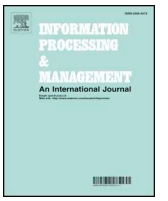
Information Processing and Management 57 (2020) 102149

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman



Effectiveness evaluation without human relevance judgments: A systematic analysis of existing methods and of their combinations

Kevin Roitero, Andrea Brunello, Giuseppe Serra, Stefano Mizzaro*

University of Udine, Udine, Italy

ARTICLE INFO

Keywords:
Information retrieval evaluation
Automatic evaluation
Machine learning
Topic difficulty

ABSTRACT

In test collection based evaluation of retrieval effectiveness, it has been suggested to completely avoid using human relevance judgments. Although several methods have been proposed, their accuracy is still limited. In this paper we present two overall contributions. First, we provide a systematic comparison of all the most widely adopted previous approaches on a large set of 14 TREC collections. We aim at analyzing the methods in a homogeneous and complete way, in terms of the accuracy measures used as well as in terms of the datasets selected, showing that considerably different results may be achieved considering different methods, datasets, and measures. Second, we study the combination of such methods, which, to the best of our knowledge, has not been investigated so far. Our experimental results show that simple combination strategies based on data fusion techniques are usually not effective and even harmful. However, some more sophisticated solutions, based on machine learning, are indeed effective and often outperform all individual methods. Moreover, they are more stable, as they show a smaller variation across datasets. Our results have the practical implication that, when trying to automatically evaluate retrieval effectiveness, researchers should not use a single method, but a (machine-learning based) combination of them.

1. Introduction

In Information Retrieval (IR), test-collection based effectiveness evaluation is a well-known and quite standard method. The whole evaluation process has a cost, in terms of resources needed, effort made by the research community, and also money; thus it is not surprising that researchers tried and are still trying to reduce such costs, for example by using fewer topics, more sensitive effectiveness metrics, shallower pools, or cheaper (usually, crowdsourced) human relevance judgments. A more radical approach is to avoid human relevance judgments altogether, as it has been proposed by several researchers (Aslam & Savell, 2003; Diaz, 2007; Nuray & Can, 2003; 2006; Sakai & Lin, 2010; Soboroff, Nicholas, & Cahan, 2001; Spoerri, 2007; Wu & Crestani, 2003). In this paper, we set out to provide a detailed and complete analysis of the methods for effectiveness evaluation without human relevance judgments, as well as study if they can be fruitfully combined. More in detail, we review the methods that have been proposed (Section 2), we outline the motivations for this work and propose three research questions (Section 3), we describe the experimental setting (Section 4), and we answer each research question (Sections 5–7).

* Corresponding author.
E-mail addresses: kevin.roitero@spes.uniud.it (K. Roitero), andrea.brunello@uniud.it (A. Brunello), giuseppe.serra@uniud.it (G. Serra), mizzaro@uniud.it (S. Mizzaro).

https://doi.org/10.1016/j.ipm.2019.102149
Received 1 March 2019; Received in revised form 27 August 2019; Accepted 20 October 2019

2. Related work

Table 1 summarizes the proposals to use no human relevance assessments when evaluating IR effectiveness. The first proposal is by Soboroff et al. (2001): Their method performs a random sample from the pool of documents (i.e., the documents retrieved by at least one system); the sampled documents are deemed to be relevant, while the remaining ones are non relevant, and the evaluation is performed accordingly. The underlying assumption is that relevant documents tend to be retrieved by many systems, and thus pooled.

Another method, proposed by Wu and Crestani (2003), is based on data fusion, and consists in merging the ranked lists of documents retrieved by each retrieval system querying the same test collection for a certain topic. The idea is to assign a weight to each retrieved document and to use such weights to rank retrieval systems. Thus, good systems are those that retrieve "popular" documents. In the simplest version of the algorithm (WUCv0), the weight, called reference count, sums up the occurrences of each document retrieved by a system which is present in the ranked lists of other systems. The four variants assign a weight to the reference count differentiating the position in which each document appears in the ranked list.

Aslam and Savell's method (2003) measures the similarity of each system to the others (by computing the ratio between the cardinality of the intersection of the documents of the ranked lists and their union) and uses this similarity to evaluate them. This evaluation is highly correlated to Soboroff et al.'s method. One issue is that the average similarity is computed by means of "the grossest possible measure" (Aslam & Savell, 2003, p. 362). This work also presents one of the main criticisms to this approach: The observation that runs are ranked by popularity rather than effectiveness. Such "tyranny of the masses" effect is penalizing for best runs, that are underestimated. We use a slightly modified version of this method, keeping the raw topic scores instead of computing their mean value over the topic set.

The method by Nuray and Can (2003, 2006) consists of three phases: (i) select the runs, (ii) compute the popularity of each document according to various methods, and (iii) the top 30% of the most popular documents are said to be relevant. The run selection can be done in two ways: either "normal", where each run is selected, or "bias", where the runs selected are the top 50% of runs which have a list of retrieved document that is farther from the "norm". The document ranking can be performed according to three strategies taken from theory of voting: "Rank Position", "Borda" (Emerson, 2013), and "Condorcet" (Fishburn, 1977).

The method by Spoerri (2005) selects one run for each participating organization, and forms a set of trials containing five runs (we borrow this terminology from Sakai & Lin, 2010) in a way that each run occurs exactly five times (in different trials); then, it computes the percentage of the set of documents either retrieved by the run exclusively (called "Single"), the set of documents retrieved by all the five runs in the trial ("AllFive"), and the "Single minus AllFive" measure. Finally, to obtain a trial-independent behavior, the three computed measures for each run are averaged over the five trials in which the run occurs.

The method by Sakai and Lin (2010) is very similar to Condorcet method, even if statistically different and more efficient.

All the above methods have been experimentally evaluated using as datasets some TREC test collections as detailed in Table 1

Table 1
The 17 prediction methods used in this paper.

#	Acronym (version)	Accuracy Measures	Datasets	Effectiveness Metrics
Soboroff et al. (2001)		τ, charts	TREC 3,5,6,7,8	MAP
1	SNC			
Wu and Crestani (2003)		r_s	TREC 3,5,6,7,	R-Precision,
	2001	P@10,30,50,100		
2	WUCv0 (Basic)			
3	WUCv1 (Variation 1)			
4	WUCv2 (Variation 2)			
5	WUCv3 (Variation 3)			
6	WUCv4 (Variation 4)			
Aslam and Savell (2003)		τ, ρ,	TREC 3,5,6,7,8	MAP
scatterplots				
7	AS			
Nuray and Can (2006)		r_s	TREC 3,5,6,7	MAP
8	NC-NRP (Normal Rank Position)			
9	NC-NB (Normal Borda)			
10	NC-NC (Normal Condorcet)			
11	NC-BRP (Bias Rank Position)			
12	NC-BB (Bias Borda)			
13	NC-BC (Bias Condorcet)			
Spoerri (2007)		ρ, scatterplots	TREC 3,6,7,8	MAP, P@1000
14	SPO-S (Single)			
15	SPO-A (AllFive)			
16	SPO-SA (Single - AllFive)			
Sakai and Lin (2010)		τ, τ_{ap}, charts, scatterplots	R03, R04, CLIR6-JA, CLIR6-CT, IR4QA-CS	MAP, nDCG, Q-measure
17	SL			

(third column), with the only exception of Sakai and Lin (2010) who used, to run their experiments, also NTCIR collections. The table also shows in the last column the IR effectiveness measure(s) used in each experimental evaluation. The accuracy of the methods[1] has been measured as correlations between the predicted and actual MAP values, again as detailed in the table. Overall (but we will see a more detailed analysis in the following sections) the accuracy of the methods is rather limited and they often do not significantly outperform the original proposal by Soboroff et al. (2001). Hauff, Hiemstra, Azzopardi, and de Jong (2010) noted that the low accuracy might depend on having human intervention (the "manual runs") in the best systems: in the datasets where the best systems are completely automatic, human relevance assessments are less needed. Later, Hauff and de Jong (2010) compared the no assessment and the fewer assessment approaches, finding a rather good correlation and claiming that it is still unclear whether manual assessments are really needed. Moreover, as noted by Sakai and Lin (2010) if the organizers of a test collection initiative can release a so called "system ranking forecast", this can be useful when no "true" assessments are available.

Roitero, Passon, Serra, and Mizzaro (2018a) provide a full re-implementation of such algorithms and discuss their reproducibility. Recent work (Roitero, Soprano, Brunello, & Mizzaro, 2018b; Roitero, Soprano, & Mizzaro, 2018c) proposes to use the described methods in a practical way: reproduce some of the previous result and use the discussed methods to identify a subset of few good topics for retrieval evaluation; Mizzaro, Mothe, Roitero, and Ullah (2018) use the methods in the setting of query and topic performance prediction. When compared to their work, in this paper we use more datasets, more methods, and we also analyze several fusion strategies including those based on machine learning techniques. Moreover, we do not simply aim at reproducibility but we also focus on comparisons across methods and collections, as we detail in the following.

In our experiments, we will use the methods listed in Table 1. We believe that we have included all the proposals from the literature, with the only exception of Diaz's one (2007): We leave it for future work since it uses the text of topics and documents, and we are interested in providing a complete and uniform account of the methods that do not use the text of documents, nor the topic descriptors.

3. Research questions

When analyzing the literature on the methods for effectiveness evaluation without relevance judgments, one can notice that their accuracy is often evaluated using different measures, on different datasets, and on the basis of different effectiveness metrics (see the last three columns of Table 1). This means that it is not clear what the relative accuracies are, and how these vary across the datasets. Therefore, our first researchquestion is aimed at establishing a solid baseline for these effectiveness evaluation prediction methods:

- **RQ1**: What is the comparative accuracy of the various methods for effectiveness evaluation without relevance judgments when they are evaluated under the same conditions? What about different collections, and different measures?

Some comparisons do exist, although they are made in a rather implicit and incomplete way. The most similar works to ours are those by Hauff et al. (2010) and Sakai and Lin (2010). Hauff et al. present a comparison of most of the methods, but their aim is to study the variations across topics, and to understand what happens when selecting the "right" topics subset. The work by Sakai and Lin is more related to our RQ1, but again it focuses on just 6 methods (while we analyze 17 of them) and uses 2 TREC and 3 NTCIR collections (instead, we test them on 14 TREC collections). Also, we report a more complete set of accuracy measures and, finally, we consider the actual accuracy of the methods as a means for the remaining two research questions, rather than as an end in itself.

Going beyond accuracy figures, one might wonder whether the methods are really different from each other, or rather whether they all measure, more or less, the same thing. Our second research question specifically addresses this issue:

- **RQ2**: What are the relationships among the methods? Do they tend to measure the same phenomenon with almost no differences, or is there any variability that can be exploited?

If the methods are indeed different, it is natural to ask whether this diversity can be exploited by combining them. Therefore, our third and last research question is:

- **RQ3**: Can the methods be combined in an effective way? What combination strategies lead to the highest accuracy?

4. Experimental setting

We describe the overall setting common to all the experiments. We present the basic definitions, the measures, and the datasets used.

4.1. Notation, background, and terminology

Fig. 1 shows the basic outcome of a test collection evaluation exercise, represented as a matrix and two vectors. Each row of the

[1] In an attempt of avoiding confusion, and consistently with other authors (Nuray & Can, 2006; Sakai & Lin, 2010; Wu & Crestani, 2003), we reserve the term "effectiveness" for retrieval effectiveness and "accuracy" for the accuracy in predicting system effectiveness by a method.

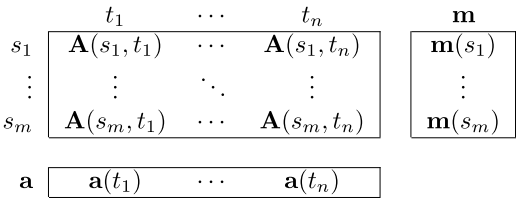


Fig. 1. AP ($\mathbf{A}(s_i, t_j)$), MAP ($\mathbf{m}(s_i)$), and AAP ($\mathbf{a}(t_j)$) for n topics and m systems (adapted from Mizzaro and Robertson (2007); Roitero et al. (2018a)).

matrix is a system s_i (or run), each column is a topic t_j, and each cell (i, j) is the effectiveness of system s_i on topic t_j. Averaging each row on the n topics one obtains a measure of system effectiveness (for all systems, this is the column vector on the right); averaging each column on the m systems one obtains a measure of topic ease (the row vector on the bottom).

In this paper we focus on Average Precision (AP) as the effectiveness measure. We use the following notation. $\mathbf{A}(s_i, t_j)$ is the AP value of system s_i on topic t_j, $\mathbf{A}$ is the matrix of AP values, and $\mathbf{m}$ and $\mathbf{a}$ are the vectors of the MAP (Mean AP) and AAP (Average AP) values. Although we acknowledge that "Mean Average Precision" is a questionable term, we use it to distinguish from both average precision (the individual effectiveness value of a system on a topic) and Average Average Precision (Mizzaro & Robertson, 2007; Roitero et al., 2018a) (a measure of topic ease).

