Information Processing and Management 60 (2023) 103332

Contents lists available at ScienceDirect

Information Processing and Management

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

An in-depth investigation on the behavior of measures to quantify reproducibility

Maria Maistro [a,*], Timo Breuer [b], Philipp Schaer [b], Nicola Ferro [c]

[a] University of Copenhagen, Denmark
[b] TH Köln - University of Applied Sciences, Germany
[c] University of Padua, Italy

ARTICLE INFO

Dataset link: https://github.com/irgroup/ipm-reproducibility

Keywords:
Reproducibility
Information retrieval
Evaluation

ABSTRACT

Science is facing a so-called reproducibility crisis, where researchers struggle to repeat experiments and to get the same or comparable results. This represents a fundamental problem in any scientific discipline because reproducibility lies at the very basis of the scientific method. A central methodological question is how to measure reproducibility and interpret different measures. In Information Retrieval (IR), current practices to measure reproducibility rely mainly on comparing averaged scores. If the reproduced score is close enough to the original one, the reproducibility experiment is deemed successful, although the identical scores can still rely on entirely different result lists. Therefore, this paper focuses on measures to quantify reproducibility in IR and their behavior. We present a critical analysis of IR reproducibility measures by synthetically generating runs in a controlled experimental setting, which allows us to control the amount of reproducibility error. These synthetic runs are generated by a deterioration algorithm based on swaps and replacements of documents in ranked lists. We investigate the behavior of different reproducibility measures with these synthetic runs in three different scenarios. Moreover, we propose a normalized version of Root Mean Square Error (RMSE) to quantify reproducibility better. Experimental results show that a single score is not enough to decide whether an experiment is successfully reproduced because such a score depends on the type of effectiveness measure and the performance of the original run. This study highlights how challenging it can be to reproduce experimental results and quantify the amount of reproducibility.

1. Introduction

Researchers in all areas of science are confronted with the so-called replication or reproducibility crisis that became apparent in more and more disciplines during the last years (Open Science Collaboration, 2015). While initially being discussed in psychological science (Pashler & Wagenmakers, 2012), it quickly became clear that researchers continuously fail to reproduce previous experiments and findings, regardless of being their own or foreign work. As reported by Baker (2016), approximately 70% of researchers in physics and engineering are not able to reproduce someone else's experiments, and roughly 50% fail to reproduce even their own experiments. Computational and data-intensive sciences (Freire, Fuhr, & Rauber, 2016; National Academies of Sciences, Engineering, and Medicine, 2019) are no exception, as shown in Artificial Intelligence (AI) and Machine Learning (ML) (Gibney, 2020) and most recently in IR (Breuer et al., 2020).

* Corresponding author.
E-mail addresses: mm@di.ku.dk (M. Maistro), timo.breuer@th-koeln.de (T. Breuer), philipp.schaer@th-koeln.de (P. Schaer), ferro@dei.unipd.it (N. Ferro).

https://doi.org/10.1016/j.ipm.2023.103332
Received 22 November 2022; Received in revised form 20 February 2023; Accepted 20 February 2023
Available online 14 March 2023

In IR, the well-known Cranfield paradigm (Cleverdon, 1962, 1967) has helped researchers in the last decades to live in a state of felt comfort and security as test collections allowed a relatively high level of standardization and best practice. In this context, reproducibility was taken for granted. However, reproducibility was far too long a "close enough" attitude. Researchers put reasonable effort into understanding how an approach was implemented and how an experiment was conducted. Then, after several iterations, when they obtain performance scores that somehow resemble the original ones, they decide that an experimental result is reproduced.

Obviously this approach is not enough to guarantee the reproducibility of IR evaluation settings as many different aspects of the systems and experimental settings are neglected, even more today in the current deep learning and neural era (Lin, 2018; Marchesin, Purpura, & Silvello, 2020; Yang, Lu, Yang, & Lin, 2019). To discuss the different facets of reproducibility, to create awareness within the community, and to formulate concrete guidelines and best practices, a number of events took place in recent years. In 2011 the DESIRE workshop focused on data infrastructures for managing experimental IR data (Agosti, Ferro, & Thanos, 2012). A year later a SIGIR workshop on open source IR systems (Trotman et al., 2012) discussed how to support reproducibility from the developers' perspective. Workshops on evaluation as a service (Hopfgartner et al., 2015) and reproducible baselines (Arguello, Crane, Diaz, Lin, & Trotman, 2015) followed a little later. In SIGIR 2022, a tutorial provided an introduction to reproducible experiments in IR (Lucic et al., 2022). The European Conference on Information Retrieval (ECIR) in 2015 and the International ACM Conference on Research and Development in Information Retrieval (SIGIR) in 2022 picked up these ideas and introduced a special Reproducibility Track, where the community is encouraged to submit papers that repeat, reproduce, generalize, and analyze prior work. Here a successful reproduction is not a hard requirement, but the authors are invited to provide an in-depth evaluation of the reproduction process. Next to ECIR and SIGIR, reproducibility is also considered as part of the reviewing process in other major venues.

Different models and definitions were introduced to form a general understanding and to find a common terminology while discussing reproducibility issues in IR. As a general framework the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science" introduced the Platform, Research goal, Implementation, Method, Actor, and Data (PRIMAD) model in 2016 to describe the different aspects of reproducibility. The ACM "Artifact Review and Badging" guidelines[1] introduce a consistent terminology, which is heterogeneous across disciplines. For the case of reproducibility the ACM guidelines describe "...[reproducibility] means that an independent group can obtain the same result using the author's own artifacts", but does not clarify how to compare these results and what measurement should be used. In recent years the IR community came to the conclusion that simply comparing averages of performance scores does not tell us anything about the actual degree of reproducibility of the systems at hand. If we simply compare the averages across topics with the "close enough" attitude, we disregard per-topics scores. This means that two systems with the same average score, might behave very differently on different topics. Moreover, per-topic retrieval performances are typically compared by looking at relevance assessments for a given position in the result list, without looking at the actual documents and their position itself, so two completely distinct result lists can potentially produce the same performance score (see Fig. 1).

Therefore Breuer et al. (2020) introduced different measures which allow for comparing experimental results at different levels from most specific to most general: the ranked lists of retrieved documents; the actual scores of effectiveness measures; the observed effects; and significant differences. These measures illustrate that nearly identical performance scores do not necessarily mean that the results were "bitwise" reproduced (National Academies of Sciences, Engineering, and Medicine, 2019), which in IR would mean reproducing the same ranked lists of results. With the help of the `repro_eval` toolkit (Breuer, Ferro, Maistro, & Schaer, 2021), the proposed measures can be computed for any given IR experiment. While this is a huge step towards providing a robust environment and setting for reproducible IR experiments, applying and interpreting these measures remains an open question. Indeed, the fundamental methodological question is how to measure reproducibility and provide guidelines on reading and interpreting the different measures.

In this paper, we investigate these issues by considering a set of controlled experiments based on artificially deteriorated runs to find relations between the measure score and the amount of detriment in each experiment. Our aim is to discover reproducibility patterns and achieve a better understanding of how to interpret measures which quantify reproducibility. While in our previous work we investigated both reproducibility (different team, same dataset) and replicability (different team, different dataset) setting, this study's approach – in terms of the ACM guidelines – is only applicable for reproducibility experiments. By focusing on reproducibility only we gain a fixed evaluation environment that allow this research to investigate factors that affect reproducibility of IR experiments, how to measure, and to provide insights on reproducibility measures and their behavior.

The main contributions of this paper are: (1) a critical analysis of IR measures to quantify reproducibility: to the best of our knowledge this is the first study that investigates IR reproducibility with simulated runs; (2) a deterioration algorithm to simulate reproducibility runs: this algorithm is based on 2 operations, i.e., replacing and swapping documents in a ranked list, and allows to control the amount of reproducibility error and to generate different deterioration archetypes; (3) a normalization procedure for RMSE: this mitigates the variability in RMSE score due to the run type and always returns scores in $[0, 1]$, which are easier to interpret.

We conduct experiments with 3 different scenarios and more than 3 millions simulated reproducibility runs (source code publicly available.[2]) Experiments show that quantifying and interpreting reproducibility results is not trivial because reproducibility scores depend on different factors, as the type of reproducibility run under investigation and the measure used to quantify reproducibility,

[1] https://www.acm.org/publications/policies/artifact-review-and-badging-current

[2] https://github.com/irgroup/ipm-reproducibility

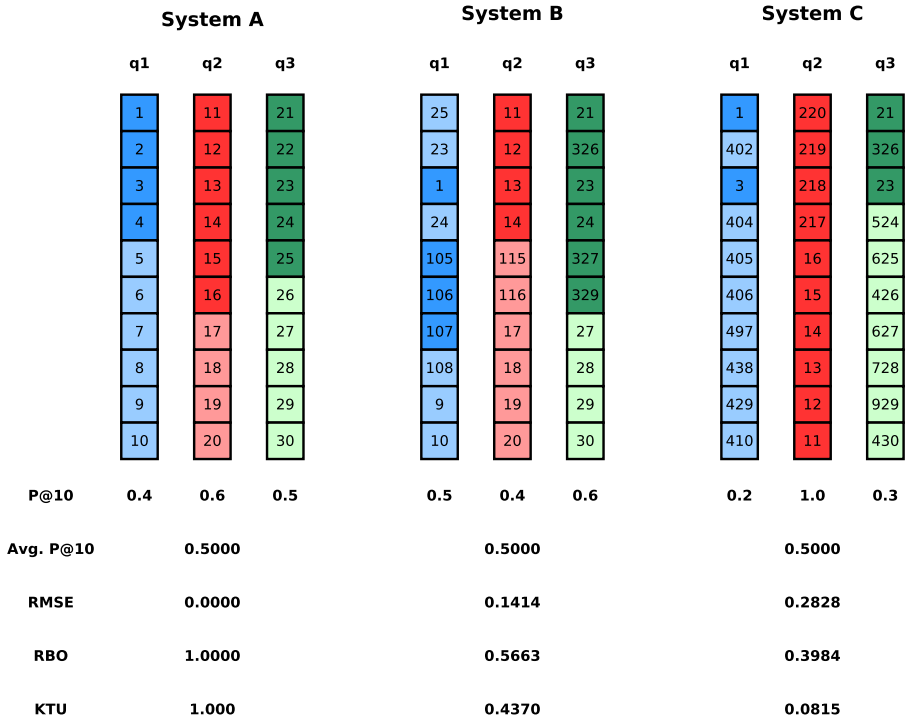


Fig. 1. Toy example with rankings that result in the same P@10 but different reproducibility scores. The numbers in the ranking list for each query represent the document identifiers. Ranking items with a darker color represent relevant documents.

together with its underlying effectiveness measure. This highlights once more that simple comparisons of average scores, the "close enough" attitude, is not enough to determine whether an experiment is successfully reproduced or not.

The remainder of this paper is organized as follows: Section 2 presents related work, Section 3 revises IR reproducibility measures and describes our approach to deteriorate runs; Section 4 reports on the experimental results, and Section 5 presents our conclusions and directions for future work.

2. Related work

As recently analyzed, the overall scientific progress driven by computational experiments is limited and reproducibility issues together with weak baselines are often seen as a reason for stagnation (Dacrema, Boglio, Cremonesi, & Jannach, 2021; Dacrema, Cremonesi, & Jannach, 2019; Lv et al., 2021). Yang et al. (2019) re-evaluated this circumstance and confirmed that it is still true for IR research almost ten years after it had already been pointed out by Armstrong, Moffat, Webber, and Zobel (2009). With special regards to the computational sciences, Ivie and Thain (2018) argue that non-reproducible results "stem from a need to simultaneously satisfy the needs of both the computer and a human". According to them, compromises between the concrete operations of a computer and the human reasoning about the experiment on a more abstract level have to be made. These compromises such as software, workflows, and statistics can introduce barriers to reproducibility.

While there is scientific discourse on how reproducibility and replicability are defined and how the terminology should be applied to the scientific experiments (De Roure, 2014; Plesser, 2017), we follow the definitions by the updated ACM Policy on Artifact and Review Badging. The definition of reproducibility aligns with the International Vocabulary for Metrology (VIM). More specifically, *reproducibility* describes the validation of results by a different team using the same experimental setup. Analogously, we consider an experiment to be reproduced if the results are validated with the same test collection.

Recently, the topic of reproducible research came into focus of the IR community (Ferro, 2017). In the following, we review initiatives and attempts towards reproducible IR experiments. Overall, attempts were made on a conceptual level in the form of workshops, conference tracks, technical platforms and tools, as well as by the explicit review of experimental resources and artifacts.

Conceptually, the *Platform, Research goal, Implementation, Method, Actor, and Data (PRIMAD)* model (Ferro et al., 2016) focuses on defining which experimental components are kept the same and which one are changed in a reproducibility study and, especially, on what is gained and understood in reproducing or failing to reproduce a study.

Since 2015, ECIR hosts a dedicated reproducibility track for analyzing the extent to which previous studies are valid and reproducible. SIGIR introduced its reproducibility track in 2022. Likewise, there have been workshops like RIGOR (Arguello et al., 2015) that analyze reproducibility, inexplicability, and generalizability with a special focus on open-source software, or

OSIRRC (Clancy, Ferro, Hauff, Sakai, & Wu, 2019) that evaluates reproducible retrieval systems packaged with a technical framework based on Docker, or CENTRE (Ferro, Fuhr, Maistro, Sakai, & Soboroff, 2019a; Sakai et al., 2019; Soboroff, Ferro, Maistro, & Sakai, 2019) that validates previous experiments at CLEF, NTCIR, and TREC, and a tutorial on best practices to ease reproducibility of IR research (Lucic et al., 2022).

