 = 1.00$, $\theta_3 = 1.00$, $\theta_4 = 0.99$, $\theta_5 = 0.99$, $\theta_6 = 888{,}888$, $\theta_7 = 0.44$)
(4) BM25=[0.36-0.55]$\wedge tf$=[0.28-0.45]→ r=0
($\theta_1 = 0.47$, θ_2=0.01, θ_3=0.50, θ_4=0.40, θ_5=0.45, θ_6=0.40, θ_7=0.05)

Table II. Test Set: Ranked Lists produced by different Metrics and Ideal Rank Aggregation.

	Retrieved Documents				r^d	Ranked Lists							Agg.
	id	PageRank	BM25	*tf*		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	
$\mathcal{T}$	d_{10}	[0.51-0.64]	[0.36-0.55]	[0.28-0.45]	0	0.33	0.08	0.26	0.30	0.28	**0.00**	0.10	0.00
	d_{11}	[0.85-0.92]	[0.56-0.70]	[0.46-0.61]	1	0.60	0.81	0.62	0.68	0.66	**0.99**	0.83	0.99
	d_{12}	[0.34-0.50]	[0.22-0.35]	[0.46-0.61]	1	0.48	0.72	0.47	0.56	0.48	0.00	**0.77**	0.77
	d_{13}	[0.74-0.84]	[0.71-0.80]	[0.62-0.76]	0	0.49	**0.39**	0.44	0.44	0.47	0.61	0.44	0.39

(5) tf=[0.28-0.45]$\rightarrow$ r=0
($\theta_1 = 0.53$, $\theta_2 = 0.10$, $\theta_3 = 0.50$, $\theta_4 = 0.60$, $\theta_5 = 0.55$, $\theta_6 = 625{,}000$, $\theta_7 = 0.05$)

These rules are combined in order to estimate $r^{d_{10}}$. Different ranked lists are obtained from different metrics, as shown in Table II. Conflicts or disagreements in the ranked lists produced by these metrics may provide complementary information. Ideally, a rank aggregation method should be able to select the best rank value for each document from each ranked list. In this case, as shown in the last column of Table II, the selected rank value should be the one which is closest to the true relevance of each document (which are are the values highlighted in bold). Next we will discuss aggregation methods that are based on this idea.

4. RANK AGGREGATION

Selecting an appropriate association metric is a major issue while devising an associative ranking method. As will be shown in our experiments, depending on characteristics of the documents, some metrics may be more suitable than others. This suggests that different metrics are only able to accurately rank a subset of the entire set of documents, which is the domain of competence of such metric. In this section we present aggregation methods that learn the domain of competence of these metrics, and use this information to aggregate ranked lists produced by different metrics.

4.1 Learning Domains of Competence

The optimal match between association metrics and documents is valuable information. In this section we present an approach to estimate such matching. We start by defining the *ranking competence* of a metric. Then, we present approaches that learn to separate documents that are accurately ranked by some metric, from documents that are not.

The ranking competence of metric θ_i, with regard to document d, which is denoted as $\phi(\theta_i, d)$, is defined in Equation 11. It is essentially given by the discrepancy between the estimated relevance of d (i.e., $rank(\theta_i, d)$) and the true relevance of d (i.e., r^d). Metric θ_i is more competent to rank d than metric θ_j if $\phi(\theta_i, d) < \phi(\theta_j, d)$. For instance, the most competent metrics are highlighted in bold in the example shown in Table II (i.e., those metrics with lowest $\phi(\theta_i, d)$ values).

$$\phi(\theta_i, d) = \mid rank(\theta_i, d) - r^d \mid \tag{11}$$

The ranking competence of a metric is novel information that may be used to enhance the original training data, $\mathcal{D}$. Specifically, for each document in $\mathcal{D}$, it is informed which metric is the most competent one by following a procedure which is similar to cross-validation (one query in $\mathcal{D}$ is used for testing, and the remaining queries in $\mathcal{D}$ are used for training). This process, which is illustrated in Algorithm 1, creates an enhanced training data, denoted as $\mathcal{D}^*_{dc}$, so that for each document $d \in \mathcal{D}^*_{dc}$ we have the most competent metric to rank d.

Algorithm 1 Representing the Competence of Metrics.

Require: The original training data $\mathcal{D}$
Ensure: The enhanced training data $\mathcal{D}^*_{dc}$

split $\mathcal{D}$ into k partitions, $\{p_1 \cup \ldots \cup p_k\}$ (each partition p_i is composed of the documents retrieved for query q_i)
$\mathcal{D}^*_{dc} \Leftarrow \emptyset$
for each partition p_i **do**
  for each document $d \in p_i$ **do**
    $\mathcal{R}_d \Leftarrow$ rules $\mathcal{X} \rightarrow r_j$ extracted from $\{\mathcal{D} - p_i\}$ such that $\mathcal{X} \subseteq d$
    for each metric θ_l **do**
      produce $rank(\theta_l, d)$ using $\mathcal{R}_d$
    end for
    $d^* \Leftarrow d + \theta_i$, where $\phi(\theta_i, d) \leq \phi(\theta_j, d) \forall \phi(\theta_j, d)$ (i.e., the most competent metric is appended to document d, creating d^*)
    $\mathcal{D}^*_{dc} \Leftarrow \mathcal{D}^*_{dc} \cup d^*$
  end for
end for

Algorithm 2 Selecting a Competent Metric.