Turning to the effectiveness values predicted by a method, we denote with $\hat{\mathbf{A}}$ the matrix of predicted AP values, and with $\hat{\mathbf{m}}$ and $\hat{\mathbf{a}}$ the vectors of predicted MAP and AAP values, respectively. We will first and mainly focus on MAP ($\mathbf{m}$ and $\hat{\mathbf{m}}$) in this paper, as others have done, but we will also study and exploit AAP and AP. Thus, the main question will be the accuracy of $\hat{\mathbf{m}}$ as a prediction of the ground truth $\mathbf{m}$, but we will also study the accuracy of $\hat{\mathbf{a}}$ and $\hat{\mathbf{A}}$ as predictions of the original $\mathbf{a}$ and $\mathbf{A}$.

4.2. Accuracy measures

One can imagine several accuracy measures, and indeed many alternatives have been used in the past studies (see Table 1). Kendall's or Spearman's rank correlations are reasonable choices when one is interested in the order of the values, an option that is quite common when a ranking of the systems according to their effectiveness is desired. Often, the top positions of a rank are the most important, and in such a case a top-heavy rank correlation like Tau-AP (τ_{ap}) (Yilmaz, Aslam, & Robertson, 2008) can be used. Pearson's linear correlation can be used when one wants to understand if the predicted values have a linear relation with the original ones, i.e., when measuring the ability of methods in predicting the exact values, not just the ranks. Correlations are a natural measure when working on the vectors $\mathbf{m}$ and $\mathbf{a}$, but they can be used also on the matrix $\mathbf{A}$ by converting it to a vector. Vectorization of a matrix is a linear operation that concatenates all the columns of the matrix into a column vector. However, for a matrix a similarity measure based on matrix difference is also meaningful. In the following we will use:

- Pearson's linear correlation (denoted with ρ, i.e., $\rho(\mathbf{m}, \hat{\mathbf{m}})$, $\rho(\mathbf{a}, \hat{\mathbf{a}})$, $\rho(\mathbf{A}, \hat{\mathbf{A}})$, the latter being the correlation between the vectorized AP matrices);
- Kendall's rank correlation ($\tau(\mathbf{m}, \hat{\mathbf{m}})$, etc.);
- Spearman's rank correlation ($r_s(\mathbf{m}, \hat{\mathbf{m}})$, etc.);
- Tau-AP, a top-heavy rank correlation (Yilmaz et al., 2008) ($\tau_{ap}(\mathbf{m}, \hat{\mathbf{m}})$);
- Matrix difference ($\delta(\mathbf{A}, \hat{\mathbf{A}}) = \frac{1}{nm} \sum |(\hat{\mathbf{A}}(i, j) - \mathbf{A}(i, j)|$).

4.3. Datasets

Table 2 summarizes the 14 datasets considered in this paper, showing an acronym (used in the following), a longer name, the year, the number of topics (m) and of systems (n), as well as the topic identifiers in the dataset. We use several TREC collections, spanning 20 years, selecting among those having large enough sets of topics and systems/runs. For each dataset we produced the corresponding table as in Fig. 1. The three Web track collections (last three rows) adopted a non-binary notion of relevance; we computed AP values collapsing relevance levels -2 and 0 into irrelevant and 1, 2, and 3 into relevant, and then running trec_eval.[2] The code to conduct the experiments can be found at https://github.com/KevinRoitero/LeToE-Code.

5. RQ1: Individual methods accuracy

We now turn to presenting and discussing the results of our experiments. We start by focusing on RQ1, aimed at quantifying the accuracy of the individual methods. Figs. 2–9 show the accuracy of prediction as eight box-plot charts. These charts show the accuracy (Y-axis) of the individual methods (X-axis) on each dataset (legend). As indicated on the Y axes, the prediction is of MAP ($\mathbf{m}$) in the first four charts, of AP ($\mathbf{A}$) in the following two, and of AAP ($\mathbf{a}$) in the last two; accuracy is measured by τ, r_s, ρ, τ_{ap}, and δ. Each dot shows the accuracy of the prediction of a method on a dataset. So, for example, the dot on the top-left of the first chart

[2] See https://trec.nist.gov/trec_eval/ .

Table 2
The 14 datasets used in this paper.

	Acron.	Name	Year	Topics	Runs	Used Topics
1	TREC3	Ad Hoc	1994	50	40	151–200
2	TREC5	Ad Hoc	1996	50	61	251-300
3	TREC6	Ad Hoc	1997	50	74	301-350
4	TREC7	Ad Hoc	1998	50	103	351-400
5	TREC8	Ad Hoc	1999	50	129	401–450
6	TREC01	Ad Hoc	2001	50	97	501-550
7	R04	Robust	2004	249	110	301–450, 601–700[a]
8	TB04	TeraByte	2004	49	69	701-750[b]
9	R05	Robust	2005	50	74	See (Voorhees, 2003, Figure 1)
10	TB05	TeraByte	2005	50	58	751-800
11	TB06	TeraByte	2006	149	61	701–850[b]
12	W11	Web Track	2011	50	61	101-150
13	W12	Web Track	2012	50	48	151-200
14	W13	Web Track	2013	50	55	201-250

[a] 672 excluded.
[b] 703 excluded.

(Fig. 2) is $\tau(\mathbf{m}, \hat{\mathbf{m}})$, Kendall's τ correlation of the actual MAP values in TB04 with the MAP values predicted by SNC. The box-plots synthetically represent the distributions of the accuracy values (dots) for each method by showing the 95% range, the 25th and 75th percentiles, and the median, as well as the mean (the dashed black horizontal line). The rightmost panes will be discussed in Section 7.1.

Analyzing each chart individually, we can make the following observations. Let us start by focusing on measuring accuracy of MAP prediction by τ (Fig. 2). On average, i.e., looking at medians and means, the three most accurate individual methods seem to be SNC, NC-NB, and NC-NC. Other methods (WUCv1, NC-BRP, and NC-BB) look almost as accurate, other ones (WUCv0, WUCv2, WUCv3, NC-BC, AS, and SL) are not much less effective and the last five (WUCv4, NC-NRP, CPO-S, SPO-A, and SPO-SA) are clearly outperformed, with WUCv4 showing a very low accuracy.

To better understand the differences in accuracy, we ran a paired Wilcoxon's significance test[3] (Wilcoxon, Katti, & Wilcox, 1970) between the methods, considering the series of τ values on the 14 datasets; we used the Bonferroni's method (Dunn, 1961) to deal with multiple comparisons. We found no statistical significant difference between the top six methods.

There is a somehow consistent behavior of datasets across methods: some of them have steady higher τ values (e.g., TB04) other ones have lower τ (e.g., TREC7). Clearly, MAP prediction is easier for some datasets, as others have already reported (Hauff, Hiemstra, & de Jong, 2008).

There is some variation over datasets, and this variation is quite similar across methods (i.e., the sizes of the box-plots, representing the inter-quartile range, are quite similar). When looking at individual collections, there are many exceptions to the average behavior: for example, SNC is slightly less accurate than SL for R04, TREC8, and TREC01. This means, when considered with the just noted consistent variation over datasets, that if a researcher wants to evaluate effectiveness on a new unseen dataset, it is not completely clear which method should be used, as well as which is the expected accuracy of the method.

Fig. 10 provides a dual representation of the same data for MAP τ (the other accuracy measures show a similar behavior). Here we group in each box-plot all the methods on a single dataset: Figs. 2–9 show the variability of the different methods when applied to different datasets and Fig. 10 shows the consistency of the different methods on each dataset. Clearly, the average accuracy and variance of the methods depend on the specific dataset on which they are applied.

Going back to Figs. 3–5, the observations on the basis of these three charts are quite similar Indeed, there is no significant statistical difference among the top accuracy methods (although the specific results are not reported here for brevity reasons). The fact that τ_{ap} outcome is very similar means that method accuracies remain relatively stable even when weighting more the top ranks. In the 5th chart in Fig. 6 we see that the situation is slightly different for AP, with accuracy predicted by ρ. Correlation values in the AP ρ chart (usually below 0.6) are clearly lower than in the previous MAP ρ chart (usually above 0.6): Predicting AP is more difficult than predicting MAP. Also, here the most accurate method is now AS, which does not only outperform the other ones, but also shows a much smaller variation over the collections. Considering statistical significance, AS is indistinguishable from NC-NB, NC-NC, NC-BC, and SL. These five methods are always statistically significantly more accurate than the other ones.

Accuracy of AP prediction measured by δ is shown in the 6th chart in Fig. 7; since smaller differences are better, the best methods are those with the lower values, and the scale on the Y axis is inverted for consistency with the other charts. When using AP δ as the accuracy measure, the top three methods are SNC, AS, and SPO-A: a different set with respect to that in the previous AP ρ chart. Also, they are statistically significant better than all the other ones at the.01 level. Accuracy in predicting AP seems to be neither necessary nor sufficient to accurately predict MAP, and the ρ and δ measures do not agree much. Although AP δ measure seems quite a reasonable one in principle, it turns out that its results show a different behaviour with respect to those exhibited by the other measures. This is probably due to normalization: since some methods try to predict AP values, while others are just interested in the

[3] See https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.wilcoxon.html.

rank, we normalized all the predicted AP values into [0,1],[4] to be able to compare the predicted AP values obtained from the methods with the real AP values that are within the same range. However, the normalization that we used, although standard, might have harmed some methods more than others. Because of the difference between the behaviour of δ and the other variables, in the following we do not report δ anymore.

Accuracy in AP prediction might be considered an artificial measure, but this would be a mistake: it has a practical usefulness, since one might be interested in knowing the effectiveness of a specific system on a specific topic, or in comparing the effectiveness of all systems on specific topics. Moreover, a better AP prediction could be related to a better AAP prediction, as we now discuss. The last two charts in Figs. 8 and 9 show accuracy in AAP prediction. Whereas, when predicting MAP, rank-based correlations seem a better option as accuracy measure than linear correlation (usually one is interested to know which is the best system), for AAP the choice is less clear (ranking the topics by difficulty seems as interesting as knowing their difficulty values). Anyway, for both ρ and r_s, AS is again (as for AP ρ) the best method on average, although its variation is not lower than the other methods as it was in the AP case. When considering statistical significance, however, AS is not more accurate than NC-NB, NC-NC, NC-BB, NC-BC, and SL.

Finally, we remark again the particularly low accuracy of WUCv4. This might be due to our failure in reproducing its normalization algorithm, which is not fully detailed in the original paper (Wu & Crestani, 2003). In the following we exclude this method from most of our analyses.

6. RQ2: Relations between methods

Having established a common ground consistent with the previous literature, as well as some baselines to compare to, we now focus on RQ2, a question which is more central for our paper. Our analysis is aimed at understanding if the various individual methods measure different aspects or are very correlated. The heat-map in Fig. 11 shows ρ correlations between the AP values predicted by the individual methods, for each pair of the 17 individual methods, and for the two datasets R05 (bottom left triangular part) and TB06 (upper right). In other terms, the heat-map contains, for each pair of methods *i*, *j*, $\rho(\hat{\mathbf{A}}_i, \hat{\mathbf{A}}_j)$. Observe that, given the accuracy measures that we use, the heat-map is symmetric. Thus, we chose to report the results for two datasets into a single heat-map, in which the upper triangular part shows the outcomes on a dataset, and the lower triangular part on the other one. Presenting two datasets in the same heat-map has also the advantage of clearly emphasizing graphically that the values are similar across datasets. We do not show the heat-maps for the other 12 datasets, that are anyway very similar. A quick visual inspection immediately shows three properties:

- Some methods are highly correlated with each other (the darker cells and triangular/rectangular areas).
- Conversely, some methods do not seem to correlate well (lighter areas): they are producing quite different predictions. Some of these methods are not accurate (e.g., WUCv4), but the low correlations among SNC, AS, SL, and NC-* methods are interesting: these are accurate methods that do not correlate well. This result is promising when considering methods combinations: the low-correlation methods might provide complementary information.
- The correlations are quite consistent across the two datasets (the triangular areas usually become rectangular across the diagonal, i.e., when considering the two datasets together). This also happens for the other datasets.

The two charts in Fig. 12 provide some further details. The MAP scatterplot (left) is an example of the rather high, though not perfect, correlation between SNC and NC-NB when predicting MAP ($\rho(\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_9)$ using the numbers in Table 1). The AP hexbin scatterplot[5] (right) is a more detailed representation of the.77 ρ value in the last row, 7th column of the heat-map, and shows that even two of the most accurate methods across measures (AS and SL) do not correlate much in terms of their AP prediction. It is also clear that the relation between AS and SL in this case is not linear.

In summary, it is clear that the methods do show some differences. Given the low correlations, occurring even on accurate methods, it makes sense to try to combine them;

7. RQ3: Combining the methods

We now turn to our last research question RQ3, i.e., whether it is possible to combine in an effective way the individual prediction methods, and which is the best approach. We test two approaches to methods combination: a first one based on data fusion techniques, and a second one based on machine learning.