Recently, several open-source retrieval toolkits were introduced that facilitate the implementation of commonly used (keyword-based) retrieval methods. Anserini builds up on the Lucene library and offers a variety of regression guides for experiments with frequently used test collections (Yang, Fang, & Lin, 2018). Likewise, the Terrier toolkit implements common retrieval methods. Both toolkits have Python bindings (Lin et al., 2021; Macdonald & Tonellotto, 2020) that not only make it possible to implement the experiment by a scripting language but also offer interfaces to state-of-the-art machine/deep learning libraries. Even more, the Python toolkit `ir_datasets` (MacAvaney et al., 2021) offers an interface to a catalog that comprises ad-hoc test collections and related resources like topic, relevance judgments, and benchmarks.

Ferro and Kelly (2018) surveyed the SIGIR community about what reproducibility is in system-oriented and user-oriented IR, and more recently, the SIGIR community enforced artifact evaluations as part of the ACM SIGIR Artifact Review and Badging.[3] The goal is to account for experimental setups, i.e., software and data, that are made available to the community and allow reproductions and replications of the results presented in the original publication.

Besides curated test collections, the IR community promotes reproducibility by archiving experimental data from evaluation campaigns at TREC (Voorhees, Rajput, & Soboroff, 2016) or CLEF (Agosti, Di Nunzio, Ferro, & Silvello, 2019). Thus, the results of previous experiments, i.e., system runs, can be used as points of reference to which we compare our reimplementations. When it comes to software archival, Potthast, Gollub, Wiegmann, and Stein (2019) implemented the TIRA Integrated Research Architecture (TIRA), which has been supporting several shared task events since 2012 (Gollub, Stein & Burrows, 2012; Gollub, Stein, Burrows & Hoppe, 2012). TIRA handles software submissions with a cluster to store and host virtual machines. This allows not only storing software permanently but also executing it at any later time. Based on public code repositories and a Docker container environment, the STELLA framework (Breuer & Schaer, 2021; Schaer et al., 2021) offered a distributed approach to deploy reproducible software artefacts for IR experiments.

Nonetheless, there are few methods to measure the extent of reproducibility. As an answer, reproducibility measures for the system-oriented IR experiments were introduced (Breuer et al., 2020). The framework comprises a set of different measures that quantify reproducibility (and replicability) with different levels of specificity ranging from fine-grained comparisons of document rankings to more general comparisons of topic score distributions. The introduced reproducibility measures were validated with the help of reimplemented results in reference to the original systems runs. Since no dedicated reproducibility dataset was available, the retrieval method was altered in a principled way to simulate a researcher's attempt to reimplement a system with different parameters and configurations. While it is possible to control the overall retrieval performance, it is not always intuitive how specific parameters might influence the final document ranking. Since the reimplementations are analyzed by their final outputs, i.e., the system runs, different reimplementations could likewise be simulated by modifying the final ranking only to have even more control of how the ranking is modified. Similar simulated outputs were recently exploited by Parapar and Radlinski (2021) as systematic perturbations of an ideal ranking to evaluate novel metrics for Recommender Systems research.

Score distribution has been studied since the early days of IR (Arampatzis & Robertson, 2011; Robertson, 2001; Swets, 1963) and applied to various tasks such as, for example, rank fusion (Manmatha, Rath, & Feng, 2001), thresholds in ranking and filtering (Arampatzis & van Hameren, 2001; Arampatzis & Kamps, 2009), studying the interaction between topics and systems (Robertson & Kanoulas, 2012), and query performance prediction (Cummins, 2014). Recently, Parapar, Losada, Presedo Quindimil, and Barreiro (2020) relied on score distribution to simulate perturbed runs. This approach assumes that relevant and non relevant documents follow different score distributions. The score of an IR system is then simulated with a mixture of 2 distributions, in Parapar et al. (2020) these are 2 log-normal distributions that are fitted with observed scores from IR systems. We did not follow this approach to generate our simulated runs, but we propose a different algorithm (see Section 3.2), which allows to generate deteriorated runs in a principled way, by swapping or replacing relevant documents in a ranked result list. Theoretical results by Ferrante, Ferro, and Maistro (2015) ensure that, for example, a positive swap does not decrease the measure score. In this way, we can generate decreasing perturbed runs and keep full control of what modifications in the ranked list produced that decrease. This latter aspect is important to understand and explain the behavior of perturbed runs and to relate patterns to specific modifications in the ranked result list; it is also matters when it comes to quantify the amount of reproducibility when using measures that consider the ranked result list, as those by Breuer et al. (2020). On the other side, approaches based on score distributions directly generate modified scores, without giving the possibility to link them back to modifications in the ranked result list. Moreover, they have a stochastic component, therefore it is not possible to determine to what extent the variations in measure scores are due to the actual perturbations and to some random effects.

3. Approach

Breuer et al. (2020) presented a number of measures for reproducibility and replicability. These measures were validated with a constellation of runs, generated by systematically changing some set of parameters, e.g., the vocabulary size, the tolerance stopping criterion, the regularization strength, etc. While this work provides an overall analysis of reproducibility and replicability evaluation

[3] http://sigir.org/general-information/acm-sigir-artifact-badging/

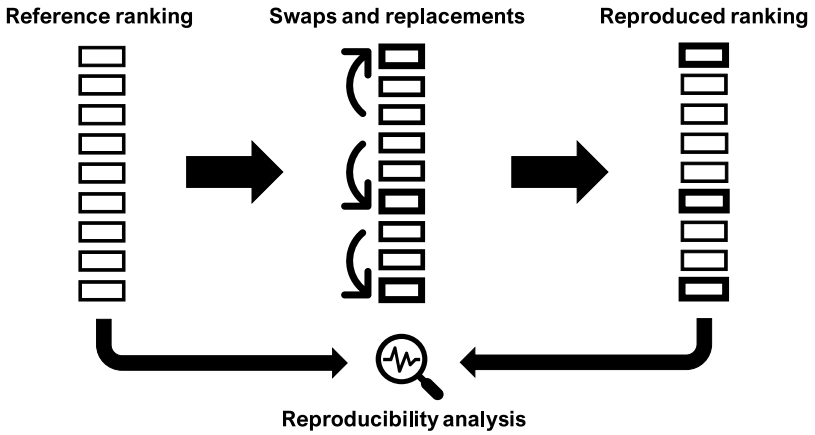


Fig. 2. The overall approach to investigate the reproducibility measures is based on the comparison of a *reproduced ranking* to a *reference ranking*. Swapping and replacing documents give us control of the deterioration in the reproduced ranking.

measures, it still remains unclear how to interpret or quantify the scores of such measures. For example, consider a reproducibility experiment where the RMSE score between the original and the reproduced run is equal to 0.05. How different are the original and reproduced runs? Can we consider the experiment as successfully reproduced?

In the following sections, we investigate these questions and dive deeper down in the problem of reproducibility experiments. Fig. 2 illustrates our overall approach. By starting with a *reference ranking*, we deteriorate the ranking by *swaps and replacements*. Swapping and replacing documents give us better control of the deterioration in the *reproduced ranking*. Both the reference and reproduced rankings are required for the *reproducibility analysis*, which we use in this study to investigate the reproducibility measures. First, we present reproducibility measures from Breuer et al. (2020) and we propose a new approach to normalize RMSE for reproducibility (Section 3.1). Then, we present our approach to generate deteriorated runs (Section 3.2). Starting from any run, we apply two basic operations, swap and replacements, to gradually modify both the order of ranked documents and the set of retrieved documents. Artificially deteriorating the runs allows us to define the amount of deterioration in each run, thus controlling how different the deteriorated run from the original run is. Finally, we describe our algorithm to generate deteriorated runs.

3.1. Reproducibility measures

In the following we describe IR reproducibility measures in details. In Breuer et al. (2020), reproducibility measures were categorized in 4 main groups: measures based on the *ordering of documents*, *effectiveness measures*, *statistical tests*, and *overall effect over a baseline*. We use the same categorization and focus on the first 3 groups. Measures that consider the overall effect over a baseline, i.e., Effect Ratio (ER) and Delta Relative Improvement (DeltaRI), are excluded because they require both a baseline and advanced run to be reproduced, thus a different deterioration algorithm. Their investigation is left for future work. To the best of our knowledge, there are no other reproducibility measures that could have been included in this work.

Let r be the original run and r' be the reproduced run. The set of topics is $\mathcal{T} = \{1, \dots, T\}$, where $t \in \mathcal{T}$ denotes a specific topic. The original ranked list of documents in response to topic t is $r_t = (d_1, \dots, d_N)$, where d_k is the document at rank position k. Similarly, r'_t is the reproduced ranked list of documents for topic t and d'_k denotes the document at rank position k. Any IR evaluation measure such as Average Precision (AP) or normalized Discounted Cumulated Gain (nDCG) (Järvelin & Kekäläinen, 2002) is denoted by M.

Ordering of documents. Kendall's τ Union (KTU) (Breuer et al., 2020; Ferro, Fuhr, Maistro, Sakai, & Soboroff, 2019b; Ferro, Maistro, Sakai, & Soboroff, 2018) and Rank-Biased Overlap (RBO) (Webber, Moffat, & Zobel, 2010) compare the actual rankings of documents in the original and reproduced runs. Since Kendall's τ is not defined for rankings that contain different sets of documents, we compute KTU instead. For each topic, KTU computes $r_t \cup r'_t$, the union of the original and reproduced rankings, by removing duplicate documents. Then, from the original and reproduced rankings, we create a list of rank positions, l_t and l'_t, where the item at position k in l_t is the rank position of d_k in $r_t \cup r'_t$. Similarly, the item at position k in l'_t is the rank position of d'_k in $r_t \cup r'_t$. Finally, we compute Kendall's τ (Kendall, 1945) between the lists of rank positions as usual:

$$\begin{aligned} \mathrm{KTU}_t(r_t, r'_t) &= \tau(l_t, l'_t) = \frac{P - Q}{\sqrt{(P+Q+U)(P+Q+V)}} \\ \mathrm{KTU}(r, r') &= \frac{1}{T}\sum_{t=1}^{T} \tau(l_t, l'_t) \end{aligned} \tag{1}$$

where P is the total number of concordant pairs (document pairs that are ranked in the same order in both rankings), Q is the total number of discordant pairs (document pairs that are ranked in the opposite order in the two rankings), U and V are the numbers

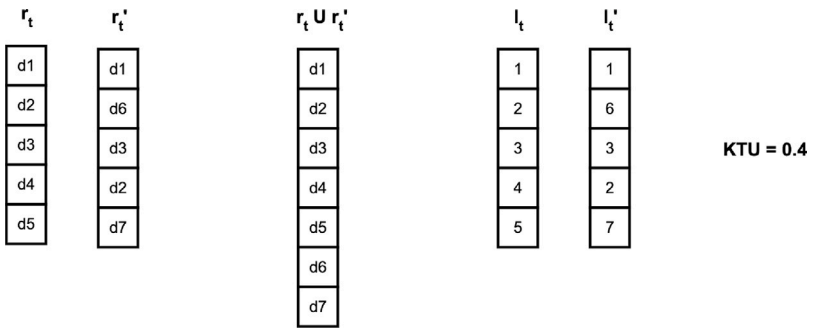


Fig. 3. Toy example of computing KTU: r_t and r'_t are the original and reproduced rankings, $r_t \cup r'_t$ is their union, and l_t and l'_t are the lists of rank positions of documents from r_t and r'_t in the union list.

of ties, in l_t and l'_t respectively. Finally, KTU(r, r') is the average computed across topics. Fig. 3 illustrates how to compute KTU for two rankings.

Differently from Kendall's τ, RBO can deal with rankings that retrieve a different set of documents. RBO assumes that r_t and r'_t are infinite rankings and is computed as follows:

$$\begin{aligned} \text{RBO}_t(r_t, r'_t) &= (1-\phi)\sum_{k=1}^{\infty}\phi^{k-1}\cdot A_k \\ \text{RBO}(r, r') &= \frac{1}{T}\sum_{t=1}^{T}\text{RBO}_t(r_t, r'_t) \end{aligned} \tag{2}$$

where $\phi \in [0, 1]$ is a parameter to account for top-heaviness: the lower ϕ the higher the weight at the top of the ranking. A_k is the overlap between r_t and r'_t, it is defined as the cardinality of the intersection of r_t and r'_t up to rank position k, divided by k. Intuitively, RBO computes the proportion of overlap between the original and reproduced rankings at each cut-off and then applies a discount based on the cut-off threshold: the higher the cut-off, the higher the overlap, so the more severe the discount.

Effectiveness measures. Root Mean Square Error (RMSE) (Breuer et al., 2020; Ferro et al., 2019b, 2018) computes the quadratic error per topic. Given an IR evaluation measure M, it is defined as follows:

$$\text{RMSE}(r, r', M) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(M(r_t) - M(r'_t))^2} \tag{3}$$

Differently from KTU and RBO, RMSE does not consider the order of documents, but just their labels: if two documents with the same label are swapped, RMSE is not affected, since the underlying IR evaluation measure M is not affected. Therefore, we can consider RMSE less sensitive or less stricter than ranking based measures.

One limitation of RMSE, as defined in Eq. (3), is the interpretability of its score. RMSE ranges in $[0, B]$, where the upper bound $B = \max_{r'}\{RMSE(r, r', M)\}$ depends on the original run r and the IR measure M. For example, assume that a reproduced run r' achieves an RMSE score of 0.15 with respect to AP. We cannot infer whether this is successfully reproduced, unless we know the score of the worst case, i.e., RMSE upper bound. If the upper bound was 0.9, then the reproduced run would be a successful experiment, since its score is reasonably far from the worst case. Conversely, if the upper bound was 0.2, then the reproduced run could possibly be improved.