Require: $\mathcal{D}^*_{dc}$ (or $\mathcal{D}^*_{qc}$), and a document $d \in \mathcal{T}$
Ensure: The most competent metric for document d

$\mathcal{R}_d \Leftarrow$ rules $\mathcal{X} \rightarrow \theta_i$ extracted from $\mathcal{D}^*_{dc}$ (or $\mathcal{D}^*_{qc}$) such that $\mathcal{X} \subseteq d$
Estimate $\hat{p}(\theta_i|d)$, according to Equation 2
return metric θ_j such that $\hat{p}(\theta_j|d) > \hat{p}(\theta_i|d) \forall i \neq j$

Another enhanced training data, $\mathcal{D}^*_{qc}$ can be obtained following a very similar approach. The difference is that, instead of associating to each document $d \in \mathcal{D}$ the most competent metric for d, the metric to be associated to d is the one which was the most competent for the majority of documents retrieved for the corresponding query.

4.2 Competence-Conscious Aggregation

Any method discussed in Section 2 can be used to aggregate ranked lists produced by different metrics. However, such methods neglect information about the competence of these metrics. In this section we discuss how to exploit $\mathcal{D}^*_{dc}$ and $\mathcal{D}^*_{qc}$ in order to produce competence-conscious aggregation methods. Next we present methods that properly select a competent metric θ_i to rank a specific document d.

Document-Centric(DC^3A) and Query-Centric(QC^3A) Competence-Conscious Aggregation. The aggregation methods to be presented are based on selecting, for each document $d \in \mathcal{T}$, a metric θ_i, which is likely to be competent for d (i.e., $\phi(\theta_i, d)$ is expected to be low). Both $\mathcal{D}^*_{dc}$ and $\mathcal{D}^*_{qc}$ can be used to estimate the ranking competence of each metric. Instead of extracting rules of the form $\mathcal{X} \rightarrow r_i$, DC^3A and QC^3A extract rules of the form $\mathcal{X} \rightarrow \theta_i$. Specifically, the only difference between DC^3A and QC^3A is that, while DC^3A extracts rules from $\mathcal{D}^*_{dc}$ (where the domains of competence are fine-grained), QC^3A extracts rules from $\mathcal{D}^*_{qc}$ (where the domains of competence are coarse-grained). The extracted rules are combined and the association metric with the highest likelihood of being the most competent one for d, is finally selected. This is essentially a meta-learning process, since the competence of metrics is learned from inputs, that are, in turn, the outputs of associative learning methods (i.e., $\mathcal{D}^*_{dc}$ and $\mathcal{D}^*_{qc}$). This process is illustrated in Algorithm 3.

Once metric θ_i is selected for document d, a rank value $rank(\theta_i, d)$ is finally produced. It is expected

Algorithm 3 Competence-Conscious Rank Aggregation.

Require: $\mathcal{D}^*_{dc}$ (or $\mathcal{D}^*_{qc}$), and a document $d \in \mathcal{T}$
Ensure: The rank value for document d

1: select θ_i (the most competent metric for d), using Algorithm 3
2: $\mathcal{R}_d \Leftarrow$ rules $\mathcal{X} \xrightarrow{\theta_i} r_j$ extracted from $\mathcal{D}^*_{dc}$ (or $\mathcal{D}^*_{qc}$) such that $\mathcal{X} \subseteq d$
3: **return** $rank(\theta_i, d)$, using $\mathcal{R}_d$

Table III. Enhanced training data, $\mathcal{D}^*_{dc}$ and $\mathcal{D}^*_{qc}$.

Query	Retrieved Documents				r^d	Metric $\mathcal{D}^*_{dc}$	Metric $\mathcal{D}^*_{qc}$
	id	PageRank	BM25	*tf*			
q_1	d_1	[0.85-0.92]	[0.36-0.55]	[0.23-0.27]	1	θ_3	θ_3
	d_2	[0.74-0.84]	[0.36-0.55]	[0.46-0.61]	1	θ_1	θ_3
	d_3	[0.51-0.64]	[0.56-0.70]	[0.23-0.27]	0	θ_3	θ_3
q_2	d_4	[0.74-0.84]	[0.36-0.55]	[0.28-0.45]	0	θ_5	θ_5
	d_5	[0.65-0.73]	[0.56-0.70]	[0.46-0.61]	1	θ_5	θ_5
	d_6	[0.93-1.00]	[0.36-0.55]	[0.62-0.76]	0	θ_3	θ_5
q_3	d_7	[0.74-0.84]	[0.22-0.35]	[0.12-0.22]	0	θ_1	θ_1
	d_8	[0.65-0.73]	[0.56-0.70]	[0.46-0.61]	0	θ_1	θ_1
	d_9	[0.85-0.92]	[0.71-0.80]	[0.46-0.61]	1	θ_3	θ_1

that, at the end of the process, the final ranked list will be composed of rank values produced by the most competent metric for each document. This process is shown in Algorithm 3.