7.1. Oracle combination

First, we compute an optimal result in which an oracle selects the best method. In particular, for each collection, we select the method that achieves higher correlation (we call it the “Oracle Method”). This is not the best that can be done, it is rather a sub-optimal best; indeed, it can be that combining together a subset of the methods will lead to achieve higher correlation values than the

[4] We used the standard normalization $\frac{\mathbf{A} - \min(\mathbf{A})}{\max(\mathbf{A}) - \min(\mathbf{A})}$, and we also tried another standard normalization ($\frac{\mathbf{A} - \text{mean}(\mathbf{A})}{\text{std}(\mathbf{A})}$) but results looked very similar.

[5] A normal scatterplot would be too cluttered as it would have, in this case, $74 \times 50 = 3700$ points. The hexbin scatterplot bins the points in hexagonal areas, and shows the density of points by a color gradient (logarithmic in our figure, as shown on the right).

oracle. Nevertheless, the oracle method sets a simple and reasonable upper bound to aim to with the combination of one or more methods.

Figs. 2–9 show, in the rightmost panes, the oracle accuracy of prediction for the eight box-plot charts. Analyzing the results, we can make the following observation: concerning all of AP, MAP, and AAP, the correlation values the oracle achieves are always similar to any other method; furthermore, in cases where the oracle has a higher median correlation value, the differences from the oracle to any other method are not statistically significant. This means that any trivial combination of the methods can improve only partially the correlation values obtained by the best method. On the other hand, if we compare the oracle with the worst performing method, we can observe that the oracle has always higher correlation values, and this difference is statistically significant. Based on these observations we conclude it is worth trying to combine the methods; indeed, this combination may be used when evaluating automatically a new collection, without having any prior knowledge of which of the methods will perform better. More details are presented in the following sections.

7.2. Data fusion approaches

In the following subsections we detail the data fusion approaches used in this paper: we define the setting (Section 7.2.1), list the algorithms that we use (Section 7.2.2), and present the results (Section 7.2.3).

7.2.1. Data fusion setting

The basic idea is to define a fusion operation that merges the results of the individual prediction methods. We can sketch the situation using the following three equations:

$$\hat{\mathbf{m}}_* = \mathrm{DF}(\hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_q) \tag{1}$$

$$\hat{\mathbf{a}}_* = \mathrm{DF}(\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_q) \tag{2}$$

$$\hat{\mathbf{A}}_* = \mathrm{DF}(\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_q). \tag{3}$$

Focusing on MAP first (Equation (1)), the MAP vectors $\hat{\mathbf{m}}_i$ predicted by the individual methods are combined into $\hat{\mathbf{m}}_*$ by a data fusion function DF. In our experimental setting we have $q = 17$ individual methods (though we will use fewer as detailed below in Section 7.2.2). Besides working directly on MAP (and, symmetrically, on AAP, Equation (2)), we also try the same techniques on AP values (Equation (3)). This makes sense in an attempt to avoid losing information: the predicted AP matrices $\hat{\mathbf{A}}_i$ are combined into $\hat{\mathbf{A}}_*$. The latter is the only possible approach when aiming at predicting AP; conversely when aiming at MAP (and AAP) prediction, two approaches can be used, as one can directly predict MAP and AAP, or predict AP and then average the obtained values.

7.2.2. Data fusion algorithms

We use four basic and well known data fusion approaches (some of which are also used by some individual methods, see Section 2):

- *Average function*. Arithmetic average of predicted values $\hat{\mathbf{m}}_i$, $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{A}}_i$. We therefore obtain three data fusion functions: MAP-AVG, AAP-AVG, AP-AVG.
- *Rank*. Combination of $\hat{\mathbf{m}}_i$, $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{A}}_i$ according to the rank position of each systems: MAP-RP, AAP-RP, AP-RP. In summary, this approach assigns a score based on the rank in which the system occurs. Let us consider the following toy example, with two systems s_i, s_j and three methods. Suppose the system s_i occurs in the 1st, 2nd, and 3rd position in the ranked list of $\mathbf{A}$ inferred from the respective methods, and the system s_j occurs in the 2nd, 1st, and 1st position. The score for the system s_i in the fusion list is $1/(1/1 + 1/2 + 1/3) = 0.55$, and the score for the system s_j is $1/(1/2 + 1/1 + 1/1) = 0.4$. Thus, in the fusion list, s_j will be ranked before s_i (the lower the score the better) and their respective scores will be 0.4 and 0.55.
- *Borda count* (Emerson, 2013). Predicted values $\hat{\mathbf{m}}_i$, $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{A}}_i$ are treated as expression of preferences, which are then combined based on the rank position of the systems: MAP-B, AAP-B, AP-B. In summary, the Borda count assigns a score to each so called candidate considering the reverse proportion of its ranking. Referring to the previous example, the score for the system s_i in the fusion list is $(3 - 1) + (3 - 2) + (3 - 3) = 3$, and the score for the system s_j is $(3 - 2) + (3 - 1) + (3 - 1) = 5$. Thus, in the fusion list, s_j will be ranked before s_i (the higher the score the better) and their respective scores will be of 5 and 3 (to be then normalized in [0,1]).
- *Condorcet* (Fishburn, 1977). A majority method of pairwise comparisons between ranked retrieval systems: MAP-C, AAP-C, AP-C. In summary, in the Condorcet method the winner is the candidate that is preferred to any other candidate, when compared to the opponents one at a time according to a scoring system based on preferences. The Condorcet method works as follows (see the example above): for each pair of systems, in this case just (s_i, s_j), we build a table in which we count for the three methods how many times s_i is preferred to s_j (in this case we count a "win" for s_i), how many times it happens the opposite (in this case we count a "lose" for s_i), and how many times there is no preference (in this case we count a "tie" for both systems). In our example we have that for method 1 s_i is preferred to s_j, for method 2 s_j is preferred to s_i, and for method 3 s_j is preferred to s_i. Thus, s_i will have win = 1, lose = 2, tie = 0, and s_j will have win = 2, lose = 1, tie = 0. Then, we rank the systems according to their wins, lose, and tie values.

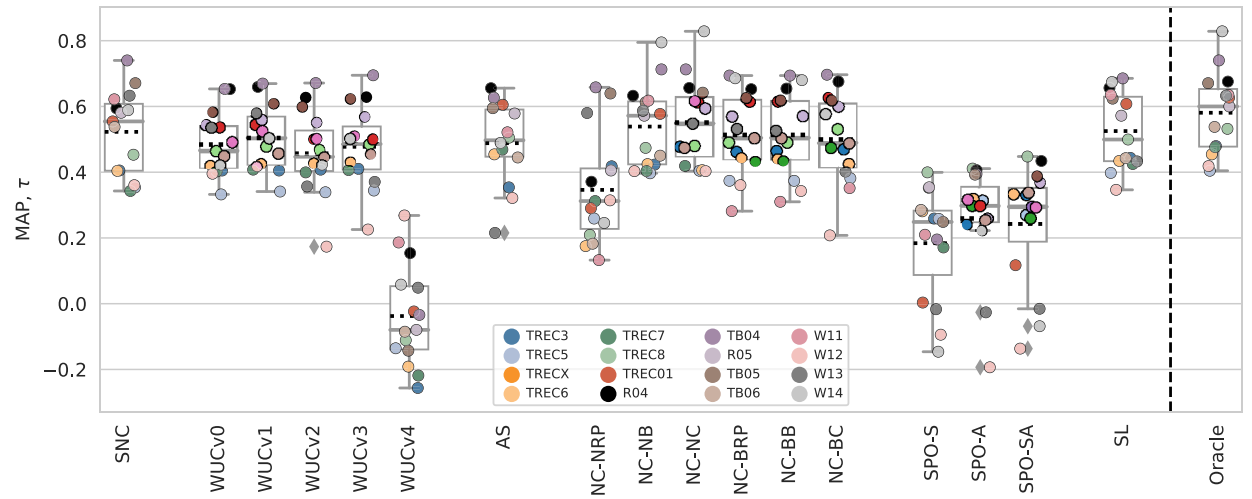


Fig. 2. Accuracy of the methods: MAP τ.

Since not all the individual methods aim to return AP values on the same scale of the original ones, we again apply the same standard normalization operation to map the predicted AP values into [0,1] (see Footnote [4]). We do not use WUCv4 in the data fusion approaches, given its low accuracy. For MAP prediction, we also try removing the four worst individual methods NC-NRP, SPO-S, SPO-A, SPO-SA as found in Section 5. The obtained data fusion functions are labeled with an "s" (for "selected"): MAP-AVGs, MAP-RPs, MAP-Bs, and MAP-Cs.

7.2.3. Results

Fig. 13 shows the results for MAP τ (as in Fig. 2). The leftmost pane shows again the accuracy of the three top individual methods, according to the median (i.e, these are selected from Fig. 2); the other two panes show the accuracy of using the data fusion approaches. Although combining methods seemed an interesting and promising idea, and despite the use of a spectrum of state of the art data fusion techniques, it is clear that no accuracy improvement is obtained with the data fusion approaches. Instead, usually the combinations by data fusion are less accurate in a statistically significant way than the best individual methods. Results do not change when using the other accuracy measures: all the corresponding charts to Figs. 2–9 look very similar to Fig. 13, and therefore we omit them for brevity. Anyway we remark that reporting the negative results and failed attempts is important, especially if they are obtained using non trivial techniques; this will prevent future researchers to waste time and resources trying the same ineffective approaches. This position is also supported by other authors: both in general (Fanelli, 2012; Knight, 2003), and within IR (Ferro, 2017; Ferro et al., 2016).

One possible reason for the ineffectiveness of the data fusion approaches is that they somehow "go towards the mean" of the individual methods being combined, i.e., they produce an outcome that is similar to the average of the individual methods, but they cannot improve the overall effectiveness: the best individual methods are somehow hampered by the other ones, and this negative

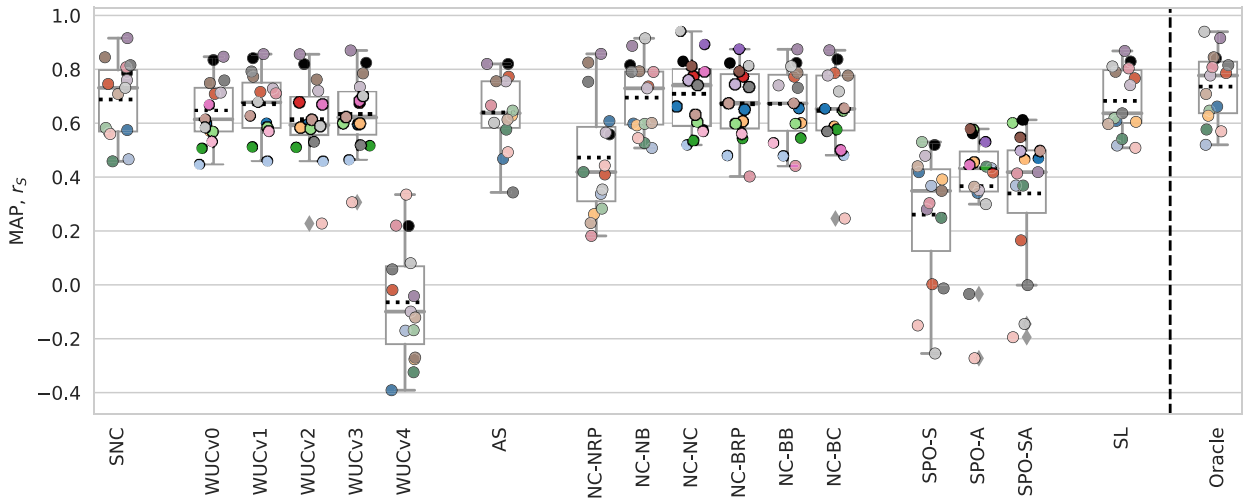


Fig. 3. Accuracy of the methods: MAP r_s.

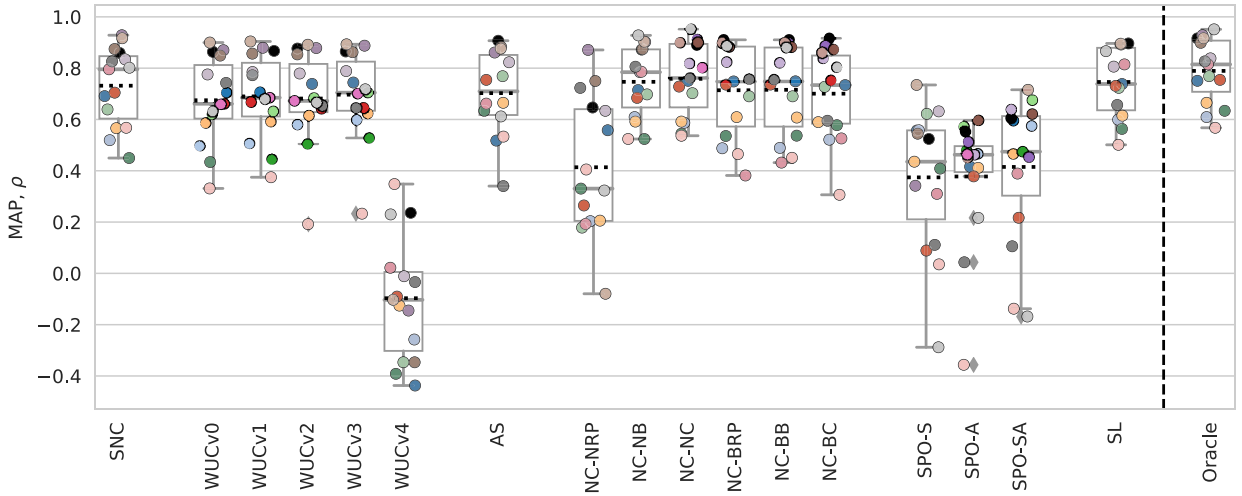


Fig. 4. Accuracy of the methods: MAP ρ.