We propose to normalize RMSE by its maximum score, to ease the interpretability of this measure. The maximum value of RMSE for a given run is computed as follows:

$$\text{max-RMSE}(r, M) = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\max\{M(r_t), 1 - M(r_t)\})^2} \tag{4}$$

and normalized Root Mean Square Error (nRMSE) is the ratio between RMSE score and its upper bound:

$$\text{nRMSE}(r, r', M) = \frac{\text{RMSE}(r, r', M)}{\text{max-RMSE}(r, M)} \tag{5}$$

If we consider the example above, in the first case (upper bound 0.9) nRMSE is approximated by ~ 0.17, while in the second case (upper bound 0.2) it is 0.75. This immediately informs us on the quality of the reproduced run: in the first case the reproduced run is close to the original run, in the second case more work is needed to get a better reproducibility run.

Eq. (4) holds for any IR evaluation measure M ranging in $[0, 1]$ as explained next. For a topic t, the maximum difference is achieved by considering the worst case, i.e., the reproduced ranking scoring 0 or 1:

- if $M(r_t) \geq 0.5$, then the maximum difference is achieved by the reproduced ranking scoring 0: $|M(r_t) - M(r'_t)| = M(r_t)$;
- if $M(r_t) < 0.5$, then the maximum difference is achieved by the reproduced ranking scoring 1: $|M(r_t) - M(r'_t)| = 1 - M(r_t)$.

If the measure M does not range in $[0, 1]$, e.g., Discounted Cumulated Gain (DCG), the above can be adjusted by replacing 0 and 1 with the measure lower and upper bounds, and the decision threshold 0.5 with the middle point of the measure range.

Statistical tests. Breuer et al. (2020) propose to use statistical tests as a mean to compare the original and reproduced runs. A paired two-tailed t-test is performed between the scores of the original and reproduced runs for each topic. Then, the p-value is used as an indicator of the quality of the reproducibility experiment: the smaller the p-value, the stronger the evidence that the reproduced run is different from the original run, thus the reproduced run does not succeed in its purpose.

Note that the p-value is the least sensitive measure among those listed above. Not only it is not affected by swapping documents with the same label, as RMSE based measure, but it is also not sensitive to (random) fluctuations on the distribution of scores across topics.

3.2. Run deterioration

We investigate the behavior of the reproducibility measures in Section 3.1 in a synthetic experimental setting. This allows us to control the amount of "reproducibility error", i.e., quantify how much different is the original run from the reproduced run.

We started from the typical scenario of an IR system implemented as a multi-stage retrieval pipeline: a first retrieval stage, where candidate relevant documents are selected in the complete collection, is followed by a re-ranking stage, where the goal is to push the most relevant documents at the beginning of the ranking. For instance, a cost-efficient key-word based method retrieves the top-k results that are re-ranked by more costly relevance feedback or deep learning approaches.

We identify these two stages in the retrieval pipeline with two types of operations[4]:

- *Stage 1 - Replacement*: a document in the ranking is replaced by another document outside the ranking, i.e., a not retrieved document;
- *Stage 2 - Swap*: two documents in the ranking are swapped.

When reproducing the first retrieval stage, a researcher determines the set of retrieved documents. In terms of comparison between the original and reproduced runs, this translates in the replacement of some documents of the original run with others that were not retrieved, or were retrieved below the top k. Analogously, when reproducing the second re-ranking stage, a researcher determines the order of retrieved documents. In terms of comparison between the original and reproduced runs, this translates in swapping pairs of documents in the original run.

Depending on their impact on the ranking, we categorize replacements and swaps as *positive* or *negative*. A positive replacement increases the number of relevant documents retrieved, i.e., a non-relevant document is replaced with a relevant document that was not retrieved. Conversely, negative replacements replace a relevant document with a non-relevant one. A positive swap moves a relevant document towards the top of the ranking, i.e., a relevant document at the bottom of the ranking is swapped with a non-relevant document at the top of the ranking. Conversely, negative swaps move relevant documents from the top to the bottom of the ranking.

Based on the different combinations of swaps and replacements, we define four different archetypes, which basically correspond to the four quadrants of the Cartesian plan in Fig. 4. These archetypes relate to the corresponding implementations in a multi-stage retrieval pipeline as explained in the following:

- *Archetype I*: Replacing non-relevant with relevant documents. Swapping non-relevant with relevant documents. *Example:* First stage ranker improves, as well as the second stage re-ranker.
- *Archetype II*: Replacing non-relevant with relevant documents. Swapping relevant with non-relevant documents. *Example:* First stage ranker improves, but the second stage re-ranker deteriorates.
- *Archetype III*: Replacing relevant with non-relevant documents. Swapping relevant with non-relevant documents. *Example:* First stage ranker deteriorates, as well as the second stage re-ranker.
- *Archetype IV*: Replacing relevant with non-relevant documents. Swapping non-relevant with relevant documents. *Example:* First stage ranker deteriorates, but the second stage re-ranker improves.

In Breuer et al. (2020), runs were deteriorated in a different way by modifying parameters such as the vocabulary size, the tolerance of the stopping criterion, adding/removing pre-processing steps, etc. This mimics the behavior of a researcher, who tests different values of some parameters in order to find the best combination to reproduce the original work. Even if more realistic, this approach has some limitations. First, it is not straightforward to determine how varying some parameters will affect the original run. For example, assuming that the original run was generated by keeping stopwords, it is hard to estimate a priori the impact of stopwords removal: will the reproduced run score better or worse than the original run? Moreover, it is even harder to predict the impact at the document level. How many replacements and/or swaps will occur when removing stopwords? Second, for some parameters it might be intuitive to estimate their impact in terms of evaluation scores, but this tends to represent only the case of a reproduced run which is worse than the original run (archetype III). For example, reducing the vocabulary size will almost certainly decrease the quality of the reproduced runs. Finally, in both the above situations, we cannot quantify ahead the amount of error introduced in the reproduced run.

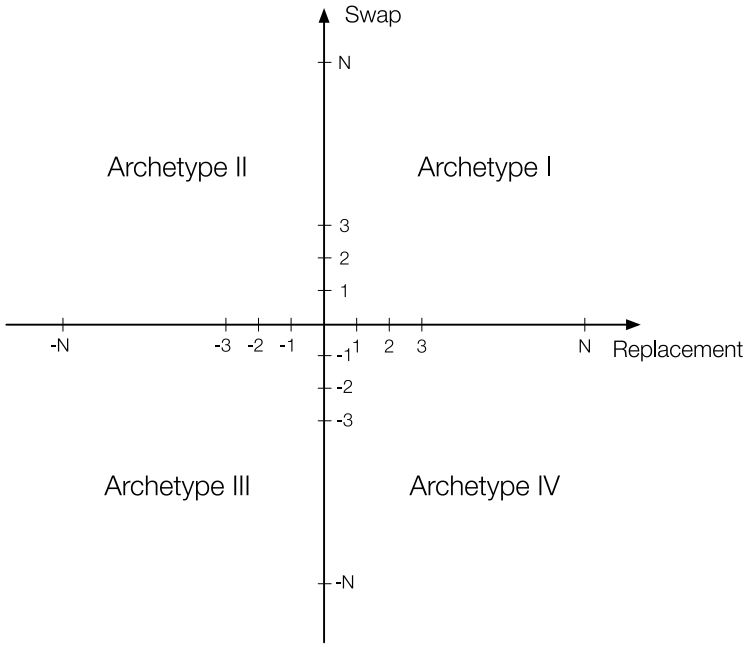


Fig. 4. Representation of the four archetypes in terms of replacements and swaps.

Algorithm to deteriorate runs. Algorithm 1 provides the pseudo-code in the case of archetype I, i.e., positive swaps and positive replacements, for a single ranked list of documents, i.e., one topic. The same algorithm can be applied to all the other archetypes with some simple adjustments by performing the appropriate checks (in Fig. 5) on the number of non-relevant and relevant documents and changing the direction of swaps and replacements. The provided source code includes the implementation of all archetypes and a variable number of topics.

The algorithm takes as input: r the original ranking which will be deteriorated; s and p, the maximum number of positive swaps and replacements to perform; $[a, b]$ and $[c, d]$, the source and destination intervals, where $a < b$, $c < d$ and $b < c$, i.e., $[a, b]$ is closer to the top of the ranking than $[c, d]$ and $[a, b] \cap [c, d] = \emptyset$. The source and destination interval are the regions of the ranking, i.e., rank positions, where we want to perform swaps and replacements. For positive swaps, the algorithm swaps non-relevant documents from the source interval with relevant documents in the destination interval. For positive replacements, the algorithm replaces non-relevant documents from the source interval with relevant not retrieved documents. Note that swaps and replacements will not interfere with each other, i.e., the rank positions where swaps and replacements are performed do not overlap. This means that a document that was swapped cannot be replaced, and a document that was replaced cannot be swapped.

The algorithm starts by checking that the number of non-relevant documents in the source interval is greater than zero (line 5). If all documents in the source interval are relevant, we cannot perform any positive swap or replacement, therefore the ranking cannot be deteriorated. In this case, we simply return the original ranking r.

If there are non-relevant documents in the source interval, the algorithm determines the actual number of swaps and replacements, i.e., the number of swaps and replacements that can be performed. The algorithm always performs the highest number of actual swaps and replacements, which is lower or equal to the inputted number of swaps and replacements. This is done in 3 steps, detailed in the following.

First, the algorithm checks that the number of swaps to perform does not exceed the number of relevant documents in the destination interval. Indeed, relevant documents in the destination interval are swapped with non-relevant documents in the source interval. Therefore, the number of actual swaps is set to the minimum between the inputted number of swaps and the number of relevant documents in the destination interval (line 6). If there are no relevant documents in the destination interval, no swaps are performed.

Second, the algorithm checks that the number of replacements does not exceed the number of relevant documents that are not retrieved. Therefore, the actual number of replacements is set to the minimum between the inputted number of replacements and the number of relevant documents not retrieved (line 7). If all the relevant documents are retrieved, no replacements are performed.

[4] The same operations were identified by Ferrante et al. (2015) in a different context, i.e., they defined a partial order among rankings via swaps and replacements. In this paper we focus on the differences between rankings rather than their mutual order.

Algorithm 1 Deterioration with positive swaps and replacements

Input: Original ranking r, Number of swaps s, Number of Replacements p, Source interval $[a, b]$, Destination interval $[c, d]$
Output: Deteriorated ranking r'

1: NR_src ← number of non-relevant documents in $[a, b]$
2: R_dst ← number of relevant documents in $[c, d]$
3: R_nrt ← number of relevant and not retrieved documents
4: $r' \leftarrow r$ ▷ initialize the deteriorated ranking
5: **if** NR_src > 0 **then**
6: $\hat{s} \leftarrow \min(s, \text{R\_dst})$ ▷ actual number of swaps
7: $\hat{p} \leftarrow \min(p, \text{R\_nrt})$ ▷ actual number of replacements
8: tot_op $\leftarrow \hat{s} + \hat{p}$ ▷ total number of operations
9: **if** tot_op $>$ NR_src **then**
10: $c \leftarrow \left\lfloor \frac{s \cdot \text{NR\_src}}{s+p} \right\rceil$
11: $\hat{s} \leftarrow \min(c, \text{R\_dst})$
12: $\hat{p} \leftarrow \min(\text{NR\_src} - \hat{s}, \text{R\_nrt})$
13: tot_op $\leftarrow \hat{s} + \hat{p}$
14: **end if**
15: rk_NR_src ← rank positions of non-relevant documents in $[a, b]$
16: rk_NR_src.random_shuffle()
17: rk_R_dst ← rank positions of relevant documents in $[c, d]$
18: rk_R_dst.random_shuffle()
19: **for** $k = 0, \ldots, \hat{s}$ **do**
20: $i \leftarrow$ rk_NR_src$[k]$
21: $j \leftarrow$ rk_R_dst$[k]$
22: $r'[i] = r[j]$ ▷ Document at rank j becomes the document at rank i
23: $r'[j] = r[i]$ ▷ Document at rank i becomes the document at rank j
24: **end for**
25: **for** $k = \hat{p}, \ldots,$ tot_op **do**
26: list_R_nrt ←list of relevant and not retrieved documents
27: $i \leftarrow$ rk_NR_src$[k]$
28: $r'[i] \leftarrow$ list_R_nrt.pop() ▷ Replacement at rank i
29: **end for**
30: **end if**
31: **return** Deteriorated ranking r'

Third, the total number of operations performed by the algorithm is set to the sum of the number of actual swaps and replacements (line 8). The total number of operations has to be lower or equal to the number of non-relevant documents in the source interval. If this requirement is not met (line 9), then the actual number of swaps and replacement is decided so that the original proportion of inputted swaps and replacements is preserved. This is determined by solving the following system of equations with variables $\hat{s}$ and $\hat{p}$:

$$\begin{cases} \frac{\hat{s}}{\hat{p}} = \frac{s}{p} \\ \hat{s} + \hat{p} = \text{NR\_src} \end{cases} \tag{6}$$

where NR_src is the total number of non-relevant documents in the source interval. The integer solution is given in Eq. (7):

$$\hat{s} = \left\lfloor \frac{s \cdot \text{NR\_src}}{s+p} \right\rceil \qquad \hat{p} = \text{NR\_src} - \hat{s} \tag{7}$$

where $\lfloor \cdot \rceil$ denotes the closest integer (round function).

Finally, the actual number of swaps and replacements needs to be compared with the available slots. Therefore, the actual number of swaps is set to the minimum between the value in Eq. (7) and the number of relevant documents in the destination interval (line 11). The actual number of replacements, is set to the minimum between the value in Eq. (7) and the number of relevant documents that are not retrieved (line 12).