Example. We illustrate how competence-conscious algorithms work by using Tables I, II, and III. Lets discuss DC^{3}A first. DC^{3}A takes as input the enhanced training data, $\mathcal{D}^*_{dc}$, which was produced according Algorithm 1, and is shown in Table III. Next, DC^{3}A takes as input $\mathcal{T}$, and uses Algorithm 3 in order to select the most competent metric for each document $d \in \mathcal{T}$. In the last step, DC^{3}A uses the selected metric, θ, in order to produce rules $\mathcal{X} \rightarrow r_i$. Then, DC^{3}A calculates $rank(\theta, d)$ according to Equation 3. QC^{3}A essentialy works in the same way as DC^{3}A. The only difference resides in the second step, since QC^{3}A uses $\mathcal{D}^*_{qc}$ instead of $\mathcal{D}^*_{dc}$.

5. EXPERIMENTAL EVALUATION

In this section we empirically analyze the proposed rank aggregation methods, DC^{3}A and QC^{3}A. We first present the collections employed in the evaluation, and then we discuss the effectiveness of the methods in these collections.

5.1 The LETOR Benchmark

LETOR 3.0 [Liu et al. 2007] makes available 7 subsets (OHSUMED, TD2003, TD2004, HP2003, HP2004, NP2003 and NP2004), each containing a set of queries, document features, and relevance judgments. Features cover properties [Baeza-Yates and R-Neto 2011] such as BM25, PageRank, HITS etc. In order to conduct 5-fold cross validation, each subset is arranged in 5 folds, including training, validation and test data. Performance is evaluated using NDCG, precision, and MAP measures [Baeza-Yates and R-Neto 2011]. Pre-processing involved only the discretization [Fayyad and Irani 1993].

The LETOR 3.0 benchmark also makes available a set of ranking methods, including Ranking SVM [Yue et al. 2007], RankBoost [Freund et al. 2003], FRank [Tsai et al. 2007], and ListNet [Cao et al. 2007]. Details about these methods can be found in the LETOR 3.0 website (`http://research.microsoft.com/users/LETOR`). Our evaluation is based on a comparison against these ranking methods, as well as against several aggregation methods, such as CombMNZ [Shaw and Fox

Table IV. MAP numbers for the OHSUMED subset. Best results, including ties, are shown in bold.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	**0.349**	0.344	**0.352**	0.316	**0.349**	**0.352**	0.345	0.304	0.332	0.333	0.346	0.342
2	0.450	0.447	**0.463**	0.395	0.445	0.451	**0.462**	0.447	0.445	0.438	0.450	0.454
3	**0.466**	0.449	0.460	0.438	**0.466**	**0.463**	0.444	**0.465**	0.456	0.456	0.461	**0.462**
4	**0.521**	0.512	**0.521**	0.498	0.515	**0.521**	**0.519**	0.499	0.508	0.513	0.511	**0.518**
5	0.479	0.476	**0.482**	**0.488**	0.481	0.480	0.466	0.453	0.464	0.481	0.461	0.450
Avg	**0.453**	0.446	**0.456**	0.427	0.451	**0.453**	0.447	0.433	0.441	0.444	0.446	0.445

Table V. MAP numbers for the TD2003 subset.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	0.132	0.115	0.169	0.132	0.156	0.152	0.144	0.164	0.110	0.113	**0.192**	0.172
2	0.284	0.281	0.293	0.289	0.291	0.291	0.270	0.258	0.291	0.297	**0.325**	0.237
3	0.362	0.356	0.365	0.322	0.361	0.364	0.350	**0.401**	0.251	0.155	0.381	0.342
4	**0.390**	0.378	**0.394**	0.373	**0.386**	0.381	0.371	0.237	0.262	0.212	0.275	0.276
5	0.211	0.209	0.220	0.202	0.218	0.217	0.198	**0.249**	0.222	0.238	0.202	0.196
Avg	0.277	0.268	**0.288**	0.264	0.281	0.281	0.267	0.263	0.227	0.203	0.275	0.244