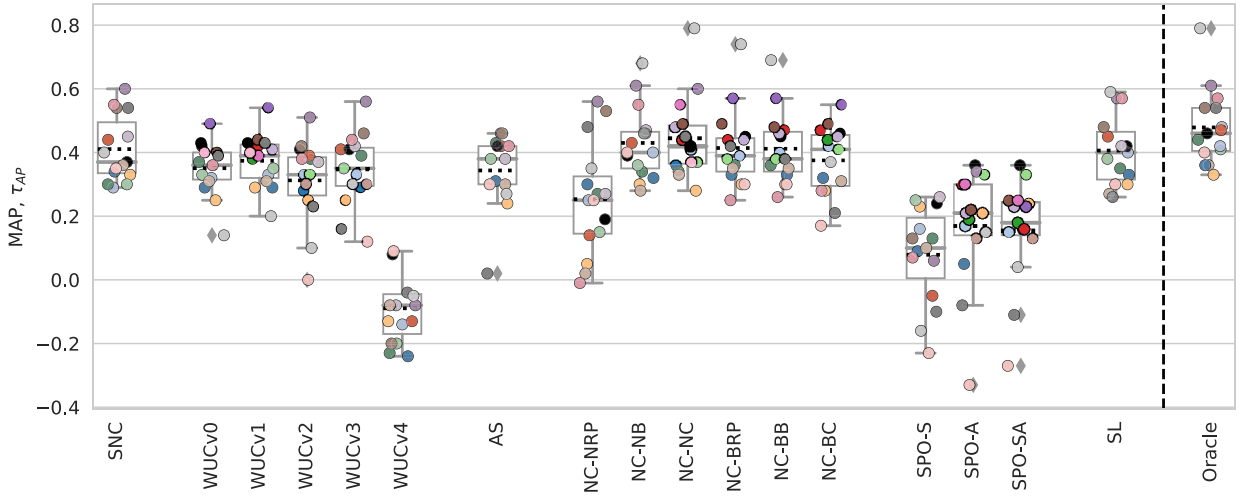


Fig. 5. Accuracy of the methods: MAP τ_{ap}.

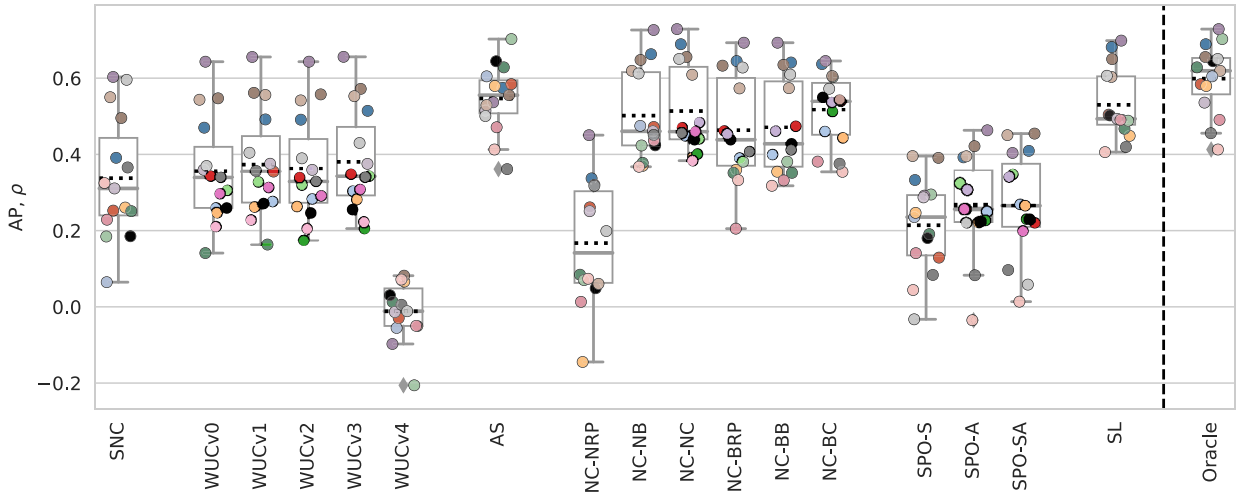


Fig. 6. Accuracy of the methods: AP ρ.

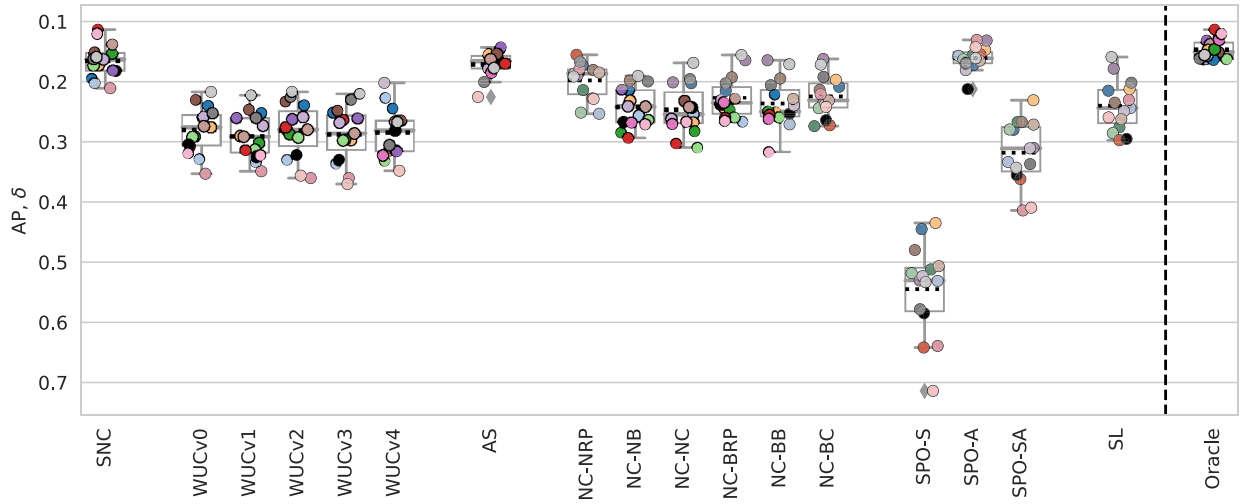


Fig. 7. Accuracy of the methods: AP δ.

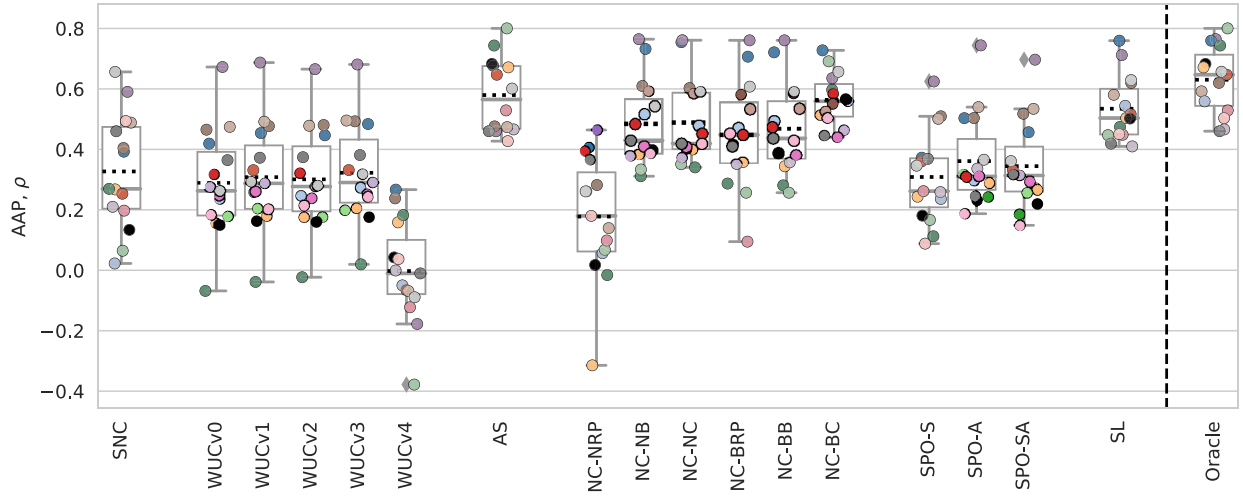


Fig. 8. Accuracy of the methods: AAP ρ.

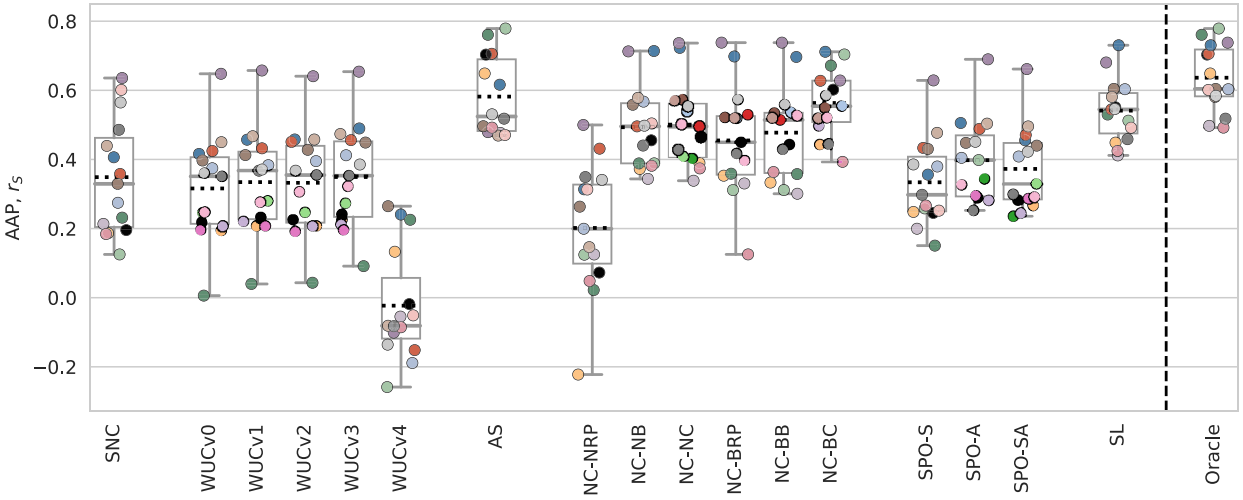


Fig. 9. Accuracy of the methods: AAP r_s.

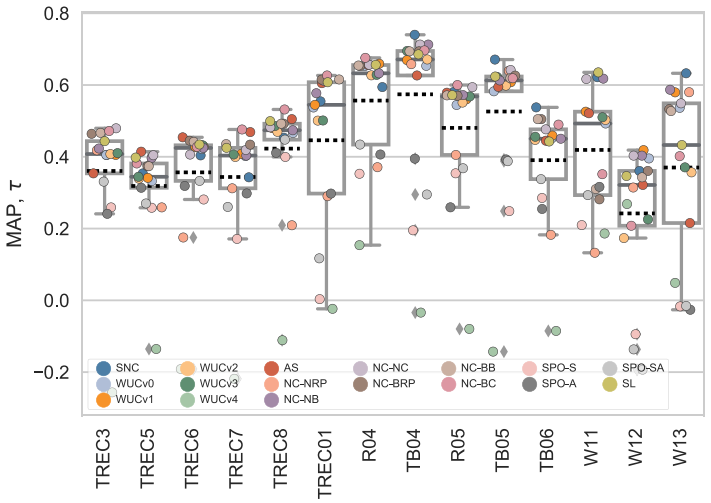


Fig. 10. Accuracy (MAP τ) of the methods across datasets.