Fig. 5 reports the upper bounds for the number of operations with respect to each combination of swap and replacements: $\text{rel}[\cdot, \cdot]$ and $\text{nrel}[\cdot, \cdot]$ are the number of relevant and non-relevant documents in the specified interval, R_rt and R_nrt are the number of relevant documents retrieved and not retrieved, and NR_rt is the number of non-relevant documents retrieved. Note that there are two different cases, defined by the location where operations are performed: (1) swap and replacement operate on the same set of relevant or non-relevant documents (archetypes I and III); (2) swap and replacement operate on disjoint sets of documents, relevant versus non-relevant documents (archetypes II and IV). The case determines the upper bound on the total number of operations: in

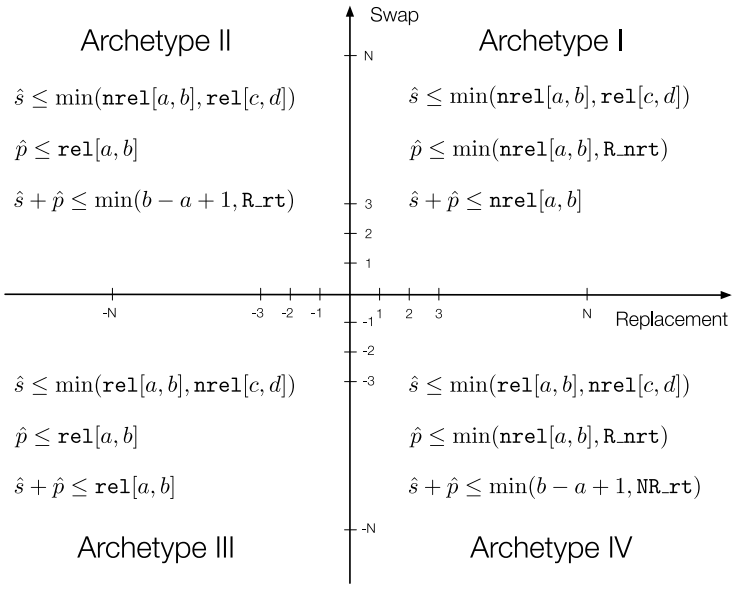


Fig. 5. Upper bounds on the number of operations for each archetype.

case (1) the sum of swaps and replacements should be always lower than the available slots, i.e., the number of relevant documents in $[a, b]$ for archetype I, and the number of non-relevant documents in $[a, b]$ for archetype IV; in case (2), if we ignore all the other constraints, we can perform a swap and a replacement for each document in $[a, b]$, so the greatest upper bound is the number of rank positions in the interval, i.e., $b - a + 1$.

The rest of the algorithm performs swaps (lines 19–24) and replacements (lines 25–29). Given the list of rank positions of non-relevant documents in the source interval (line 15) and the list of rank positions of relevant documents in the destinations interval (line 17), $\hat{s}$ rank positions are selected randomly by shuffling those lists (lines 16 and 18). Then, non-relevant documents are swapped from the source interval with relevant documents in the destination interval (lines 29–24).

Similarly, $\hat{p}$ rank positions are selected randomly in the list of rank positions of non-relevant documents in the source interval. This set does not contain rank positions where a swap was performed, so a swapped document cannot be replaced. Then, given the list of relevant documents that are not retrieved, non-relevant documents are replaced by relevant ones (line 28).

In the experiments (see Section 4), Algorithm 1 is applied iteratively with the inputs p and s in specified ranges. This generates deteriorated rankings with an increasing number of replacements and swaps. In each iteration, the documents that are swapped and replaced are chosen randomly and independently of the previous iterations. Let $r'_{p,s}$ be a ranking deteriorated with p replacements and s swaps. Then $r'_{p+1,s+1}$ is not equal to $r'_{p,s}$ with an additional replacement and swap, but it is a new ranking where the rank positions of swaps and replacements are randomly chosen and might not overlap with the rank positions chosen in the previous iteration.

4. Experiments

Next, we present the experimental analysis of reproducibility measures when runs are deteriorated with swaps and replacements. First, we describe the experimental set-up and implementation details (Section 4.1), then we analyze the properties of different experimental scenarios (Section 4.2), finally we report and discuss the experimental results (Section 4.3). All the source code to run the experiments is publicly available.[5]

4.1. Experimental set-up

Experimental scenarios. We define 3 different experimental scenarios (Table 1), where we vary the original run, i.e., the input run of the deterioration algorithm. In the first 2 scenarios, we use simulated runs with one topic, i.e., a single ranking of documents. This allows us to better understand the behavior of reproducibility measures in a simple controlled setting. We generate 3 different types of simulated rankings:

- Perfect ranking: all relevant documents are at the top of the ranking;

[5] https://github.com/irgroup/ipm-reproducibility

Table 1
Experimental scenarios defined by the number and types of topics, number and types of runs, and recall score.

Scenario 1	Scenario 2	Scenario 3
1 simulated topic	1 simulated topic	50 real topics
3 simulated runs	3 simulated runs	2 real runs
Recall = 1	Recall = 0.5	–

- Realistic Ranking: relevant documents are placed with decreasing probability while going down in the ranking (exponential distribution);
- Reversed ranking: all relevant documents are placed at the end of the ranking.

We differentiate scenarios 1 and 2 by the recall of the rankings. In scenario 1, the simulated rankings retrieve all the relevant documents, i.e., recall equal to 1. In Scenario 2, the simulated rankings do not retrieve all the relevant documents, i.e., we assume recall equal to 0.5.

In scenario 3, we used 2 real runs, BM25 (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1994; Robertson & Zaragoza, 2009) and RM3 (Lavrenko & Croft, 2001), with 50 topics each. This allows us to generalize the behavior from simulated simple settings (1 and 2) to real runs.

Experimental settings. In our analysis, we consider all the reproducibility measures in Section 3.1: Kendall's τ, RBO, RMSE, nRMSE and p-values, instantiated as in Breuer et al. (2020). Specifically, RBO is computed with $\phi = 0.8$, which roughly corresponds to greater weight on the top 5 rank positions (the smaller ϕ, the more top-heavy the measure) (Webber et al., 2010). RMSE and nRMSE are instantiated with AP, nDCG, and Precision@10 (P@10). Recall that in scenarios 1 and 2 there is a single topic, so we cannot compute the p-value.

The simulated runs in scenarios 1 and 2 are the same, they retrieve 1000 documents, out of which 100 are relevant. In scenario 2, 100 extra relevant documents are added outside the ranking (relevant and not retrieved documents), so the recall is equal to 0.5. 32In scenario 3, the real runs are derived from the Washington Post Corpus (v2) and the topics of TREC Common Core 2018.[6]

We chose this collection because: (1) deep pools were used to collect relevance labels, so there is a good amount of relevant documents that can be swapped or replaced; (2) the same collection was used in Breuer et al. (2020), so the results from our experiments with deteriorated runs can be compared with the results with real reproducibility runs in Breuer et al. (2020); (3) the collection is recent and better reflects the performance of state-of-the-art IR systems. On this collection, we validate two runs derived with either the baseline method BM25 (Robertson et al., 1994) or the advanced method BM25 combined with RM3 (Jaleel et al., 2004). For both indexing and retrieval, we make use of the Anserini toolkit (Yang et al., 2018). The document collection is indexed with Porter stemming and Indri's stopword list (Strohman, Metzler, Turtle, & Croft, 2005). During retrieval, we use the topic's title and description as the query string and instantiate both BM25 ($k1 = 0.9$, $b = 0.4$) and RM3 (10 feedback terms, 10 feedback documents, an original query weight of 0.5) with Anserini's default settings.

The deterioration algorithm is implemented in Python 3.9 and run with source interval $[1, 500]$ (first half) and destination interval $[501, 1000]$ (second half). The number of swaps and replacements is varied from 0 to 250 with step 1. Indeed, having 500 available rank positions, we can perform at most 250 swaps and 250 replacements. In scenario 3, the same input parameters are applied to all topics.

The deterioration algorithm is applied iteratively for each pair (p, s) of replacements and swaps. In scenarios 1 and 2 this corresponds to 62 500 randomly deteriorated rankings for each run and to 3 125 000 randomly deteriorated rankings for each run in scenario 3.

All the reproducibility measures are computed with the Python toolkit `repro_eval` (Breuer et al., 2021). Measure scores are reported with heatmaps, as illustrated in the visual representation of the 4 archetypes in Fig. 4. The x-axis represents replacements and the y-axis swaps. Negative integers denote negative operations, i.e., negative replacements or swaps. All heatmaps are generated with seaborn[7] in Python and are down-sampled with step 5. The complete figures with step 1 are not included in this paper due to their large size (~5 MB each figure), but are stored in a public Zenodo archive.[8] For the sake of better readability and understandability, we have included the corresponding Tables with numerical results in Appendix. Note that for RMSE, nRMSE and the p-values, the Tables only contain AP scores. The remaining results of nDCG and P@10 can be found in the public code repository.

4.2. Analysis of experimental scenarios

Next we analyze the properties of the experimental scenarios. This provides the context to better understand the experimental results in Section 4.3.

[6] https://trec-core.github.io/2018/
[7] https://seaborn.pydata.org/
[8] https://zenodo.org/record/5902542

4.2.1. Scenario 1: Simulated runs with recall 1

As summarized in Table 1, scenario 1 considers 3 simulated runs with one topic. All the 3 runs retrieve all the 100 relevant documents, therefore they have recall = 1.

The Recall Base (RB), i.e., the total number of relevant documents, sets upper bounds on the number of replacements that can be performed, as detailed in Section 3.2. Specifically, no positive replacements can be performed, since there are no relevant documents that are not retrieved. For negative replacements, we can at most replace RB relevant documents with non-relevant ones, thus a maximum of RB negative replacements.

Similarly, the type of run, sets upper bounds on the number of swaps that can be performed. For the perfect ranking, no positive swaps can be performed, since all the relevant documents are already placed at the beginning of the ranking. This, combined with the absence of positive replacements, means that for the perfect ranking there is no possible deterioration within archetype I (first quadrant). Therefore, in all figures of the perfect ranking in scenario 1, the first quadrant achieves the best score for all measures.

Conversely, for the reversed ranking, we cannot perform negative swaps, since all the relevant documents are already placed at the end of the ranking. Moreover, we cannot perform negative replacements, since in the source interval (first half of the ranking) there are no relevant documents to replace. Therefore, for the reversed ranking we cannot perform any type of replacement and negative swaps, thus the run cannot be deteriorated with respect to archetypes III and IV. This means that the third and fourth quadrants achieve the best score for all measures for the reversed ranking in scenario 1.

Finally, the total number of operations, i.e., sum of swaps and replacements, cannot exceed the available slots, as detailed in Fig. 5. For example, for the perfect ranking, we can perform at most RB, i.e., 100, negative replacements and negative swaps. This explains why in all figures there is a central region where we can see variations in the measure score, this is the region where the sum of operations is lower than the available slots. Moving further from the center, the number of operations reaches the upper bound, so there is no more room to deteriorate the run and the measure reaches a plateau.

4.2.2. Scenario 2: Simulated runs with recall less than 1

Scenario 2 exploits exactly the same runs as scenario 1, but 100 extra relevant and not retrieved documents are added, so recall drops from 1 to 0.5. This allows us to consider extra operations, as positive replacements, that cannot be performed in scenario 1. Therefore, the upper bound on the number of replacements corresponds to $RB/2 = 100$ both for positive and negative replacements.

As discussed for scenario 1, the type of run determines further constraints on the number of swaps and replacements. For the perfect ranking, positive swaps cannot be performed, since all the documents are already at the top of the ranking. However, positive replacements can happen, since there are 100 relevant not retrieved documents. Therefore, differently from scenario 1, in scenario 2, the measure scores for the perfect ranking change in all quadrants.

For the reversed ranking, negative swaps cannot occur, since all the relevant documents are already placed at the end of the ranking. Negative replacements cannot occur as well, since there are no relevant documents in the source interval, i.e., the first half of the ranking. However, differently from scenario 1, we can perform positive replacements. This means that there is no deterioration for archetype III (negative replacements and swaps), while for all other archetypes we can perform some operations, so the measure scores can change.

As in scenario 1, the total number of operations (replacements and swaps) is limited by the number of available slots, as illustrated in Fig. 5. Therefore all figures corresponding to scenario 2 exhibit a central region, where the measure scores decrease, and an outer region, where the measure scores reach a plateau.

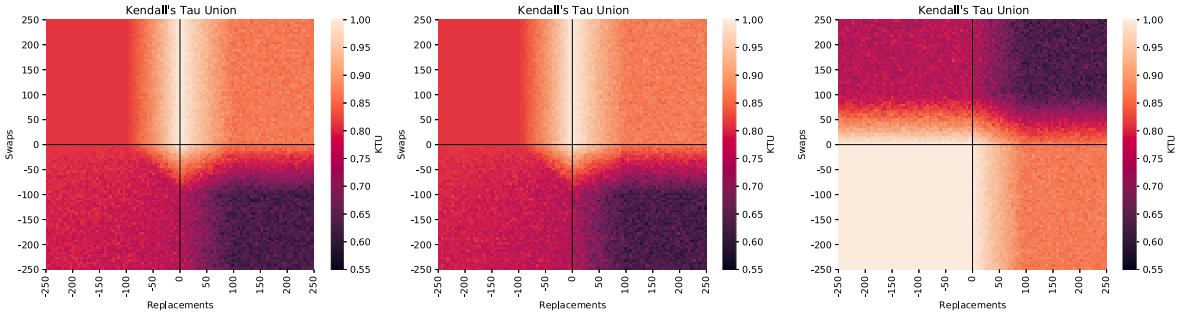
4.2.3. Scenario 3: Real runs

In scenario 3 we use real runs with 50 topics computed with BM25 and RM3. In principle, all the operations can be performed and the total number of operations is constrained by the type of run, i.e., number of relevant documents and their location, as illustrated in Fig. 5. Therefore, all figures corresponding to real runs exhibit a trend similar to scenarios 1 and 2, i.e., a central region with varying scores and an outer region with a plateau score.

On average across topics, RM3 run performs slightly better than BM25 run with respect to all IR evaluation measures under consideration (AP, nDCG and P@10). This means that, on average, RM3 retrieves more relevant documents closer to the top of the ranking than BM25. As a consequence, the total number of available positive operations is lower for RM3 than BM25. For example, since RM3 has a higher recall, there are fewer relevant not retrieved documents that can be used for positive replacements. Conversely, the number of negative operations is higher for RM3 than BM25. For example, since RM3 has a higher recall, there are more relevant retrieved documents that can be replaced by non-relevant documents.