Table VI. MAP numbers for the TD2004 subset.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	0.196	0.181	0.213	0.173	0.209	0.209	0.205	0.211	**0.247**	0.226	0.225	0.185
2	0.216	0.210	**0.280**	0.204	0.271	**0.276**	0.258	0.209	**0.281**	0.203	0.215	0.192
3	**0.281**	**0.284**	**0.285**	0.229	0.272	0.275	0.263	0.206	0.241	0.218	0.223	0.201
4	0.246	0.231	0.267	0.249	0.261	0.248	0.240	0.218	0.238	**0.285**	0.223	0.211
5	0.238	0.224	0.260	0.238	0.252	0.251	0.244	0.274	**0.299**	0.262	0.229	0.235
Avg	0.235	0.226	**0.261**	0.219	0.253	0.252	0.242	0.224	**0.261**	0.239	0.223	0.205

1994], Borda Count [Aslam and Montague 2001], Condorcet [Montague and Aslam 2002], and Linear Combination [Vogt and Cottrell 1999]. All aggregation methods combine ranked lists produced by associative methods (i.e., each method uses a different association metric).

5.2 Results

All experiments were performed on a Linux PC with an Intel Core 2 Duo 1.63GHz and 2GBytes RAM. For associative methods we followed the standard parameter setting procedure, in which a separate validation set is used to calculate MAP numbers. We maximize MAP on the validation set by selecting the appropriate parameter ($\sigma_{min} = 0.01$), hoping that this parameter value will also maximize MAP numbers in the test set.

How effective are associative methods when compared to other learning to rank methods?

Tables IV, V, VI, VII, VIII, IX, and X show MAP numbers for different subsets. The result for each trial is obtained by averaging partial results obtained from each query in the trial. The final result is obtained by averaging the five trials. We conducted two sets of significance tests (t-test) for each subset. The first set of significance tests was carried on the average of the results for each query. The second set of significance tests was carried on the average of the five trials.

As can be seen in Table IV, all methods showed competitive results in the OHSUMED subset. The worst overall result was obtained using θ_4 (0.426), while the best result was obtained using θ_3 (0.455). The main reason for so much competitiveness is that OHSUMED contains only few features, which are extracted basically from textual evidence, reducing the possibilities of obtaining large improvements.

Table VII. MAP numbers for the HP2003 subset.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	0.702	0.708	0.717	0.688	**0.729**	**0.729**	0.702	0.684	0.634	0.674	**0.728**	0.717
2	0.788	0.780	0.807	0.785	0.812	0.804	0.802	0.796	0.813	0.804	**0.852**	0.845
3	0.724	0.716	0.737	0.715	0.738	0.733	0.732	0.783	0.781	0.737	**0.821**	0.780
4	0.742	0.740	0.762	0.715	**0.780**	**0.780**	0.757	0.763	0.745	0.684	**0.772**	0.760
5	0.741	0.732	0.755	0.712	**0.771**	0.764	0.748	0.679	0.692	0.648	0.657	0.608
Avg	0.739	0.735	0.756	0.723	**0.766**	0.762	0.748	0.741	0.733	0.709	**0.766**	0.742

Table VIII. MAP numbers for the HP2004 subset.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	0.632	0.638	0.666	0.618	0.670	0.670	0.631	0.664	0.621	0.632	0.700	**0.729**
2	0.750	0.744	0.756	0.728	**0.774**	**0.774**	0.755	0.680	0.618	0.648	0.759	**0.775**
3	0.802	0.795	0.806	0.778	0.802	0.806	0.798	0.742	0.637	**0.842**	0.780	0.785
4	0.627	0.617	0.625	0.608	0.631	0.637	0.622	**0.715**	0.611	0.632	0.619	**0.719**
5	0.622	0.616	0.626	0.603	0.619	0.621	0.633	0.536	0.638	0.654	0.591	0.579
Avg	0.687	0.682	0.696	0.667	0.699	0.702	0.688	0.667	0.625	0.682	0.690	**0.718**

Table IX. MAP numbers for the NP2003 subset.

	Associative Methods							Ranking	Rank	FRank	ListNet	SVM
Trial	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	SVM	Boost			MAP
1	0.674	0.674	**0.695**	0.673	**0.699**	**0.699**	**0.697**	0.625	0.685	0.591	0.593	0.623
2	0.638	0.625	**0.676**	0.611	**0.672**	**0.677**	**0.671**	0.662	0.666	0.645	0.648	0.640
3	0.662	0.650	0.670	0.635	0.670	0.662	0.649	0.695	0.711	0.673	**0.751**	0.714
4	0.737	0.740	0.751	0.731	0.739	0.739	0.736	**0.761**	0.733	**0.769**	0.724	0.736
5	0.729	0.721	0.746	0.716	0.736	0.732	0.727	0.735	0.743	0.642	0.732	0.721
Avg	0.688	0.682	**0.707**	0.673	**0.703**	**0.702**	0.696	0.696	**0.707**	0.664	0.689	0.687

For TD2003, ListNet was the hardest baseline. As shown in Table V, gains obtained with θ_3 ranges from 15.02% (relative to ListNet) to 41.89% (relative to FRank). Similar results were obtained for HP2003 and NP2004, as shown in Tables VII and X. In contrast to OHSUMED, TD2003 and NP2004 contain more and diverse features, making possible the achievement of significant improvements.