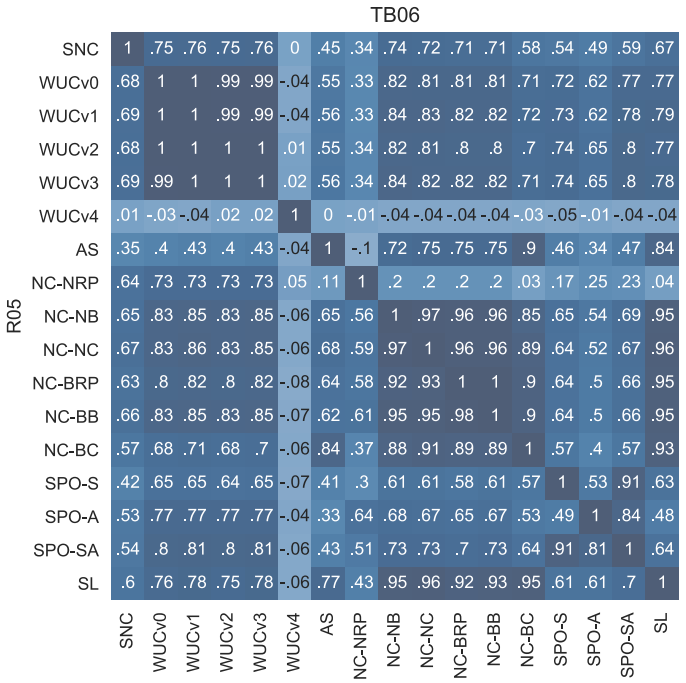


Fig. 11. ρ correlations between the AP values predicted by the methods, on both the R05 and TB06 datasets.

effect remains also when removing the worst individual methods as we tried with our "selected" ("s") approaches (see the end of Section 7.2.2). Indeed, the "selected" methods are more accurate than the all inclusive ones, but still not so effective as the individual methods.

Perhaps effectiveness could be improved with more tailored fusion approaches and/or more sophisticate normalization strategies, but these might depend on the method and on the dataset and it does not seem simple nor promising to follow this approach any further. We leave that for future work and we instead turn to a more general approach, which can be more promising considering the results shown by the oracles in Figs. 2–9.

7.3. Machine learning approaches

In the following subsections we detail the Machine Learning approaches we use in this paper: first we discuss the setting (Section 7.3.1), the algorithms we use (Section 7.3.2), and the ML results (Section 7.3.3). Then, we report on a rather natural technique to be applied in our setting: Transfer Learning (Section 7.3.4).

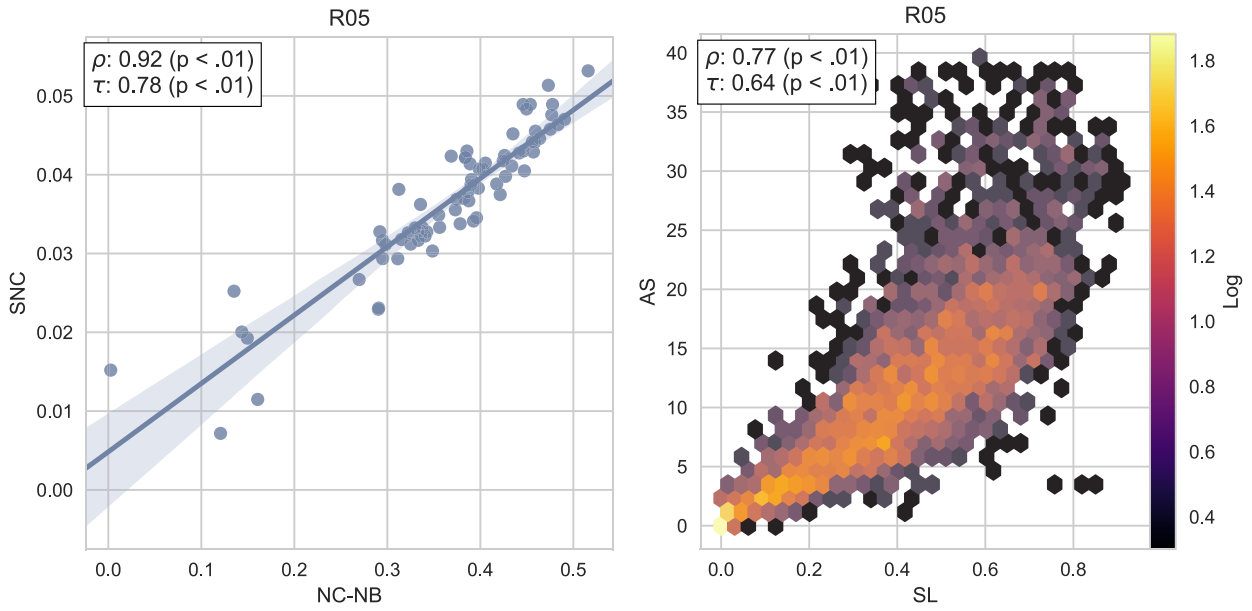


Fig. 12. Scatterplot of MAP values predicted by NC-NB and SNC; hexbin scatterplot of AP values predicted by SL and AS.

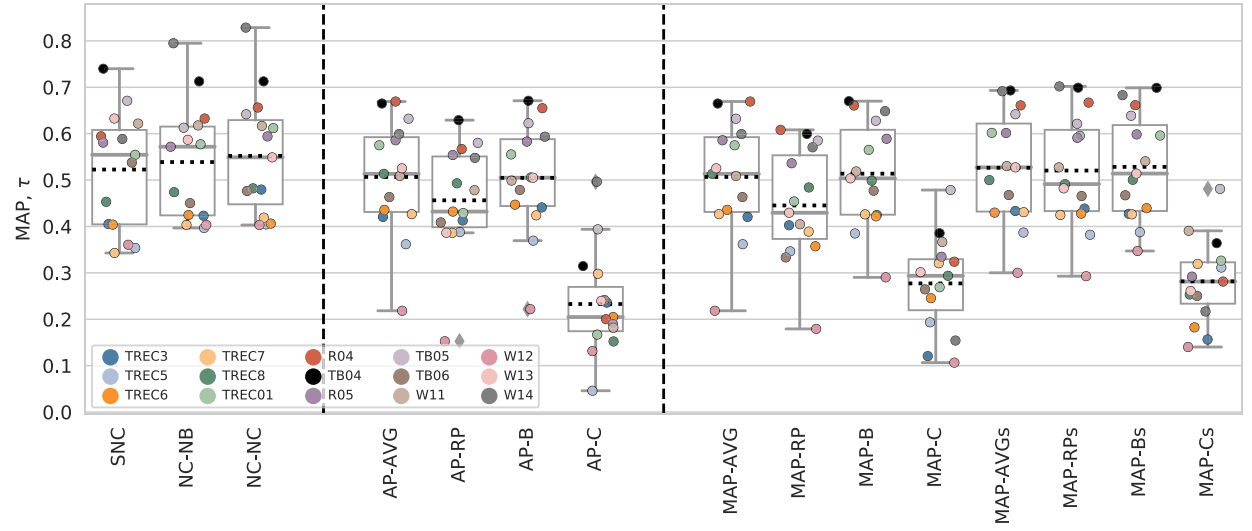


Fig. 13. Accuracy of the data fusion approaches, compared with the three best individual methods from Fig. 2: MAP τ.

7.3.1. Machine learning setting

Instead of relying on data fusion approaches, we now turn to the issue of automatically learning the DF functions of Equations (1), (2), and (3) relying on historical competitions data. In such an experimental setting, we consider, in turn, each of the collections as the test set, while the "historical" training set is composed of all the instances belonging to the previously released collections, sorted according to their release year (see Table 2).[6] To make an example, if we are considering as test collection TREC8, released in 1998, then the training set contains the collections released before 1998: TREC3, TREC5, TREC6, and TREC7. Observe that, due to our setting, TREC3 can never be considered as test data: since it is the oldest collection, this implies that it would not have any older collection to be used as training data. We generate features by running the individual methods (see Table 1), thus obtaining their predicted values $\{\hat{\mathbf{A}}_1, ..., \hat{\mathbf{A}}_q\}$, while the labels are the actual MAP, AAP, and AP values, which for past data are also considered to be known.

Since predictors and labels are numeric continuous values, we focus on a subset of machine learning algorithms, namely regression algorithms. Although ranking (e.g., to rank systems according to their effectiveness) and classification (e.g., into easy/

[6] If a collection is released in the same year as the test one, we choose not to consider it.

difficult topics) are also possible, we leave those to future work.

To train regression algorithms for estimating MAP the most intuitive choice is to consider a dataset with a row for each run, and a column for every distinct combination of individual metric and topic (plus a column for the label reporting the MAP value of the run). However, such an approach has two criticalities: first of all, the resulting training set is often too small for machine learning algorithms to be trained effectively, given the number of features. For example, even considering the collection with the largest number of runs (TREC8, see Table 2), the samples in the training set would be just 129 (equal to the number of runs), while the number of columns would be 851 (50 topics $\times$ 17 methods, plus the MAP label). Secondly, this kind of representation is strictly tied to the number, kind, and arrangement of topics. Therefore, training samples of a collection may only be combined with other collections sharing the same format of topics. Equal considerations apply when predicting AAP.

To overcome these limitations we focus on predicting AP values instead of MAP or AAP. By doing so, the dataset has a row for each distinct combination of run and topic, and a column for each individual method (plus a column for the label, which is the AP value of the run on the topic). This kind of feature representation has three important characteristics: first of all, it is fine-grained, since it includes all the estimated values of each individual method; second, for each collection the training set is much larger than previous proposal (on TREC8 we will have 129 runs $\times$ 50 topics = 6450 rows, and just 18 columns); third, its dimensionality and column arrangement is totally independent of the format of topics, therefore training samples of different collections can be combined together by simply stacking the rows.

By relying on the results of Figs. 2–10, we removed feature WUCv4 given its consistently poor performance (as observed in Section 5), therefore considering 16 methods, instead of the original 17.

7.3.2. Machine learning algorithms

We tested several machine learning algorithms, all implemented using the following Python 3.5 libraries: Scikit-learn,[7] and Keras.[8] We report the results for twelve of them:

- *LinearRegression* (Witten, Frank, Hall, & Pal, 2016) (LR in the following), the standard linear regression technique.
- *RandomForest* (Breiman, 2001) (RF in the following), an ensemble learning method that operates by constructing a set of decision trees during training, and outputting the average prediction of the trees when a new instance has to be predicted.
- *Ridge Regression* (Hoerl & Kennard, 1970) (RIDGE in the following), a regression algorithm that implements L2 regularization, and uses as objective function the minimization of the sum of square of coefficients.
- *Bayesian Ridge Regression* (Park & Casella, 2008) (BAYRIDGE in the following), a regression algorithm that uses Bayesian modeling and spherical Gaussian priors.
- *Lasso Regression* (Tibshirani, 1996) (LASSO in the following), a regression algorithm that implements L1 regularization, and uses as objective function the minimization of the sum of absolute value of coefficients.
- Neural Network (NN-*epochs-loss* in the following): A neural network regression model with a Sequential architecture composed of two dense connected layers: the first layer with 16 neurons, initialization function “uniform” and activation function “ReLu”; the second one with dimension one, initialization function “normal”, no activation function. We trained the model using “Adam” as optimizer, “MSE” and “MAE” as Loss functions (number of epoch set to 10 and 100).
- *Deeper Neural network* (DNN in the following), a neural network regression model with a sequential architecture composed of three dense connected layers: the first layer with 32 neurons, initialization function “uniform” and activation function “ReLu”; the second one with 16 neurons, initialization function “uniform” and activation function “ReLu”; the last layer with dimension one, initialization function “normal”, no activation function. We trained the model using “Adam” as optimizer, “MSE” as Loss functions (number of epoch set to 10). We did some experimentation with 100 epochs, but results where worst than with 10.
- *SVM*, which is the Python implementation of the library for Support Vector Machines (Chang & Lin, 2011), that are also capable of performing support vector regression. Specifically, we tested two nonlinear kernels: *PolyKernel* (SVM-P in the following) and *RBFKernel* (SVM-E in the following), both within the *nu-SVR* SVM type and with the normalization step active.
- *Learning to Rank* (LtR in the following), which is typically used in information retrieval to predict the correct order of retrieved documents (Liu, 2009). In this work, we use it to rank the systems of various competitions. Specifically, we rely on Python’s *XGBRegressor* package with a *rank:pairwise* objective.

To avoid over-fitting phenomena, as well as to ease reproducibility, we did not fine-tune the parameters of the algorithms, but instead relied on their default values, with the exception of *XGBRegressor*, that typically requires a tuning phase to get the best results: specifically, to select the most appropriate choices for the model parameters, we performed a tuning phase, relying on `GridSearchCV` method from `Scikit-learn` library. As its name suggests, it performs a grid search in a given parameter space, returning their best combination, according to the performance exhibited by the trained model. Such score has been evaluated through 4-fold cross-validation on the training data. Table 3 reports the tuned parameters, together with their search space and optimal values.

[7] https://scikit-learn.org/stable/ .
[8] https://keras.io/ .

Table 3
Search space and optimal values of the parameters used in `XGBRegressor`.

Parameter Name	Search Space	Optimal Value
colsample_bytree	0.5, 0.7, 0.8, 1	0.5
gamma	0, 2, 5, 7, 10, 12, 15	0
learning_rate	0.001, 0.005, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4	0.005
max_depth	1, 2, 4, 8, 16, 32, 64, 128	64
min_child_weight	1, 2, 4, 8	4
n_estimators	25, 50, 100, 200, 400	50
subsample	0.5, 0.7, 0.8, 1	1

7.3.3. Results

Results are reported in Figs. 14–19. The two leftmost panes show again the best individual methods (1st pane) and the best data fusion approaches (2nd pane). The accuracy of the twelve machine learning approaches is presented next (3rd pane), as well as two variants discussed later (4th pane). Differently from previous charts, those in this figure do not show the data point for the TREC3 dataset, the reason being that the machine learning techniques need at least a collection to be used as a training set. Thus the boxplots in the first two panes are slightly different from those presented in the previous figures, since there is one point less.