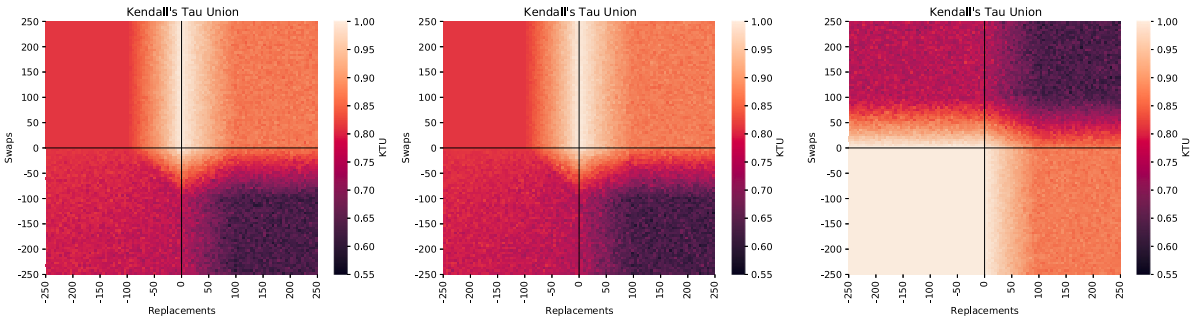
4.3. Experimental results

Next, we present the experimental results for the three different scenarios: (1) simulated rankings that retrieve all relevant documents; (2) simulated rankings that retrieve half of the relevant documents; (3) real runs computed with BM25 and RM3. The results are organized by measure, starting with ranking-based measures with KTU and RBO (Section 4.3.1), effectiveness-based measures with RMSE and nRMSE (Section 4.3.2), and statistical test measures with p-values (Section 4.3.3).



(a) Perfect Ranking (b) Realistic Ranking (c) Reversed Ranking

Fig. 6. Scenario 1 - Kendall's τ Union: Simulated runs with a single topic. Each run retrieves all relevant documents (Recall = 1). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[1, 500]$ to $[501, 1000]$.



(a) Perfect Ranking (b) Realistic Ranking (c) Reversed Ranking

Fig. 7. Scenario 2 - Kendall's τ Union: Simulated runs with a single topic. Each run retrieves only half of the relevant documents (Recall = 0.5). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

4.3.1. Ranking-based measures: KTU and RBO

Fig. 6 shows KTU values for the deteriorated runs in scenario 1. As expected, we can see that KTU reaches the best score equal to 1 for archetype I with the perfect ranking (Fig. 6(a)) and for archetypes III and IV with the reversed ranking (Fig. 6(c)). KTU for the realistic ranking (Fig. 6(b)) looks very similar to the perfect ranking. This happens because most of the relevant documents are placed in $[1, 500]$, as in the perfect ranking, and Kendall's τ is not top-heavy, so it does not account for the rank position where a swap or replacement occurred.

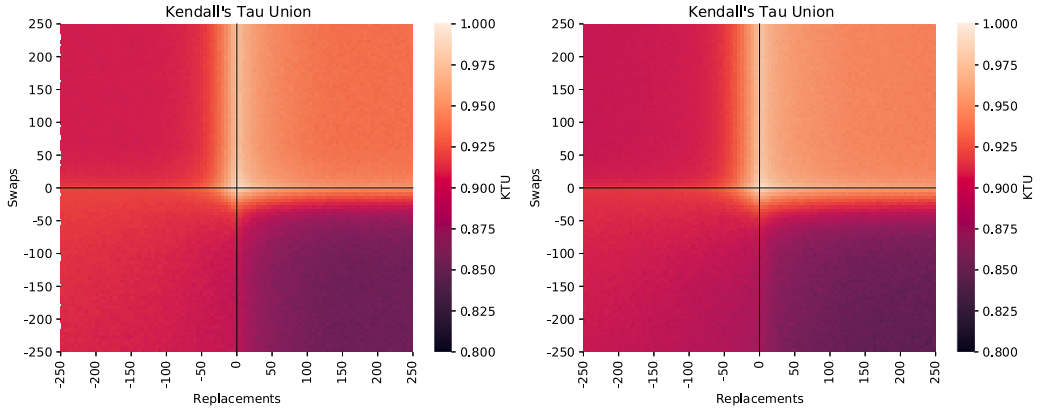
Fig. 7 reports KTU values for scenario 2, where positive replacements can occur. The effect of positive replacements is clearly seen on the right part of each subfigure, indeed Fig. 7 can be obtained from Fig. 6 by overlaying the effect of positive replacements in quadrants I and IV. Moreover, the minimum KTU score drops from scenario 1 to 2, since positive replacements represent additional operations that we can perform to deteriorate the runs.

The perfect and realistic rankings exhibit a similar behavior, as observed in scenario 1. This happens for the same reason, i.e., the perfect ranking retrieves most of the relevant documents in the source interval $[1, 500]$ and KTU does not account for the rank position.

The bright region where the measure score is close to the best score is reduced from the whole archetype I in Fig. 6 to an horizontal thick line corresponding to the zero replacements axis and positive swaps in Fig. 7. This happens because: (1) we can perform positive replacements, which affects archetype I; (2) the number of positive swaps is upper bounded by the number of relevant documents in the interval $[501, 1000]$, which is 0 or close to 0 for the perfect and realistic ranking. Therefore, for the perfect and realistic ranking we can mostly observe the effect of replacements in archetypes I and II. For the reversed ranking in Fig. 7(c), the bright region is reduced to archetype III, since no negative replacements and swaps can occur.

For the perfect and realistic rankings the worst region in terms of KTU is archetype IV, which corresponds to the region where the highest number of operations can be performed (up to 100 positive replacements and 100 negative swaps). This is followed by archetype III, where the 100 available slots are split between negative replacements and swaps, archetype II, where no positive swaps can be performed, and archetype I, where there are only positive replacements. The reversed ranking behaves in a somehow specular way, with archetype I, the one with the greatest number of operations, being the worst, and archetype III, no available operations, being the best.

Fig. 8 shows KTU values for the real runs of scenario 3. In this case we report KTU averaged across all 50 topics. At a first glance, the behavior of the real runs is very similar to the perfect and realistic rankings in Fig. 7. The main difference is that the



(a) BM25 (b) RM3

Fig. 8. Scenario 3 - Kendall's τ Union: BM25 and RM3 runs on TREC Common Core 2018. The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

vertical bright line is thinner for real runs and can be seen also for the horizontal axis corresponding to zero swaps and positive replacements. This might happen because of the upper bounds in terms of positive operations, i.e., there are not so many relevant documents that are not retrieved that we can use for positive replacements, or relevant documents in $[501, 1000]$ that we can move towards the beginning of the ranking.

As for the scenario 2, the worst region is archetype IV, where we can perform the highest number of operations because positive replacements and negative swaps are somehow complementary (see Fig. 5). The best region is archetype I, where the total number of operations is constrained by the number of relevant documents not retrieved and the number of relevant documents retrieved at the end of the ranking. When we compare BM25 and RM3, Figs. 8(a) and 8(b), we can see that archetype I is slightly brighter for RM3 than BM25. This happens because, on average, RM3 retrieves more relevant documents and closer to the top of the ranking, so there are less available slots for positive operations.

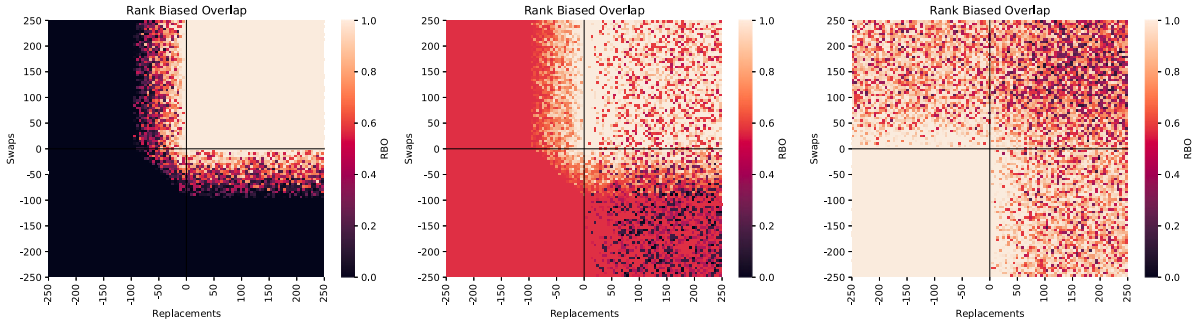
In general, all scores in KTU figures (Figs. 6, 7, and 8) are higher then those reported in Breuer et al. (2020), where KTU reaches values close to 0. This is due to two main reasons: (1) the upper bound on the number of operations; (2) the type of operations. First, with real reproducibility runs there is no upper bound to the number of negative replacements: for example we can always replace a document with a non-relevant one. Second, real runs include other types of deterioration operations, as replacements and swaps in the same relevance class. For example, swapping two non-relevant or relevant documents affects KTU, but it is not considered in our analysis. As already mentioned, this synthetic experimental set-up allows us to control for variables such as the number of replacements and swaps, while keeping the level of complexity low, which allow us to interpret the results.

> As seen from the darker regions in the heatmaps of the simulated rankings and real runs, archetype IV has the lowest KTU scores. It means that a less effective first-stage ranker combined with a more effective re-ranker leads to the least reproduced version of the original ranking in terms of KTU. If the first stage ranking deteriorates, the exact document order cannot be reproduced even if the re-ranking is better than in the original experiment.

Fig. 9 reports RBO in scenario 1. Even if both KTU and RBO considers the rank position of documents, we can clearly see the difference in their behavior, due to RBO being top-heavy. Indeed, the frontier region is more clear for RBO than KTU and RBO shows a more diverse behavior across runs.

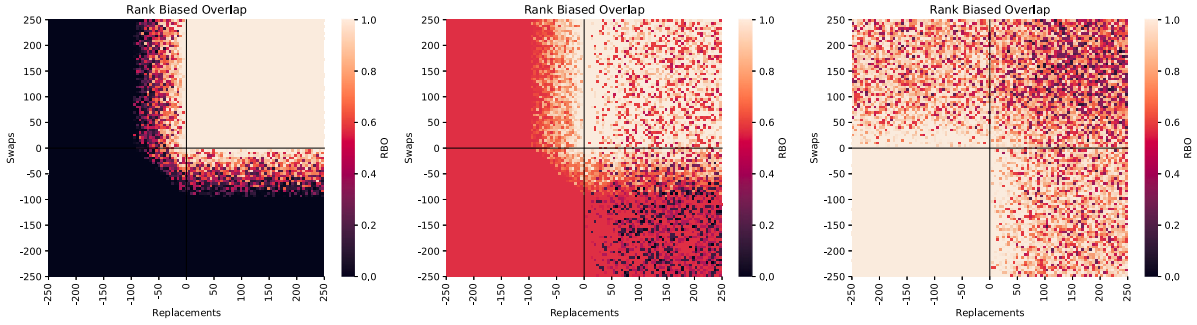
The perfect and the realistic rankings (Figs. 9(a) and 9(b)) exhibit again a similar behavior. RBO is equal or close to the perfect score for archetype I and there is a similar frontier region, where RBO transitions from the best score to lower scores. The size of the frontier region corresponds to the maximum number of operations that can be performed: a straight line corresponding to -100 in quadrant II and IV and the bisector with sum -100 in quadrant III. This happens because we can perform up to 100 negative replacements and no positive swap in quadrant II; up to 100 negative swaps and replacements in quadrant III; and up to 100 negative swaps and no positive replacement in quadrant IV.

The main difference between the perfect and realistic rankings is the color of the outer region, i.e., the value of RBO when the total number of deterioration operations is performed. For the perfect ranking this score is close to 0, the worst score, for the realistic ranking this score is close to 0.56. This behavior is due to RBO being top-heavy: for the perfect ranking, all the documents at rank positions $[1, 100]$ are replaced by non-relevant documents or moved at the end of the ranking, both these operations have a great impact on RBO. For the realistic ranking the relevant documents are spread in the ranking, so their swaps or replacement has less impact on RBO score.



(a) Perfect Ranking (b) Realistic Ranking (c) Reversed Ranking

Fig. 9. Scenario 1 - RBO: Simulated runs with a single topic. Each run retrieves all relevant documents (Recall = 1). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[1, 500]$ to $[501, 1000]$.



(a) Perfect Ranking (b) Realistic Ranking (c) Reversed Ranking

Fig. 10. Scenario 2 - RBO: Simulated runs with a single topic. Each run retrieves only half of the relevant documents (Recall = 0.5). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

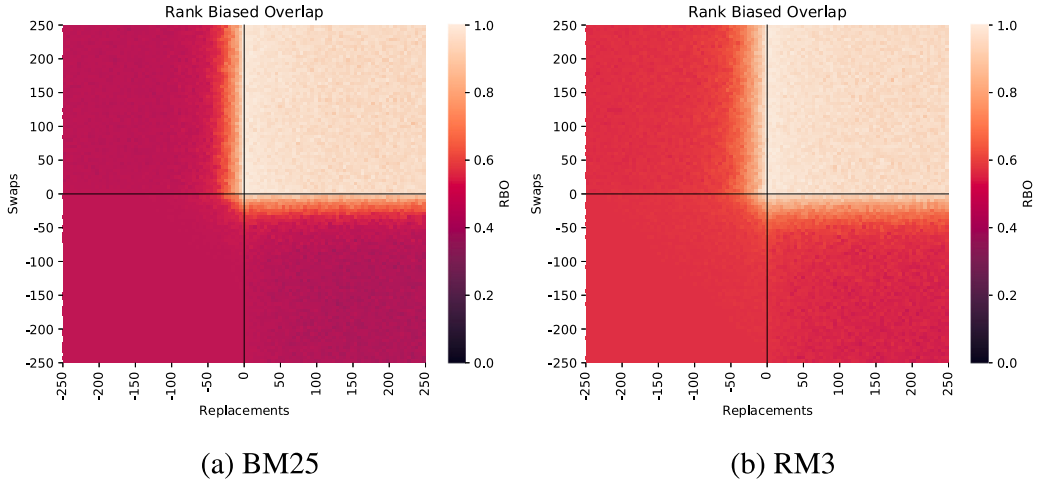
The reversed ranking (Fig. 9(c)) exhibits a different behavior. There is no change in RBO scores for quadrant III and IV, since we cannot perform any negative swap nor positive or negative replacement. Therefore, the upper half of Fig. 9(c) shows only the effect of positive swaps, from the end of the ranking to $[1, 500]$. RBO does not reach a plateau for the reversed ranking because its value heavily depends on the rank positions where relevant documents are swapped, which are randomly chosen with up to 100 choices over 500 rank positions in each iteration.