For TD2004, the hardest baseline was RankBoost. As shown in Table VI, θ_1 showed gains in the first two trials, and also the best overall results, with gains ranging from 2.35% (relative to RankBoost) to 27.32% (relative to SVMMAP). Again, the worst results were obtained using θ_4. Similar results were obtained for HP2004, as shown in Table VIII.

How is competence distributed among metrics? Can we estimate the matching between documents and metrics?

Figure 1 (Left) shows the domains of competence of each metric (we grouped documents that belong to the same domain of competence in order to ease visualization). Lighter colored regions indicate documents in the y-axis that were competently ranked by the corresponding metric in the x-axis(i.e., ϕ(x,y) is low). Darker regions, on the other hand, indicate documents that were not competently ranked by the corresponding metric (i.e., ϕ(x,y) is high). Interestingly, for most of the documents, Strength Score (θ_4) is either the most (i.e., yellow regions) or the least competent metric (i.e., black regions). This is interesting because, differently from what would be expected due to its poor effectiveness shown in Tables IV, V and VI, for most of the documents Strength Score is, in fact, a very competent metric. This paradox happens because this metric tends to assign very low rank values to documents (as shown in Figure 2). While this is advantageous for less relevant documents, this

Table X. MAP numbers for the NP2004 subset.

Trial	Associative Methods							Ranking SVM	Rank Boost	FRank	ListNet	SVM MAP
	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7					
1	0.523	0.538	0.591	0.502	0.602	0.602	0.572	0.535	0.550	0.599	0.550	0.574
2	0.642	0.621	0.648	0.625	0.640	0.645	0.641	0.608	0.559	0.629	0.659	**0.669**
3	0.832	0.812	0.869	0.837	0.869	**0.880**	0.826	0.756	0.609	0.731	0.739	0.767
4	0.614	0.609	0.611	0.601	0.610	0.618	0.619	0.694	0.531	0.485	**0.728**	0.599
5	0.612	0.526	0.649	0.583	0.634	0.639	0.608	**0.701**	0.570	0.560	0.684	**0.701**
Avg	0.645	0.641	**0.674**	0.630	**0.671**	**0.677**	0.653	0.659	0.564	0.601	**0.672**	0.662

Table XI. MAP numbers for OHSUMED subset. Best results, including ties, are shown in bold.

Trial	DC^{3}A	QC^{3}A	CombMNZ	Borda Count	Condorcet	Linear Combination	Best Metric (θ_3)
1	0.350	**0.353**	0.341	0.346	0.347	**0.360**	**0.352**
2	**0.469**	0.460	0.458	0.462	0.458	0.456	**0.463**
3	**0.477**	0.461	0.455	0.455	0.459	0.462	0.460
4	**0.522**	0.513	0.492	0.483	0.494	0.513	**0.521**
5	**0.492**	0.478	0.433	0.419	0.426	0.477	0.482
Avg	**0.462**	0.454	0.436	0.433	0.437	0.454	0.456

Table XII. MAP numbers for TD2003 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_3)
1	**0.188**	**0.184**	0.181	**0.188**	0.169
2	**0.312**	0.301	0.298	**0.306**	0.293
3	**0.409**	**0.411**	0.371	0.398	0.365
4	**0.402**	**0.397**	0.378	0.384	**0.394**
5	**0.229**	**0.221**	0.217	**0.221**	0.220
Avg	**0.308**	**0.303**	0.289	0.299	0.288

causes problems for ranking the most relevant ones (explaining its poor effectiveness). In contrast to Strength Score, Added Value (θ_1), Certainty (θ_2), and Confidence (θ_3) are more competent for ranking more relevant documents. An aggregation method can only take advantage of these different properties if the matching between documents and metrics is well estimated. Figure 1 (Right) shows the effectiveness of DC^{3}A in selecting competent metrics according to each document. As can be seem, DC^{3}A usually selects clearer regions, while avoiding darker ones.

How effective is competence-conscious rank aggregation methods when compared to other methods?