We can draw several conclusions. Looking at the median values, SVM-E is consistently the most effective machine learning approach; SVM-E is never worse than the best data fusion technique. When using AP ρ and r_s (as well as for τ, not shown here) as accuracy measures, SVM-E is the best possible option, as it outperforms the most effective individual methods, and the difference with NC-BC and SL is statistically significant at the.05 level (also note that data fusion methods are particularly ineffective in this case). SVM-E variation (measured as interquartile range) on AP r_s is also much smaller than the variation on the best individual methods. Finally, SVM-E is also the best possible option for AAP ρ (r_s and τ are similar). Moreover, the machine learning approaches are trained on AP values (the reason being the small amount of data available that makes it ineffective to work on MAP), whereas the individual methods are aimed at MAP prediction. This different objective reduces the effectiveness of learning algorithms in MAP prediction. While of course MAP prediction can be considered an interesting final aim, the fact that machine learning approaches outperform the best individual methods on AP is encouraging, also taking into account that the generality of the machine learning approach can allow to include same MAP tailoring as well.

The variants shown in the rightmost panes of these charts are obtained by learning on the single most similar collection (called

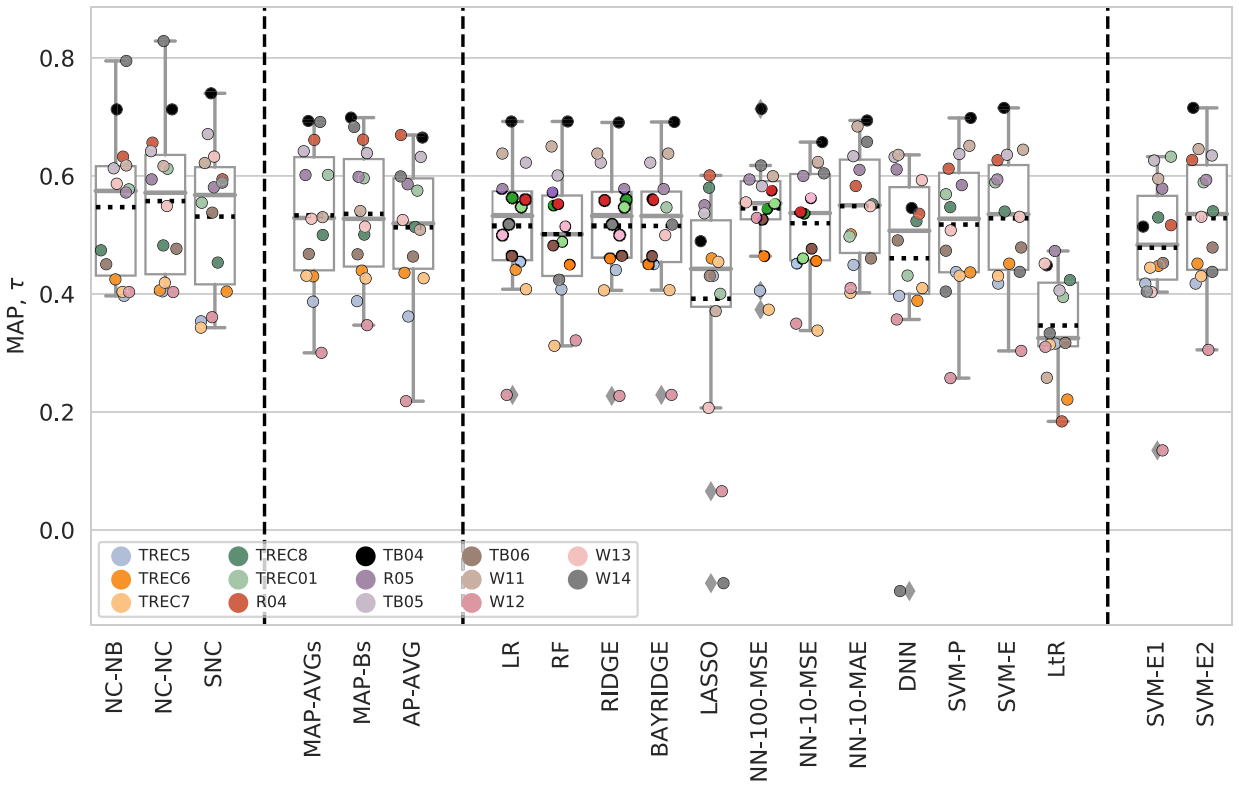


Fig. 14. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: MAP τ.

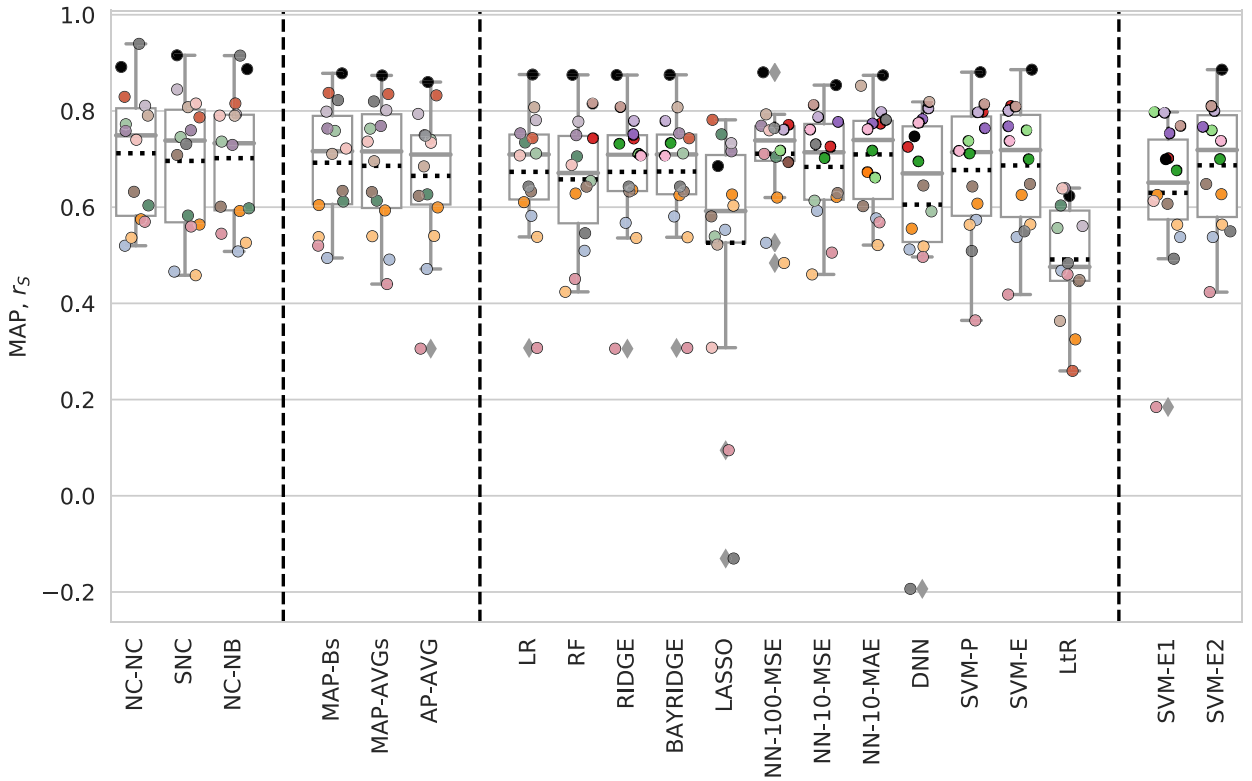


Fig. 15. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: MAP r_s.

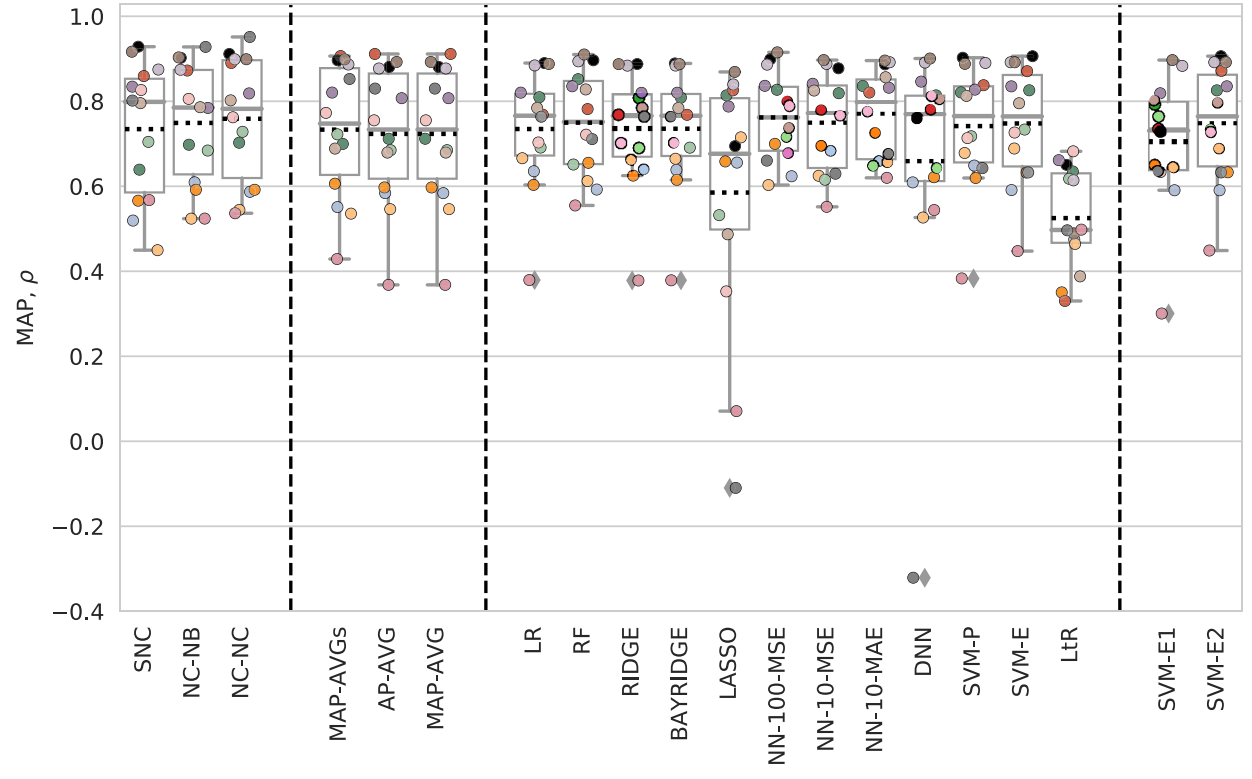


Fig. 16. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: MAP ρ.

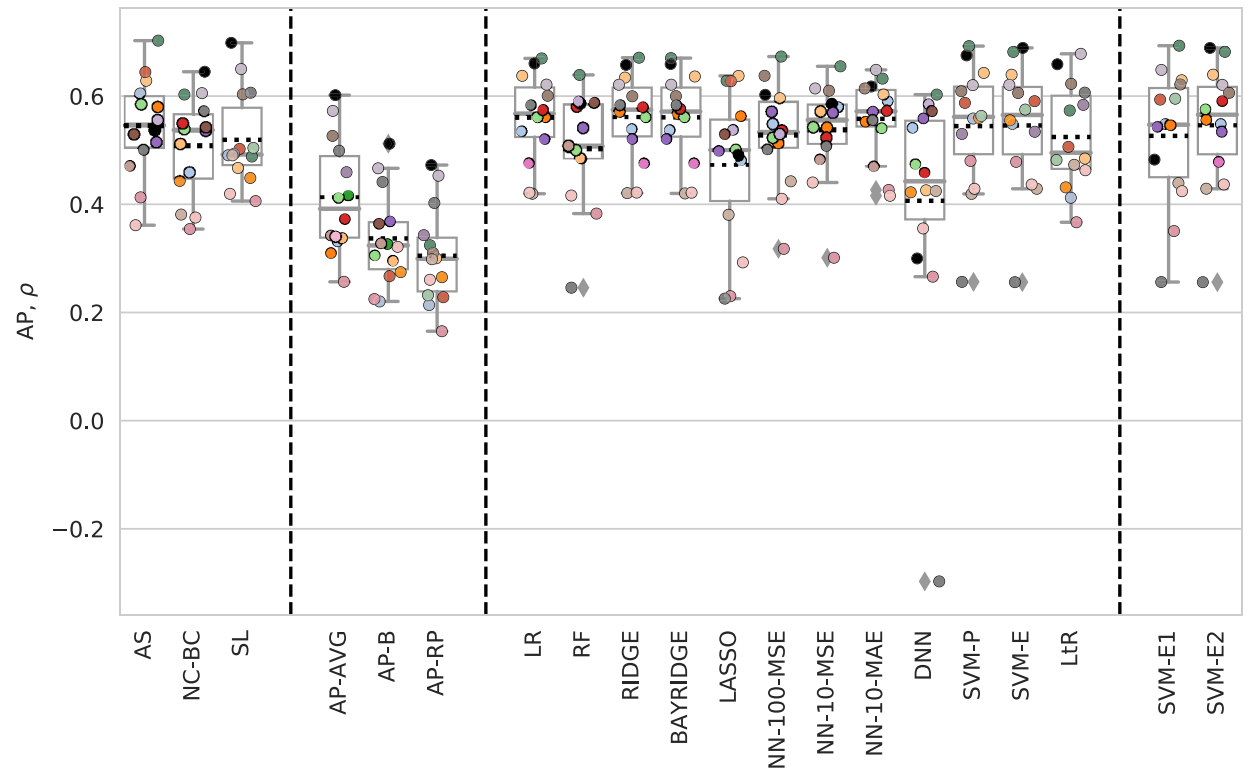


Fig. 17. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: AP ρ.