Fig. 10 shows RBO values in scenario 2, i.e., the same runs of scenario 1, but with 100 extra relevant and not retrieved documents. We can clearly see the effect of positive replacements for the realistic and reversed rankings (Figs. 10(b) and 10(c)): as for KTU, Fig. 10 can be obtained from Fig. 9 by overlaying the layer generated by positive replacements on quadrants I and IV. The effect of positive replacements cannot be seen for the perfect ranking (Fig. 10(a)) due to RBO top heaviness. Indeed the top of the ranking is already filled with relevant documents and positive replacements affect rank positions chosen at random in $[101, 500]$, which affect RBO score only to a small extent.

Conversely, for the realistic ranking, even a few replacements in the interval $[1, 100]$ can considerably decrease RBO score. Moreover, for the realistic ranking, archetype I does not exhibit a uniform color, i.e., a plateau score, because the drops in RBO heavily depends on the rank position where a positive replacement occurs. The same reasoning applies also to the reversed ranking. For the realistic ranking we can still see the vertical bright region corresponding to few replacements, which is also visible for KTU (see Fig. 7(a)).

When we compare archetypes, the difference is not marked as clearly as for KTU. Archetypes IV and I are still the worst regions for the realistic and reversed rankings respectively. These correspond to the regions where the maximum number of operation can be performed, accordingly to the constraints in Fig. 5. For the perfect ranking there is no clear distinction among archetypes where RBO reaches very low values close to 0. This happens because negative operations affects the very top of the raking in $[1, 100]$, thus they decrease RBO score to a great extent.

Fig. 11 shows RBO averaged across 50 topics for the real runs in scenario 3. The behavior of both runs is somehow in between the perfect and the realistic ranking: there is a bright area corresponding to archetype I and archetype IV is slightly darker than all other regions. The impact of positive replacements can still be seen in the quadrants I and IV of both figures, but it is not as strong as for the realistic ranking in Fig. 10(b). The reason is that RBO is averaged across topics, and for several of them there are only a few relevant documents not retrieved that can be added.



(a) BM25 (b) RM3

Fig. 11. Scenario 3 - RBO: BM25 and RM3 runs on TREC Common Core 2018. The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

In general, RBO scores in all figures (Figs. 9, 10, and 11) is within the ranges reported by Breuer et al. (2020). The only exception is the perfect ranking, which exhibits lower values. Indeed, the type of deterioration that can be performed for the perfect ranking is quite extreme and not likely in a real case, e.g., moving/replacing all documents at the top of the ranking.

RBO behaves differently from KTU because of its top-heaviness. For RBO, the deterioration operations that occur at the beginning of the run have greater impact on the scores. Therefore, rather the quantity, as for KTU, for RBO the location of those operations is the most important factor.

> Compared to KTU, the RBO can be parameterized as a more top-heavy rank correlation measure. Depending on the parametrization, RBO reaches a plateau for archetypes II, III, and IV, as can be seen for the simulated rankings and real runs. It means that a more effective first-stage ranker cannot compensate for a less effective re-ranker (and vice versa) regarding the reproduction quality of a top-weighted ranking once they deviate too much from the original ranking methods.

4.3.2. Effectiveness-based measures: RMSE and nRMSE

Fig. 12 shows RMSE value in scenario 1 for all effectiveness measures: AP (first row), nDCG (second row), and P@10, (third row). Recall that, for the perfect ranking (first column), positive replacements and swaps cannot be performed, so archetype I corresponds to RMSE = 0. Conversely, for the reversed ranking (third column), negative swaps and replacements cannot occur, so archetype III and IV have RMSE = 0. Finally, since recall = 1 in scenario 1, there are no positive replacements.

Scenarios 1 and 2 consider simple simulated runs with a single topic. These represent special cases for RMSE, where RMSE score is the absolute difference between the effectiveness score of the original and reproduced runs.

In scenario 1, RMSE behaves similarly to KTU and RBO (see Figs. 6 and 9): the perfect and realistic rankings exhibit a similar behavior across measures, due to the position of the relevant documents (mostly) at the top of the rankings. Moreover, we can observe the same shape of the frontier region, with the boundary around 100 negative replacements and swaps, and the diagonal line $y = -x - 100$ for quadrant III. Recall that this happens because there are 100 relevant and retrieved documents, so we can perform up to 100 negative replacements or swaps.

In terms of RMSE scores, the perfect ranking assumes a slightly darker tone than the realistic ranking. This is observed also for RBO (Fig. 9) and is due to: (1) the top-heaviness of IR effectiveness measures, and (2) the rank position of relevant documents. Since all relevant documents are placed at the top of the ranking, the perfect ranking can be deteriorated to a greater extent and top-heavy measures are more affected.

The main difference among IR effectiveness measures is the variation of scores across archetypes: nDCG seems more sensitive to the type of deterioration, while AP and P@10 do not exhibit any difference for archetypes II, III, and IV. This is due to the different discount functions: P@10 focuses only on the top 10 positions, thus it does not detect what happens after the cut-off threshold; AP has a steeper discount function than nDCG, thus it is less sensible to replacements and swaps at lower rank positions.

Fig. 13 shows RMSE scores for scenario 2, i.e., the same rankings as in scenario 1, but retrieving half of the relevant documents (recall = 0.5). In this scenario we can perform positive replacements, since there are 100 relevant and not retrieved documents. Therefore, the only archetype where RMSE score does not change is archetype III for the reversed ranking, where negative replacements and swaps cannot occur.

Similarly to KTU and RBO, we can clearly see the effect of positive replacements. For the realistic and reversed rankings, we can still identify this as a sort of overlaying layer in the positive replacement region, i.e., archetypes I and IV. This is more evident

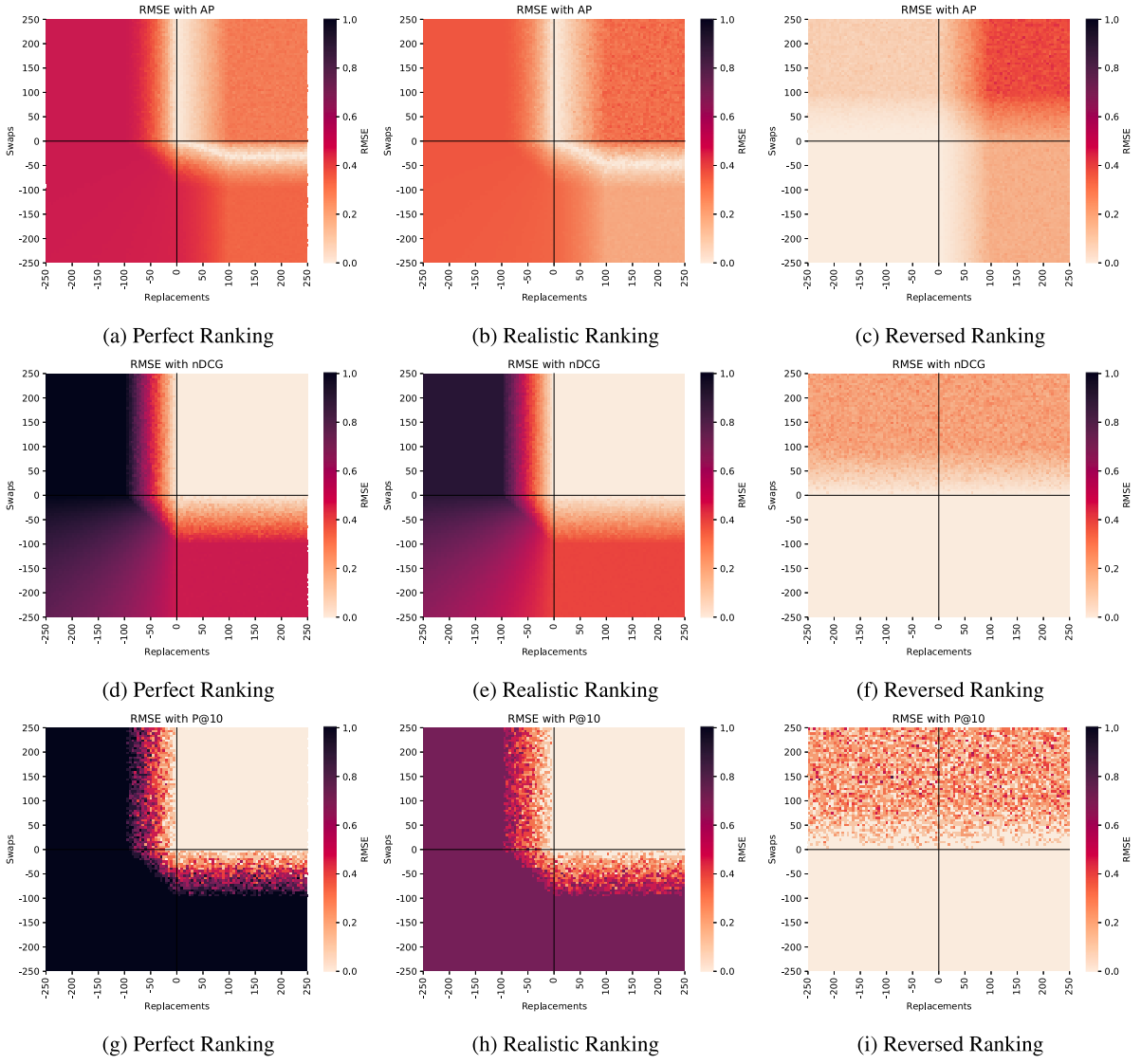


Fig. 12. Scenario 1 - RMSE: Simulated runs with a single topic. Each run retrieves all relevant documents (Recall = 1). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[1, 500]$ to $[501, 1000]$.

for the reversed ranking than the realistic ranking. The only exception, where the effect of positive replacements is not visible at all, is P@10 for the perfect ranking. In this case there is no effect of positive replacements because all the positive replacements happen below the cut-off threshold, indeed the perfect ranking is filled with relevant documents up to position 100 and all positive replacements occur below.

Again AP and nDCG behave in a similar way and in general the perfect ranking is slightly darker than the realistic ranking due to the more extreme deterioration that can be applied to the perfect ranking. Moreover, as in scenario 1, nDCG is more sensitive to different types of operations, assigning different colors to different archetypes. The worst region is located in archetype II, corresponding to negative replacements and positive swaps. Symmetrically, the best region is located within archetype IV, with positive replacements and negative swaps. In this latter region, 100 relevant documents are swapped with documents in the second half of the ranking, but their effect is somehow compensated by the 100 non-relevant documents replaced with relevant ones in the first half of the ranking.

Wile KTU and RBO exhibit a bright vertical region corresponding to the positive y-axis (few replacements and positive swaps), the same region has a different shape for RMSE computed with AP and nDCG. Indeed, the vertical bright line corresponding to the positive y-axis continues in quadrant IV, where it starts on the diagonal and then divides itself in two branches, one vertical and one horizontal. This corresponds to the boundary where negative swaps are compensated by positive replacements. The horizontal branch is higher for AP than nDCG since AP discount function is harsher than nDCG one, therefore a higher number of replacements is needed to compensate relevant documents that are moved at the end of the ranking.