Tables XI, XII, XIII, XIV, XV, XVI, and XVII, show MAP numbers for different subsets. The result for each trial is obtained by averaging results obtained from each query in the trial. The final result is obtained by averaging the five trials. In all subsets, DC^{3}A was the best overall performer, followed by QC^{3}A, which was also the best performer in the TD2003, TD2004, and NP2003 subsets. CombMNZ and Borda Count achieved the worst results in most of the subsets. This is mainly because these methods are not robust when some (possibly only one) constituent ranked lists are not accurate. Thus, the poor list produced by metric θ_4 severely impacted the effectiveness of these methods. The Condorcet method seems to be more robust, but still, DC^{3}A and QC^{3}A achieved much better results. Linear Combination was the best performer in some trials, and it was able to provide overall results that are very close to the results provided the best metric when applied in isolation. DC^{3}A was, most of the times, better than QC^{3}A, suggesting that the more fine-grained the analysis of competence, the more effectively lists are combined.

The last set of experiments evaluates the effectiveness of DC^{3}A and QC^{3}A, in terms of NDCG and precision. Fig. 3 shows results obtained from the execution of the evaluated aggregation methods. For the TD2003 subset, QC^{3}A is in close rivalry with DC^{3}A, specially in terms of NDCG. Higher gains

Table XIII. MAP numbers for TD2004 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_3)
1	**0.214**	**0.219**	0.194	0.203	**0.213**
2	**0.297**	**0.297**	0.277	0.281	0.280
3	**0.285**	271	0.272	**0.278**	**0.285**
4	**0.297**	0.282	0.276	0.281	0.267
5	**0.271**	**0.273**	0.259	0.259	0.260
Avg	**0.273**	**0.268**	0.256	0.260	0.261

Table XIV. MAP numbers for HP2003 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_5)
1	**0.743**	0.724	0.720	0.723	0.729
2	**0.833**	**0.827**	0.801	**0.824**	0.812
3	**0.758**	**0.754**	**0.742**	**0.758**	0.738
4	**0.782**	**0.780**	0.753	0.745	**0.780**
5	**0.784**	0.762	0.766	0.763	0.771
Avg	**0.780**	0.769	0.756	0.763	0.766

Table XV. MAP numbers for HP2004 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_6)
1	**0.743**	0.724	0.720	0.723	0.729
1	0.656	**0.672**	**0.667**	**0.672**	**0.670**
2	**0.794**	0.779	0.730	0.760	0.774
3	**0.819**	**0.810**	0.781	0.803	0.806
4	0.635	0.627	0.618	**0.647**	0.637
5	**0.649**	0.624	0.620	**0.643**	0.621
Avg	**0.711**	0.702	0.683	**0.705**	0.702

of DC^{3}A were achieved in the OHSUMED and TD2004 subsets. The worst baselines were Borda Count (in terms of NDCG), and CombMNZ (in terms of precision). The best baseline was Linear Combination. In terms of NDCG, the gains provided by DC^{3}A range from 9.30% (relative to Linear Combination) to 25.92% (relative to Borda Count). In terms of precision, gains range from 4.68% (relative to Linear Combination) to 13.11% (relative to CombMNZ).

What are the computational costs of QC^{3}A and DC^{3}A?

Table XVIII shows execution times for each method evaluated, namely, CombMNZ, Borda Count, Condorcet, Linear Combination, DC^{3}A, and QC^{3}A. Execution time reflects the entire process, which includes: the time spent for each constituent learning to rank associative method, in order to produce the base ranked lists (i.e., those based on θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, and θ_7), and the time spent to aggregate these lists. As can be seen, CombMNZ is the fastest method, since it simply sums and normalizes the score associated with each ranked list for each document. On the other hand, the Linear Combination method is the slowest one. Competence-conscious methods, DC^{3}A and QC^{3}A, are somewhat slower than Borda Count and Condorcet methods, but much faster than Linear Combination. The average time per query for QC^{3}A ranges from 0.37 (OHSUMED) to 2.41 (TD2004), and the average time per query for DC^{3}A ranges from 0.40 (OHSUMED) to 2.51 (TD2003).

6. CONCLUSIONS

This article focused on the important problem of ranking. We have shown that the performance of learning to rank methods that use association rules [Veloso et al. 2008] are strongly related to the metric that is used to estimate the association between document features and relevance levels. No

Table XVI. MAP numbers for NP2003 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_3)
1	**0.703**	**0.708**	0.690	0.693	0.695
2	**0.683**	**0.689**	0.677	**0.687**	0.676
3	0.659	**0.669**	**0.663**	0.651	**0.670**
4	0.765	0.752	0.761	**0.776**	0.751
5	**0.754**	**0.759**	0.745	0.748	0.746
Avg	**0.713**	**0.716**	0.707	**0.711**	0.707

Table XVII. MAP numbers for NP2004 subset.