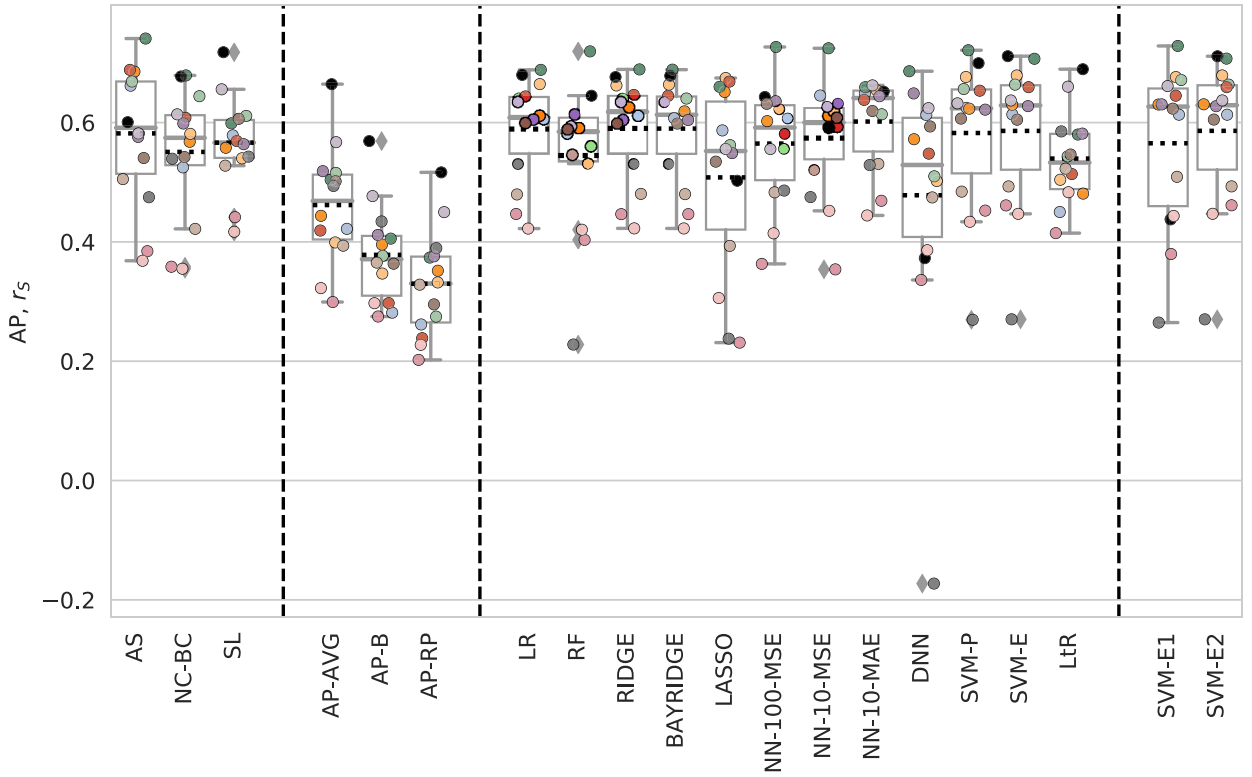


Fig. 18. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: AP r_s.

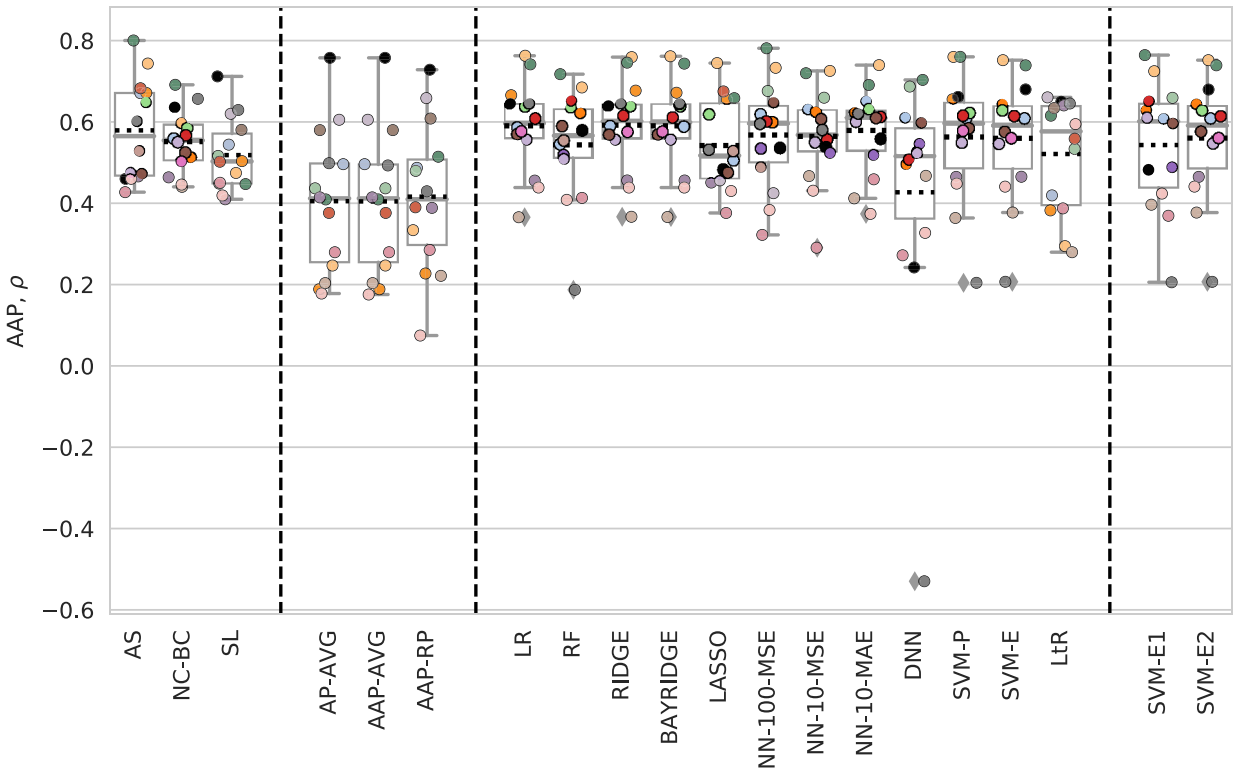


Fig. 19. Accuracy of machine learning approaches, top three individual methods, and top three data fusion approaches: AAP ρ.

SVM-E1) and on the two most similar ones (called SVM-E2). Learning on the three most similar datasets (SVM-E3, not shown) is indistinguishable from SVM-E2. As the similarity measure we use the average Kolmogorov–Smirnov distance (Massey Jr, 1951) between the distributions of AP values predicted by the individual methods. The underlying hypothesis is that the more similar the training data to the object of prediction, the smaller the training set needed, and the higher the accuracy results. Note that SVM-E1 and SVM-E2 are as effective as SVM-E. Moreover, SVM-E1 is more efficient than SVM-E as one does not need to train the regression SVM on many collections but can select just the most similar ones, thus decreasing computation times. Nevertheless, variation is usually larger on SVM-E2 than SVM-E: training on more datasets allows learning a more stable model.

However, a more careful inspection of the charts reveals that both the data fusion and the machine learning approaches perform particularly badly on specific datasets, namely the Web track collections. This is even more manifest when looking at the AP box-plots, where W11, W12, and W13 are consistently among those with lowest accuracy. These collections, as remarked in Section 4.3, feature non-binary relevance judgments: it might be that the binarization that we performed to compute AP introduced too much noise. We therefore performed the same analysis focusing on non-Web collections only, i.e., those with binary relevance. Results show (not reported here) that the top three individual methods are slightly different from Figs. 14 and 16, whereas the top three data fusion approaches do not change. When accuracy is measured with τ, as well as r_s (not shown), SVM-E approach shows the same accuracy of the top individual methods.

To conclude, we make two final remarks. When evaluating ML results, the well-known *cross-validation* technique is often used: A dataset is split into complementary subsets, and the tested machine learning approach is learnt and evaluated multiple times using different partitions (Kohavi, 1995). However, due to the intrinsic definition of our problem, we cannot rely on such a technique. In fact, we cannot do cross-validation using each collection as a whole, since we select, as the test data, a collection that has been released on a specific year and then we use as training data all the collections that have been released over the previous years (see also Section 7.3.1). In other words, given a specific year we can only treat the collection of that year as testing set and past collections as training set. Furthermore, we cannot perform cross-validation by selecting/removing some individual AP scores (i.e., ⟨system, topic⟩ pairs) from the training and test set since, in order to test the effectiveness of our ML setting, we need all the AP scores for a given collection. Thus, we can only treat a collection as a monolithic item, which can not be split into sub parts.

The second remark is that a natural extension of this work would be to provide techniques and guidelines on which combination approach is the most effective giving some particular characteristics of a dataset. We performed some preliminary analysis and tried to find patterns and correlations between the AP / MAP / AAP scores of a given collection and some of its most intuitive and straightforward features, such as the number of systems, the number of topics, the average scores of systems and topics, and so on. However, we failed to find any of such correlations. We believe that a sound and complete analysis of the correlation between the collection features and the scores would require another paper to be investigated properly; thus, we leave such an analysis for future work.

7.3.4. Transfer learning

In this setting, Transfer Learning (TL) seems a natural and promising direction to explore. In fact, TL is used when training and test data are not drawn from the same feature space and/or do not have the same distribution. Indeed, when the distribution changes, the results of a predictive learner can be degraded. Our research task falls in this context. In fact, samples of the past collections (training data) and those of the current collection (test data) are collected under different conditions, thus have different distribution. Moreover, TL has been proven to be an effective methodology in a somehow related setting: The vertical selection for web search (Arguello, Diaz, & Paiement, 2010).

Thus, as a final result of this paper we attempt to study this idea and we report some results on six datasets: TREC3, TREC5, TREC6, TREC7, TREC8, and TREC01.

We try TL on M5P, RF, SVM-P, SVM-E. We investigate a specific transfer learning algorithm called "Maximum Independence Domain Adaptation" (MIDA) (Yan, Kou, & Zhang, 2018), which achieves state-of-the-art results in several contexts. We learn a model on a single dataset only, and transfer it to another one, for two different reasons: First, the aggregation of multiple train collections into a sort of big training collection is not trivial, and might be wrong in our TL setting; the aim of TL algorithms is to transfer knowledge between different models/dataset, leveraging their differences; thus the fusion of different models/dataset should be avoided. Second, all TL algorithms, including MIDA, present a high computational complexity, and algorithm convergence issues, that prevent them to run on a large amount of data.

We did some experiments with the TL algorithm TCA (Pan, Tsang, Kwok, & Yang, 2011), but results where almost indistinguishable from the ones obtained with MIDA. We leave to future work experiments on learning on more than one dataset, and on using different TL algorithms.