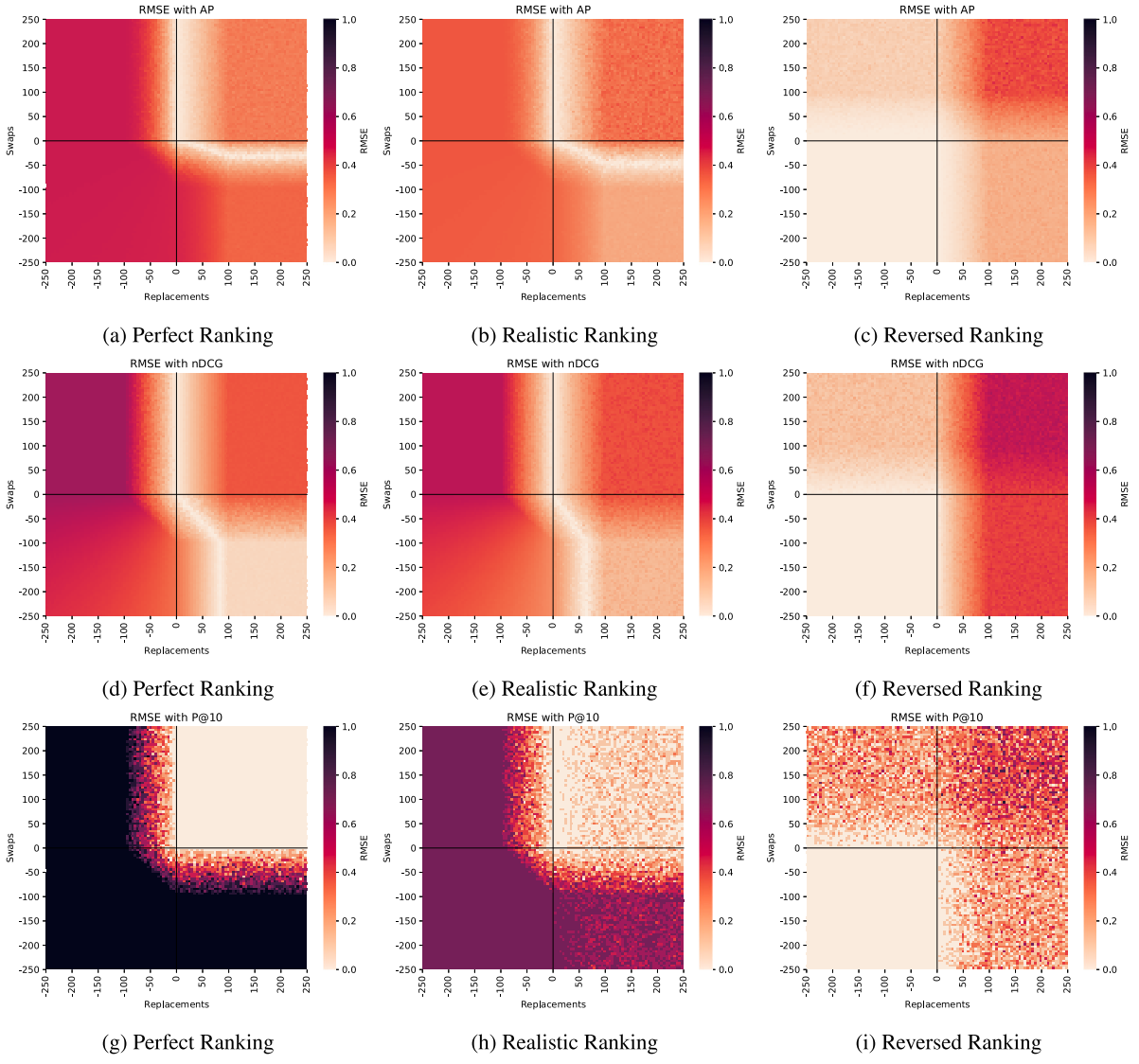


Fig. 13. Scenario 2 - RMSE: Simulated runs with a single topic. Each run retrieves only half of the relevant documents (Recall = 0.5). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

Fig. 14 shows RMSE scores for real runs in scenario 3. Both runs consider 50 topics, thus RMSE is not simply the absolute difference in effectiveness scores, as in scenario 1 and 2. Since scores are aggregated across topics, colors in the heatmaps are less variable and the horizontal and vertical bright lines are barely visible.

In scenario 3, we can observe similar trends as in the other two scenarios. First, AP and nDCG behave in a similar way. They both exhibit a thin bright line corresponding to the positive y-axis. The region in archetype IV where positive replacements compensate negative swaps is not visible for real runs, except for a thin line just below the positive x-axis. Second, nDCG is more sensitive to different archetypes while AP treats them equally. The worst region is still archetype II, where negative replacements and positive swaps occur. Since there are very few relevant documents at the end of the run, they cannot compensate the impact of negative replacements in the first half of the run. Third, RMSE is darker for RM3 than BM25, as it happens for the perfect and realistic rankings. Since RM3 performs slightly better than BM25, deterioration can be more extreme and lead to higher differences in RMSE scores.

When comparing different IR effectiveness measures, Breuer et al. (2020) report that P@10 suffers of higher variance than AP and nDCG. This is clearly shown in Figs. 12 and 13 in the frontier region, where the transition to the plateau score is smoother for AP and nDCG, and in archetype I and II for the reversed ranking, where RMSE does not reach a plateau value. In scenario 3, this effect is less strong but still visible for archetype I.

Compared to the rank correlation measures, the RMSE does not depend on the actual documents but on the relevance labels in the reproduced ranking. RMSE can be instantiated with different retrieval measures, and as such, it shows a different sensitivity behavior across the archetypes depending on the retrieval measure. When instantiated with P@10, the heatmaps are similar to those of RBO, focusing on the reproduction quality of the top-ranked results. Comparing AP and nDCG reveals that the discount function has an impact on the sensitivity of RMSE, while nDCG is more sensitive (cf. archetype II) due to the less harsh discount function. As can be seen from the simulated rankings for a single topic, a more effective re-ranker can indeed compensate for a less effective reproduced first-stage ranker to a certain extent, whereas the required replacements depend on the discount function of the retrieval measure.

Fig. 15 reports nRMSE scores for AP, nDCG, and P@10 in scenario 1. Recall that nRMSE is a version of RMSE normalized by the maximum RMSE score that can be achieved by the reproduced run, i.e., the worst possible reproduced run. Therefore, nRMSE general trend, i.e., appearance of the heatmaps, should not be different from standard RMSE, even if there is a difference in the actual score. Indeed, the heatmaps in Fig. 15 exhibit the same trend of those in Fig. 12, i.e., the same behavior across archetypes and the same shape of the frontier region. As for standard RMSE, nDCG is the only measure able to detect the difference among archetypes (see Figs. 15(d) and 15(e)).

The effect of normalization is clearly visible when comparing Figs. 12 and 15. Indeed, all heatmaps, except for the perfect ranking, are darker with nRMSE, meaning that the relative error (nRMSE) is higher than the absolute error (RMSE). The difference between the perfect and realistic rankings (first and second column) is almost imperceptible. The two rankings are somehow comparable and the same deterioration is applied, which leads to similar nRMSE scores. nRMSE and RMSE for the perfect ranking are the same (Figs. 15 and 16). This happens because the normalization factor for the perfect ranking is equal to 1 (see Eq. (4)).

Fig. 16 reports nRMSE in scenario 2, i.e., the same runs as in scenario 1 but with the addition of 100 relevant and not retrieved documents. As expected, the general appearance of nRMSE is very similar to RMSE (compare Figs. 13 and 16): same shape of the frontier region and same behavior across measures and runs. nDCG is once more the best measure to distinguish among different archetypes. The bright region where negative swaps are counterbalanced by positive replacements is still visible for nDCG and AP with respect to the perfect and realistic ranking. Again, the line where positive replacements compensate negative swaps is closer to the positive x-axis for AP than nDCG. This is due to AP requiring more positive replacements to compensate negative swaps because of its steeper discount function.

The impact of positive replacements can be identified as an additional layer in quadrants I and IV, more visible for P@10 (third row) and the reversed ranking with all measures (third column). This happens also for RMSE, RBO, and KTU. Recall that no positive replacements can occur at rank positions $[1, 10]$, so the effect of positive replacements is not visible for P@10 and the perfect ranking (Fig. 16(g)).

The effect of normalization is still visible as a general darkening of colors in the heatmaps. In scenario 2, this is true also for the perfect ranking, where the normalization factor is < 1 and not 1 as in scenario 1, because there are relevant documents that are not retrieved. The only exception is P@10 with respect to the perfect ranking, because the measure considers only rank positions up to 10. The effect of normalization does not completely remove difference among the perfect and realistic ranking, as in scenario 1, except for P@10. Again, because the perfect ranking places all the retrieved relevant documents at rank positions $[1, 100]$, we can deteriorate this ranking in a more extreme way. This is further combined with the top-heaviness of IR measures, indeed AP is more top-heavy than nDCG, i.e., the difference between the perfect and realistic ranking is more noticeable for AP than nDCG.

Fig. 17 shows nRMSE for scenario 3 with real runs, both with 50 topics. Again the general behavior of nRMSE is similar to RMSE (compare Figs. 14 and 17) and the same considerations are valid: (1) there is a thin bright line corresponding to the positive x-axis and the line where positive replacements counterbalance negative swaps is barely visible; (2) nDCG is better in distinguishing among archetypes than AP; (3) nRMSE is slightly darker for RM3 than BM25.

The effect of normalization can be clearly detected as darker colors across all measures and runs. Again, the relative error shown by nRMSE is higher than absolute differences shown by RMSE.

In comparison to RMSE, the nRMSE is normalized by the worst possible ranking. Overall, similar trends are observable for the archetypes, but the darker heatmaps indicate larger relative errors in comparison to the absolute differences.

4.3.3. Statistical tests: p-values

Next we present the same heatmaps for p-values. These are computed only for scenario 3 because this is the only scenario where there are 50 topics. Results are reported in Fig. 18.

At a first glance we can see that all figures are mainly black meaning that p-value is very sensitive to deterioration in the runs. Indeed, the darker the color, the smaller the p-value, thus the higher the evidence that the original and reproduced runs are significantly different.

Furthermore, all the figures can be interpreted as a sort of "photographic negative" of RMSE and nRMSE in Figs. 14 and 17. The bright line corresponding to the positive y-axis, could be seen also for RMSE and nRMSE, even if not so strongly. The horizontal line close to the positive x-axis represents the border where negative swaps are compensated by positive replacements. This line is closer to the x-axis for AP than nDCG. The same behavior is observed in scenario 2 for RMSE and nRMSE and is due to the different discount functions of these measures.

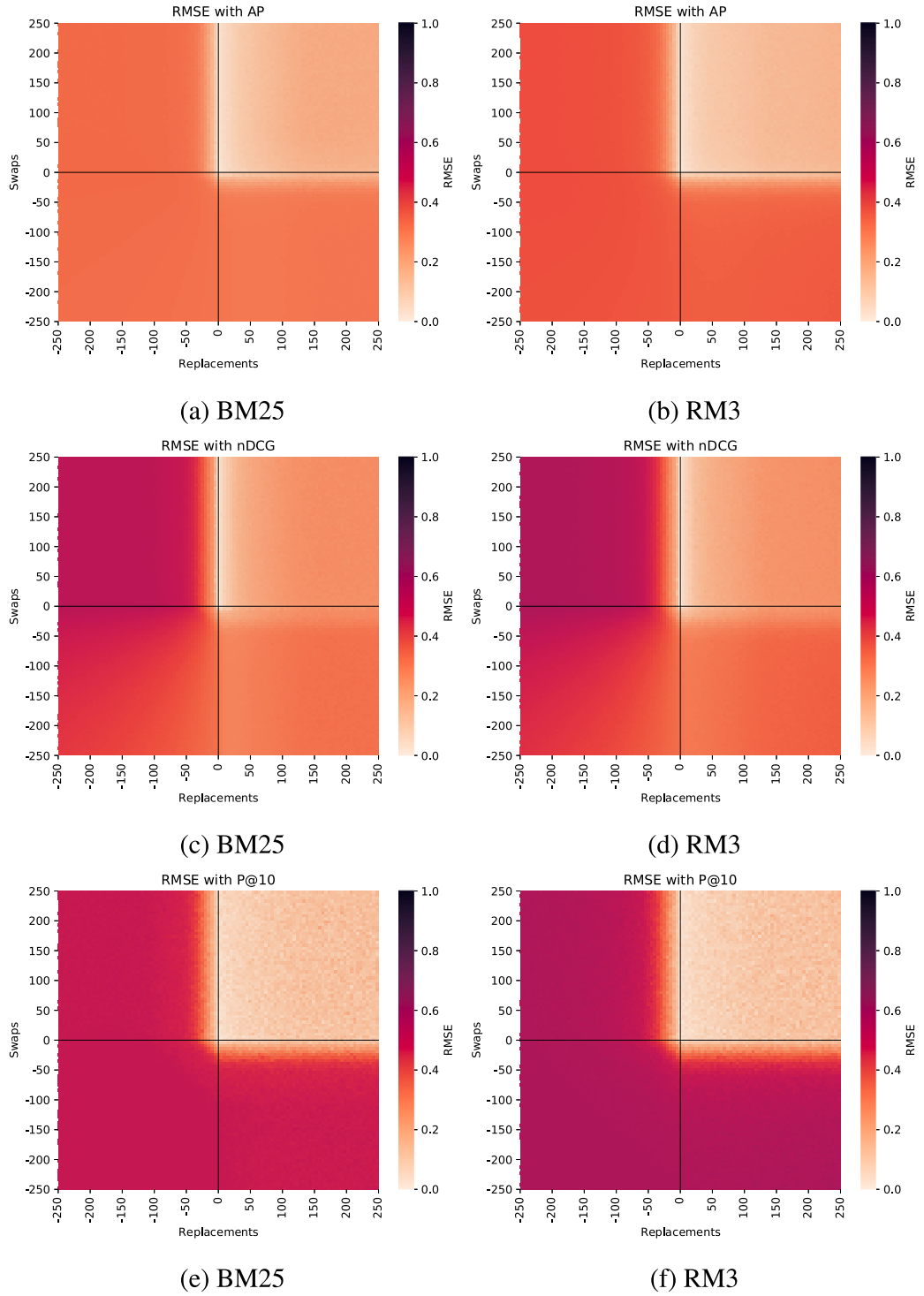


Fig. 14. Scenario 3 - RMSE: BM25 and RM3 runs on TREC Common Core 2018. The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