Trial	DC^{3}A	QC^{3}A	Condorcet	Linear Combination	Best Metric (θ_6)
1	**0.600**	**0.596**	**0.600**	**0.603**	**0.602**
2	**0.642**	0.632	**0.644**	**0.640**	**0.645**
3	**0.879**	0.865	0.861	0.868	**0.880**
4	**0.635**	**0.628**	0.623	**0.630**	0.618
5	**0.652**	**0.645**	0.629	0.639	0.639
Avg	**0.682**	0.673	0.671	**0.676**	**0.677**

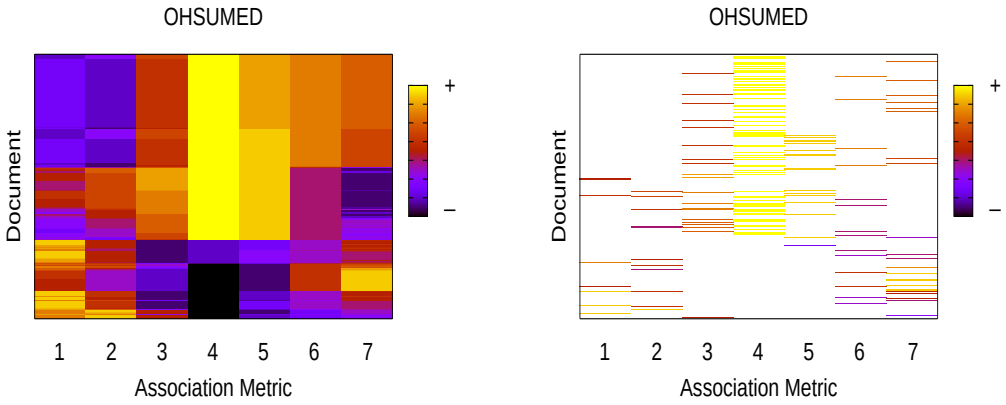

Fig. 1. Left – competence for each metric. Right – metrics selected by DC^{3}A.

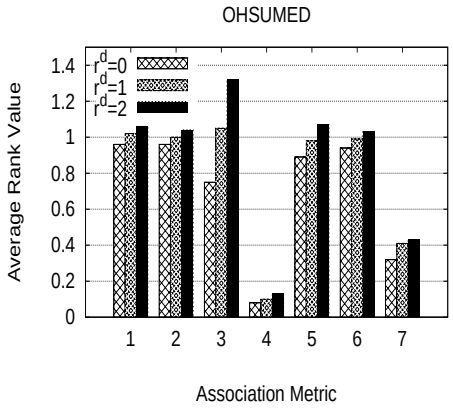

Fig. 2. Average rank values.

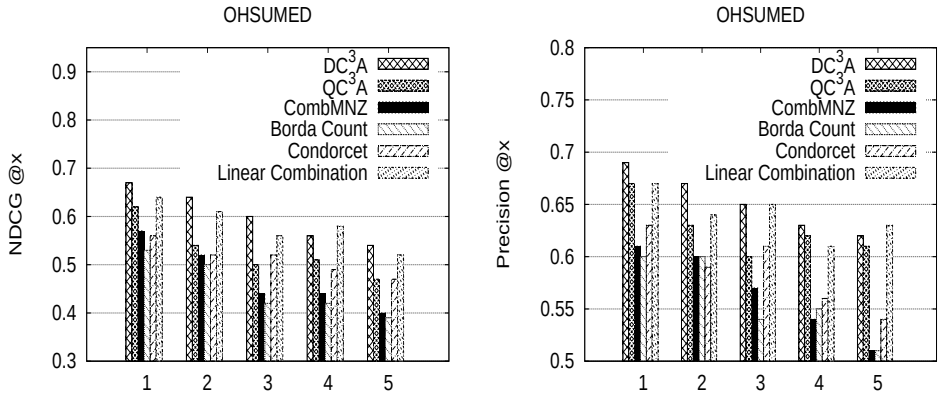

Fig. 3. NDCG and precision numbers.

Table XVIII. Execution times (in seconds) for different aggregation methods.

Method	OHSUMED	TD2003	TD2004	HP2003	HP2004	NP2003	NP2004
CombMNZ	25.656	71.485	101.206	162.994	67.727	145.395	72.285
Borda Count	32.187	92.179	138.278	204.931	91.416	193.292	82.521
Condorcet	34.117	100.766	150.164	217.400	100.136	211.845	88.071
QC^3A	40.139	115.633	181.395	266.613	118.191	244.928	103.478
DC^3A	43.284	125.239	188.723	274.282	127.788	259.180	111.827
Linear Comb.	272.223	739.481	1,289.910	2,179.376	949.668	2,647.137	784.929

metric is consistently superior than all others, in the sense that it can be safely used in isolation. In fact, each metric has a particular competence, for which it is able to produce accurate lists. We proposed to further improve the performance of these learning to rank methods, by aggregating their ranked lists. The proposed methods introduce effective innovations, such as the notion of ranking competence. Specifically, we investigate meta-learning approaches, which use the training data to learn the competence of each metric. Finally, the competence of metrics are exploited to decide which is the best metric to be applied to estimate the relevance of each document, resulting in a effective aggregation of ranked lists produced by different metrics. This aggregation paradigm, which we denote as *competence-conscious rank aggregation*, maximizes the accuracy of the final ranked list. We present the analysis of two aggregation methods that follow this new paradigm, DC^3A and QC^3A. The difference between them resides in how they perform the analysis of the domains of competence. The query-centric method (QC^3A) provides lower gains when compared to the finer-grained analysis, performed by the document-centric method (DC^3A), which outperforms all other competitors.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data Conference*. Washington, D.C., pp. 207–216, 1993.