Fig. 20 compares TL to the classical learning methods on the six datasets. The comparison is for MAP (τ) only, as the other measures show a similar behaviour and thus are not reported here. The charts show pairs of box-plots (one pair per panel): for each pair, the box-plot on the left shows some of the classical non-TL methods reported in previous figures, but when training on a single dataset; we report RF, SVM-P, and SVM-E: we include the former to use a tree based method, and the SVM variants because are the most effective in the non TL scenario. The box-plot on the right of each panel is the corresponding TL. Perhaps surprisingly, in general

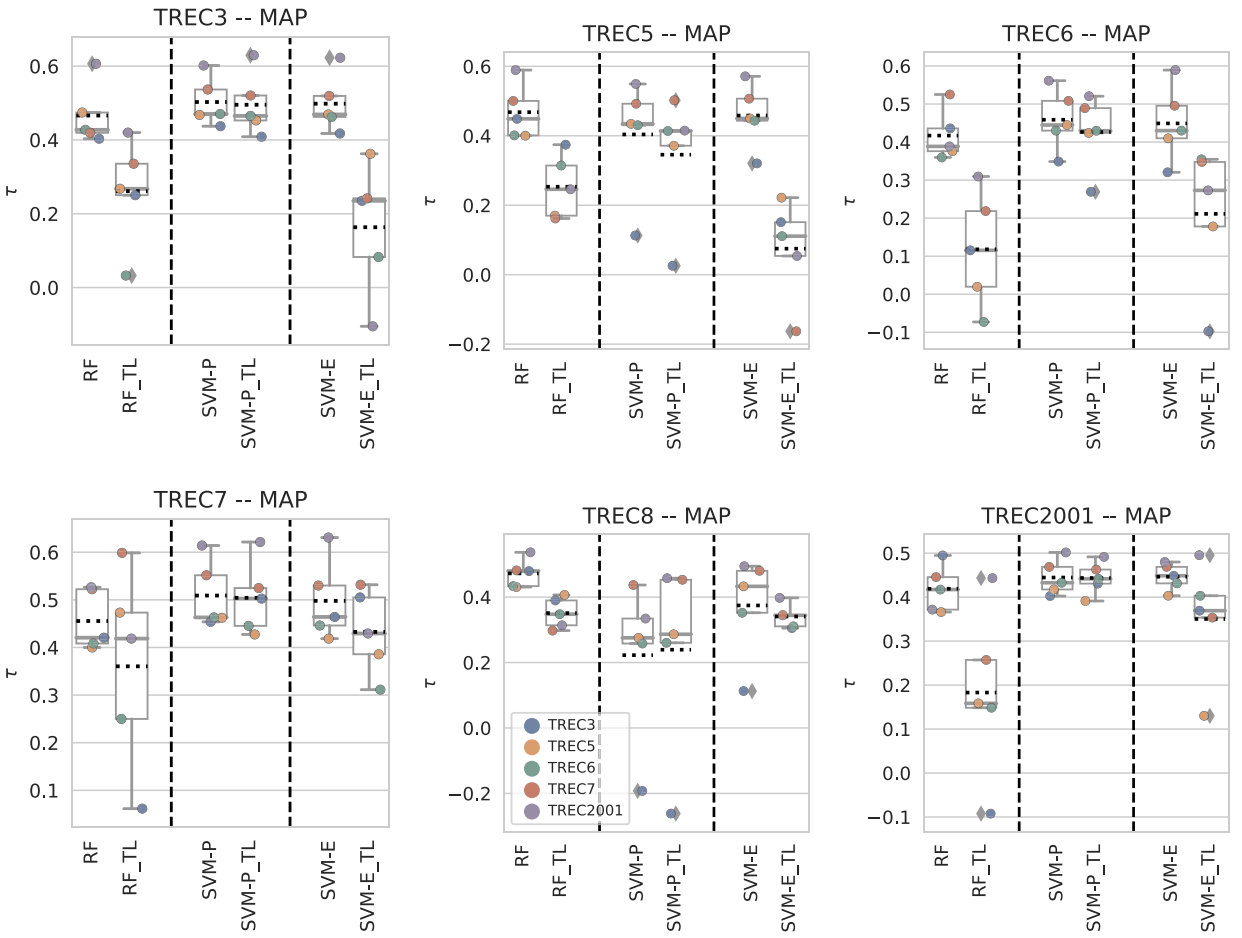


Fig. 20. Accuracy of transfer learning approaches: MAP τ.

TL is not effective; rather, it systematically and significantly harms RF, and SVM-E. The only case in which TL is competitive with the non-TL counterpart is SVM-P.

8. Conclusions and future work

We presented the results of a battery of experiments and of a rather extensive analysis over 17 prediction methods, 14 TREC collections, 15 accuracy measures (obtained by combining the three MAP, AP, and AAP with the five ρ, τ, r_s, δ, and τ_{ap}), four data fusion approaches (plus variants), and twelve machine learning algorithms (plus variants). We have provided a fourfold contribution: (i) Figs. 2–9 are a solid account of individual method effectiveness across different collections; (ii) the analysis of Section 6 highlights some interesting, potentially useful, and so far unnoticed relationships between the individual methods; (iii) the negative results on the two promising techniques of data fusion and transfer learning techniques, although not useful in practice, will avoid other researchers to perform the same attempts; and (iv) the results on method combinations by means of machine learning algorithms provide a practical methodology for the researcher that wants to run an effectiveness evaluation without human relevance assessments. Overall, our results show that the combination of the methods for effectiveness evaluation without relevance assessments is a viable approach, is effective and robust when using off-the-shelf, state-of-the-art machine learning algorithms, and provides a useful framework for future improvements. In particular, despite being sometimes outperformed by the best single method, the combination of the methods for evaluation without relevance assessments via machine learning is more effective than a random selection of the individual methods, and less risky in the real case scenario, where neither the knowledge on the performance of the individual methods nor the performance of the participating runs is known a-priori.

This research leaves plenty of space for future work. We will repeat the same analyses using other effectiveness metrics besides AP (and MAP). This issue is particularly critical for the most recent collections that feature non-binary relevance. We have used only the individual prediction methods based on systems outcomes; Diaz's method (Diaz, 2007), that requires the document collection as well, is an obvious candidate to be added. We have not focused yet on the computational complexity and time needed to learn a model on the basis of the past datasets available, and we plan to do so. Roughly, the training phase for learning a model even on all the past datasets is a matter of a few hours, and once the model has been learned its application on a new test set is very fast (a few seconds). The machine learning approach suggests a more general framework that could include other features, also derived from completely different methods (for example, analyzing the text of topic descriptions and/or documents; properties of the systems; and so on). This seems a promising approach, and we intend to pursue this research direction in the future. It would also be a way to address some limitations of the individual methods, that are quite rigid and difficult to extend.

Transfer learning could be exploited to adapt the models learned on past datasets to a new one with different properties (Li, Sanderson, Carman, & Scholer, 2016). On more technical issues, in our approach we learn AP, not MAP, since as already discussed we do not have enough data to build a regressor on MAP values. This might be one reason for the better accuracy on AP (as well as AAP), than on MAP. Thus, we might refine our learning system to take into account MAP to some extent. Also, most individual methods generally aim at predicting MAP: it might be possible to tailor them as well for more accurate predictions of AP and AAP. Furthermore, we plan to test more sophisticate data fusion and ML techniques: we plan to adapt to the setting of query performance prediction the learning-to-rank approach proposed by Raiber and Kurland (2014), as well as data fusion techniques (Jayasinghe, Webber, Sanderson, & Culpepper, 2014; Shtok, Kurland, & Carmel, 2016).

Acknowledgments

We would like to acknowledge: Alex Falcon for some improvements on Neural Network, Stefano Passador for some experimentation on Learning to Rank Baselines, Marco Basaldella for the stimulating discussion on Neural Networks, and the anonymous reviewers for their useful comments that helped to improve the overall quality of the paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ipm.2019.102149 .

References

Arguello, J., Diaz, F., & Paiement, J.-F. (2010). *Vertical selection in the presence of unlabeled verticals. Proceedings of the 33rd ACM SIGIR*, 691–698. https://doi.org/10.1145/1835449.1835564.

Aslam, J. A., & Savell, R. (2003). *On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. Proceedings of the 26th ACM SIGIR*, 361–362.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Diaz, F. (2007). *Performance prediction using spatial autocorrelation. Proceedings of the 30th ACM SIGIR*, 583–590.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association, 56*(293), 52–64.

Emerson, P. (2013). The original Borda count and partial voting. *Social Choice and Welfare, 40*(2), 353–358. https://doi.org/10.1007/s00355-011-0603-9.

Fanelli, D. (2012). Negative results are disappearing from most disciplines and countries. *Scientometrics, 90*(3), 891–904. https://doi.org/10.1007/s11192-011-0494-7.

Ferro, N. (2017). Reproducibility challenges in information retrieval evaluation. *Journal of Data and Information Quality, 8*(2) 8–1

Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., & Zobel, J. (2016). Increasing reproducibility in IR: findings from the Dagstuhl seminar on reproducibility of data-oriented experiments in e-science. *ACM SIGIR Forum, 50*(1), 69–82.

Fishburn, P. C. (1977). Condorcet social choice functions. *SIAM Journal on Applied Mathematics, 33*(3), 469–489.

Hauff, C., Hiemstra, D., Azzopardi, L., & de Jong, F. (2010). *A Case for Automatic System Evaluation. Proceedings of ECIRLNCS5993. Proceedings of ECIR*, 153–165.

Hauff, C., Hiemstra, D., & de Jong, F. (2008). *A survey of pre-retrieval query performance predictors. Proceedings of the 17th ACM CIKM*, 1419–1420.
Hauff, C., & de Jong, F. (2010). *Retrieval system evaluation: Automatic evaluation versus incomplete judgments. Proceedings of 33rd ACM SIGIR*, 863–864.
Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67.
Jayasinghe, G. K., Webber, W., Sanderson, M., & Culpepper, J. S. (2014). *Improving test collection pools with Machine Learning. Proceedings of the 2014 ADCS*, 2:2–2:9.
Knight, J. (2003). Null and void. *Nature, 422*(6932), 554–555. https://doi.org/10.1038/422554a.
Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the IJCAI*, 1137–1143.
Li, P., Sanderson, M., Carman, M., & Scholer, F. (2016). *On the effectiveness of query weighting for adapting rank learners to new unlabelled collections. Proceedings of the 25th ACM CIKM*, 1413–1422.
Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundation and Trends of Information Retrieval, 3*(3), 225–331. https://doi.org/10.1561/1500000016.
Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association, 46*(253), 68–78.
Mizzaro, S., Mothe, J., Roitero, K., & Ullah, M. Z. (2018). *Query performance prediction and effectiveness evaluation without relevance judgments: Two sides of the same coin. Proceedings of the 41st ACM SIGIR*, 1233–1236. https://doi.org/10.1145/3209978.3210146.
Mizzaro, S., & Robertson, S. (2007). *HITS Hits TREC: Exploring IR evaluation results with network analysis. Proceedings of the 30th ACM SIGIR*, 479–486.
Nuray, R., & Can, F. (2003). *Automatic ranking of retrieval systems in imperfect environments. Proceedings of the 26th ACM SIGIR*, 379–380.
Nuray, R., & Can, F. (2006). Automatic ranking of information retrieval systems using data fusion. *Information Processing & Management, 42*(3), 595–614.
Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks, 22*(2), 199–210. https://doi.org/10.1109/TNN.2010.2091281.
Park, T., & Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association, 103*(482), 681–686.
Raiber, F., & Kurland, O. (2014). *Query-performance prediction: Setting the expectations straight. Proceedings of the 37th ACM SIGIR*, 13–22.
Roitero, K., Passon, M., Serra, G., & Mizzaro, S. (Passon, Serra, Mizzaro, 2018a). Reproduce. generalize. extend. on information retrieval evaluation without relevance judgments. *Journal of Data and Information Quality, 10*(3), 11:1–11:32. https://doi.org/10.1145/3241064.
Roitero, K., Soprano, M., Brunello, A., & Mizzaro, S. (Soprano, Brunello, Mizzaro, 2018b). Reproduce and improve: an evolutionary approach to select a few good topics for information retrieval evaluation. *Journal of Data and Information Quality, 10*(3), 12:1–12:21. https://doi.org/10.1145/3239573.
Roitero, K., Soprano, M., & Mizzaro, S. (Soprano, Mizzaro, 2018c). *Effectiveness evaluation with a subset of topics: A practical approach. Proceedings of the 41st ACM SIGIR*, 1145–1148. https://doi.org/10.1145/3209978.3210108.
Sakai, T., & Lin, C.-Y. (2010). *Ranking Retrieval Systems without Relevance Assessments — Revisited. Proceeding of 3rd EVIA — A satellite workshop of NTCIR-8*. Tokyo, Japan: National Institute of Informatics, 25–33.
Shtok, A., Kurland, O., & Carmel, D. (2016). Query performance prediction using reference lists. *ACM Transactions on Information Systems (TOIS), 34*(4), 19:1–19:34.
Soboroff, I., Nicholas, C., & Cahan, P. (2001). *Ranking retrieval systems without relevance judgments. Proceedings of the 24th ACM SIGIR*, 66–73.
Spoerri, A. (2005). How the overlap between the search results of different retrieval systems correlates with document relevance. *Proceedings of the American Society for Information Science and Technology, 42*(1).
Spoerri, A. (2007). Using the structure of overlap between search results to rank retrieval systems without relevance judgments. *Information Provessing & Management. 43*(4), 1059–1070.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
Voorhees, E. M. (2003). *Overview of the TREC 2003 robust retrieval track. Trec*, 69–77.
Wilcoxon, F., Katti, S., & Wilcox, R. A. (1970). Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected Tables in Mathematical Statistics, 1*, 171–259.
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.
Wu, S., & Crestani, F. (2003). *Methods for ranking information retrieval systems without relevance judgments. Proceedings of the 2003 ACM symposium on applied computing*, 811–816.
Yan, K., Kou, L., & Zhang, D. (2018). Learning domain-invariant subspace using domain features and independence maximization. *IEEE Transactions on Cybernetics, 48*(1), 288–299. https://doi.org/10.1109/TCYB.2016.2633306.
Yilmaz, E., Aslam, J. A., & Robertson, S. (2008). *A new rank correlation coefficient for information retrieval. Proceedings of the 31st ACM SIGIR*, 587–594.