Almeida, H., Gonçalves, M. A., Cristo, M., and Calado, P. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Amsterdan, pp. 399–406, 2007.

Arunasalam, B. and Chawla, S. CCCS: a top-down associative classifier for imbalanced class distribution. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, pp. 517–522, 2006.

Aslam, J. and Montague, M. Models for metasearch. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Tampere, pp. 276–284, 2001.

Baeza-Yates, R. and R-Neto, B. *Modern Information Retrieval*. Addison-Wesley-Longman, 2011.

Bartell, B., Cottrell, G., and Belew, R. Automatic combination of multiple ranked retrieval systems. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Dublin, pp. 173–181, 1994.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine Learning*. Bonn, pp. 89–96, 2005.

Cao, Z., Qin, T., Liu, T., Tsai, M., and Li, H. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the International Conference on Machine Learning*. Amsterdan, pp. 129–136, 2007.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation methods for the web. In *Proceedings of the International World Wide Web Conferences*. Hong Kong, pp. 613–622, 2001.

Fayyad, U. and Irani, K. Multi interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Chemberry, pp. 1022–1027, 1993.

Freund, Y., Iyer, R., Schapire, R., and Singer, Y. An efficient boosting algorithm for combining preferences. *J. of Machine Learning Research* 4 (1): 933–969, 2003.

Hilderman, R. and Hamilton, H. Evaluation of interestingness measures for ranking discovered knowledge. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Hong Kong, pp. 247–259, 2001.

Joachims, T. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Alberta, pp. 133–142, 2002.

Kemeny, J. Mathematics without numbers. *Daedalus* 88 (1): 571–591, 1959.

Lavrac, N., Flach, P., and Zupan, B. Rule evaluation measures: A unifying view. *Inductive Logic Prog.* 1634 (1): 174–185, 1999.

Lee, J. Analyses of multiple evidence combination. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Washington, D.C., pp. 267–276, 1995.

Liu, T. Learning to rank for information retrieval. *Foundations and Trends in Inf. Retrieval* 3 (3): 225–331, 2009.

Liu, Y., Liu, T., Qin, T., Ma, Z., and Li, H. Supervised rank aggregation. In *Proceedings of the International World Wide Web Conferences*. Banff, pp. 481–489, 2007.

Liu, Y., Xu, J., Qin, T., Xiong, W., and Li, H. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *L2R SIGIR Workshop*. Amsterdan, 2007.

Montague, M. and Aslam, J. Condorcet fusion for improved retrieval. In *Proceedings of the International Conference on Information and Knowledge Engineering*. Mclean, pp. 538–548, 2002.

Qin, T., Zhang, X., Wang, D., Liu, T., Lai, W., and Li, H. Ranking with multiple hyperplanes. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Amsterdan, pp. 279–286, 2007.

Shaw, J. and Fox, E. Combination of multiple searches. In *Proceedings of the Text Retrieval Conference*. Gaithersburg, pp. 243–252, 1994.

Tan, P., Kumar, V., and Srivastava, J. Selecting the right interestingness measure for association patterns. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Alberta, pp. 32–41, 2002.

Tsai, M., Liu, T., Qin, T., Chen, H., and Ma, W. FRank: a ranking method with fidelity loss. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Amsterdan, pp. 383–390, 2007.

Veloso, A., Almeida, H., Gonçalves, M. A., and Meira, W. Learning to rank at query-time using association rules. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Singapore, pp. 267–274, 2008.

Veloso, A., Jr., W. M., and Zaki, M. Lazy associative classification. In *Proceedings of the IEEE International Conference on Data Mining*. Hong Kong, pp. 645–654, 2006.

Veloso, A., Zaki, M., Jr., W. M., and Gonçalves, M. A. Competence-conscious associative classification. *Statistical Analysis and Data Mining* 2 (5-6): 361–377, 2009.

Vogt, C. and Cottrell, G. Fusion via a linear combination of scores. *Inf. Retr.* 1 (3): 151–173, 1999.

Wang, L., Lin, J., and Metzler, D. Learning to efficiently rank. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Geneva, pp. 138–145, 2010.

Xu, J. and Li, H. Adarank: a boosting algorithm for IR. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Amsterdan, pp. 391–398, 2007.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. A support vector method for optimizing average precision. In *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*. Amsterdan, pp. 271–278, 2007.