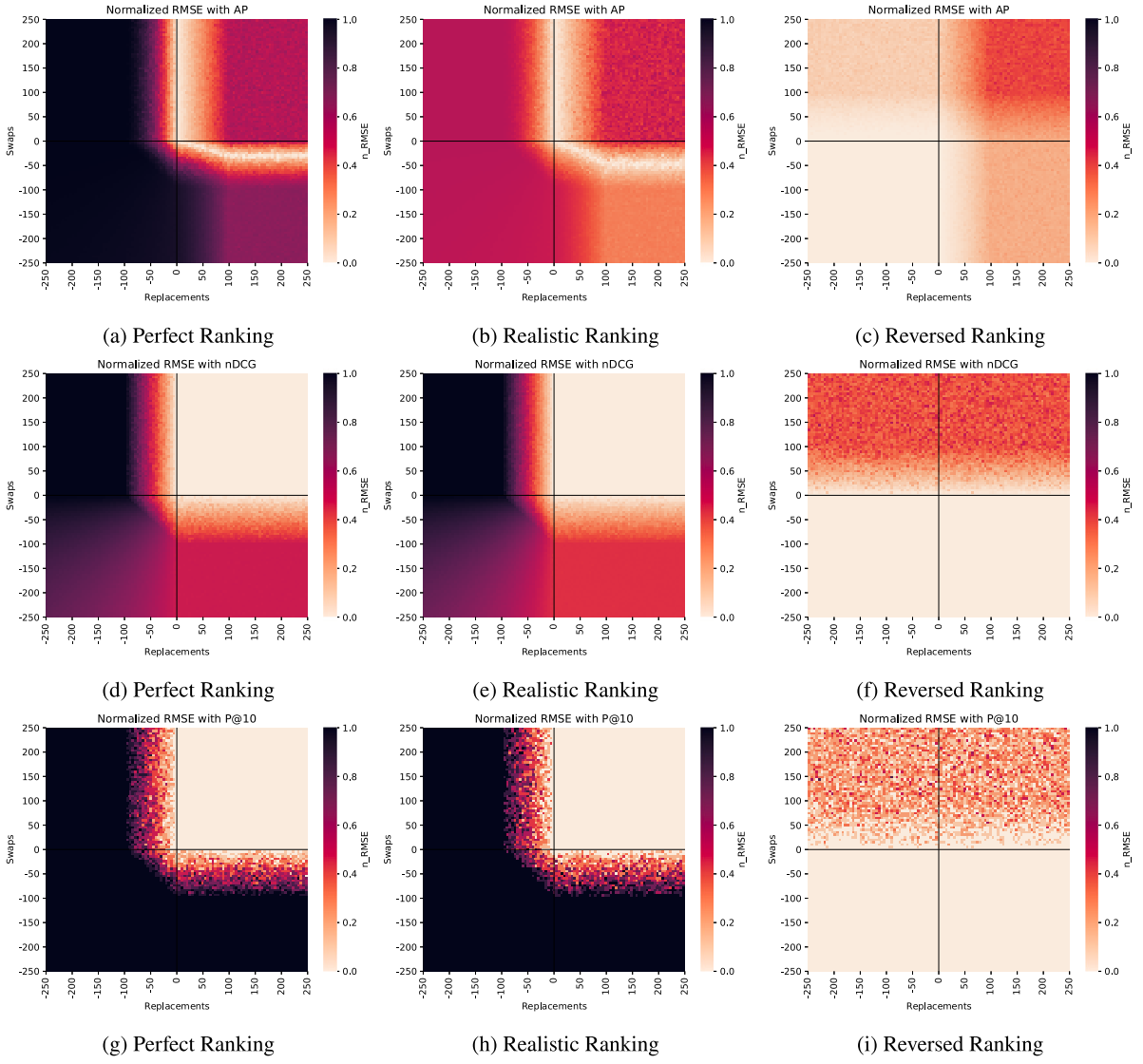


Fig. 15. Scenario 1 - nRMSE: Simulated runs with a single topic. Each run retrieves all relevant documents (Recall = 1). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[1, 500]$ to $[501, 1000]$.

Among different IR effectiveness measures, nDCG (second row) seems the most sensitive, indeed all archetypes are mostly dark. AP (first row) is less sensitive with respect to archetype I. This happens because for real runs we can perform more negative operations, since the number of relevant documents in $[0, 500]$ is in general higher than the number of relevant documents in $[501, 1000]$ or the number of relevant not retrieved. P@10 (third row) is the measure with highest variance, especially with respect to archetype I. Indeed, adding or moving relevant documents at the top of the ranking, either by positive swaps or replacements, has a stronger effect than removing them.

4.4. Final remarks

As reproducibility measures, KTU and RBO compare the ranking of documents for the original and reproduced runs. These are the most stringent reproducibility measures, because they require that the reproduced run ranks the documents in the same exact order as the original run. Both measures account more for the amount of deterioration operations and RBO also for their location, rather than the operation type, i.e., replacement or swap. Up to 50 replacements or swaps can cause KTU drop from 1 to 0.8, which is still a reasonably high correlation in IR (Voorhees, 1998, 2001). However, we can observe much lower correlations in studies with real runs (Breuer et al., 2020). This suggests that the reproduced runs, in a non-controlled experimental setting, can be very different from the original runs, with many more than 50 swaps or replacements.

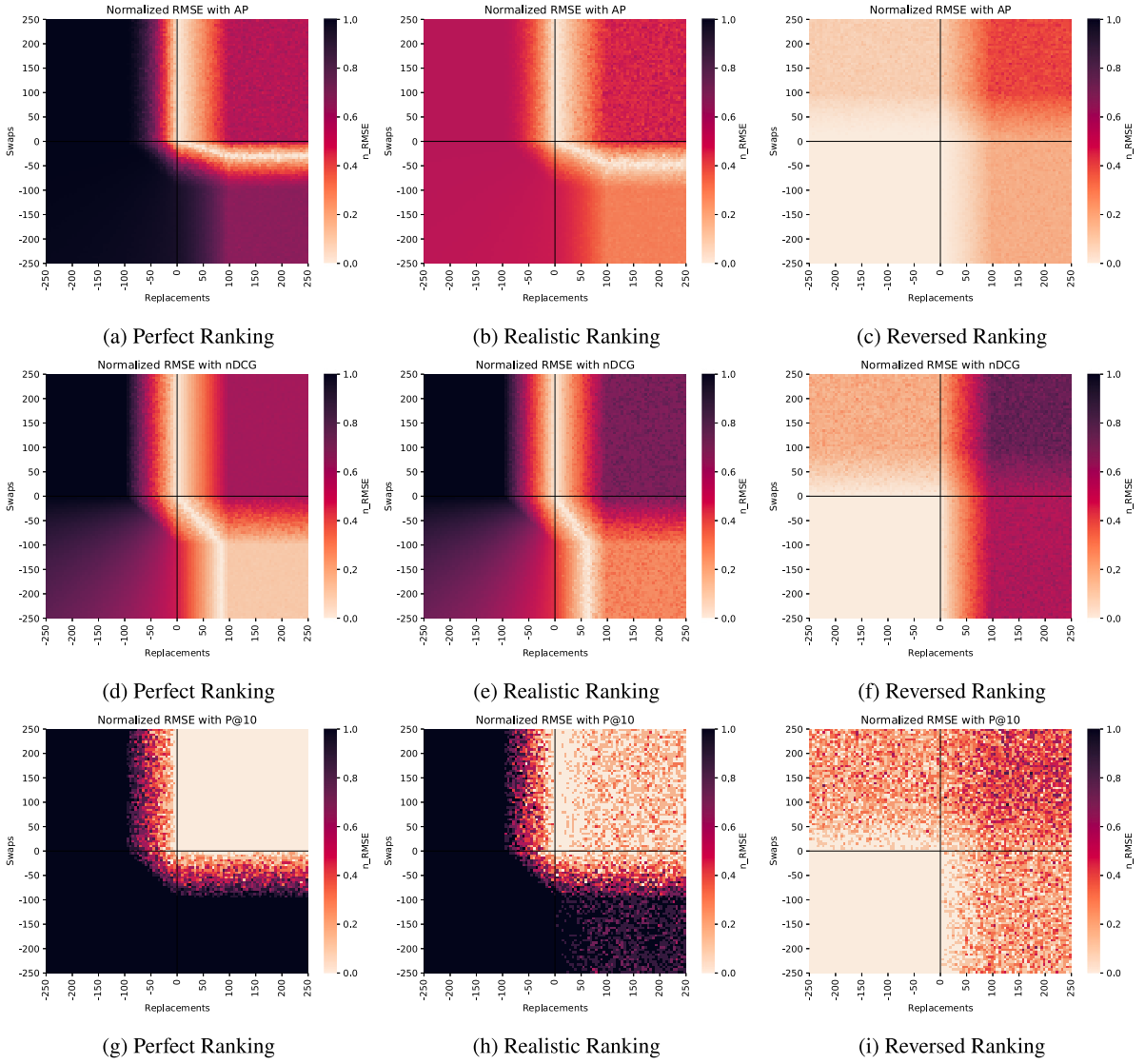


Fig. 16. Scenario 2 - nRMSE: Simulated runs with a single topic. Each run retrieves only half of the relevant documents (Recall = 0.5). The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

The main difference between KTU and RBO is due to RBO top-heaviness. This means that even less than 50 replacements or swaps can cause RBO drop to the lowest score, i.e., 0. This happens when swaps and replacements occur at the top of the ranking, because RBO is less sensitive to changes towards the end of the ranking. Therefore, an RBO score of 0.8, might denote a great overlap at the top of the ranking, but further analyses are needed when the whole ranking has to be reproduced.

RMSE, nRMSE and statistical tests depend on the underlying effectiveness measure, e.g., AP, nDCG, etc. Therefore, the reproducibility scores returned by these measures should be considered in combination with: (1) the type of effectiveness measures and its property; (2) the type of original run, the more extreme the run (very good or very bad performing run), the greater the impact in terms of reproducibility. For example, with RMSE, 50 replacements or swaps can increase RMSE up to: 0.5 with AP, 0.6 with nDCG, and 1 with P@10. If we consider only AP, 50 replacements or swaps can increase RMSE in a range from 0.5 (perfect ranking) to 0.2 (realistic ranking).

While nRMSE can mitigate the impact of the run type, not much can be done in terms of difference between measures, as they have different properties and they are intended to evaluate runs with different point of views. Top-heavy rank based measures (AP and nDCG) exhibit a region where negative swaps are counterbalanced by positive replacements, thus relatively low RMSE and nRMSE scores (lower than 0.1) or high p-values (greater than 0.8) might not be enough to conclude that a run was successfully reproduced. On the other side, RMSE, nRMSE and statistical tests with a set based measure as P@10, exhibit high variance, so a few swaps or replacements in the top 10 rank positions can cause very high scores. Finally, Table 2 summarizes our experimental results and provides an overview of the reproducibility measures and the corresponding use cases.

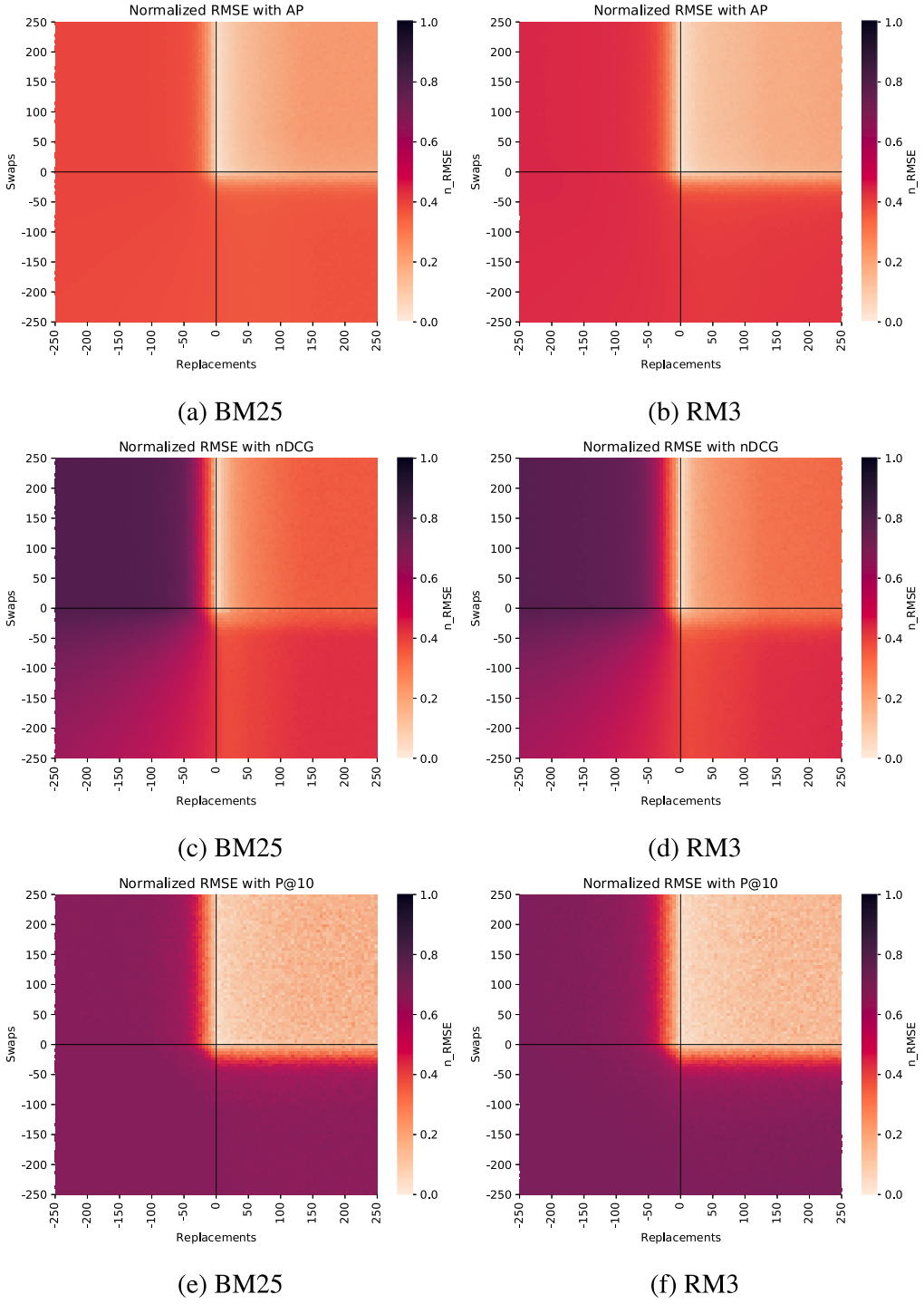


Fig. 17. Scenario 3 - nRMSE: BM25 and RM3 runs on TREC Common Core 2018. The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

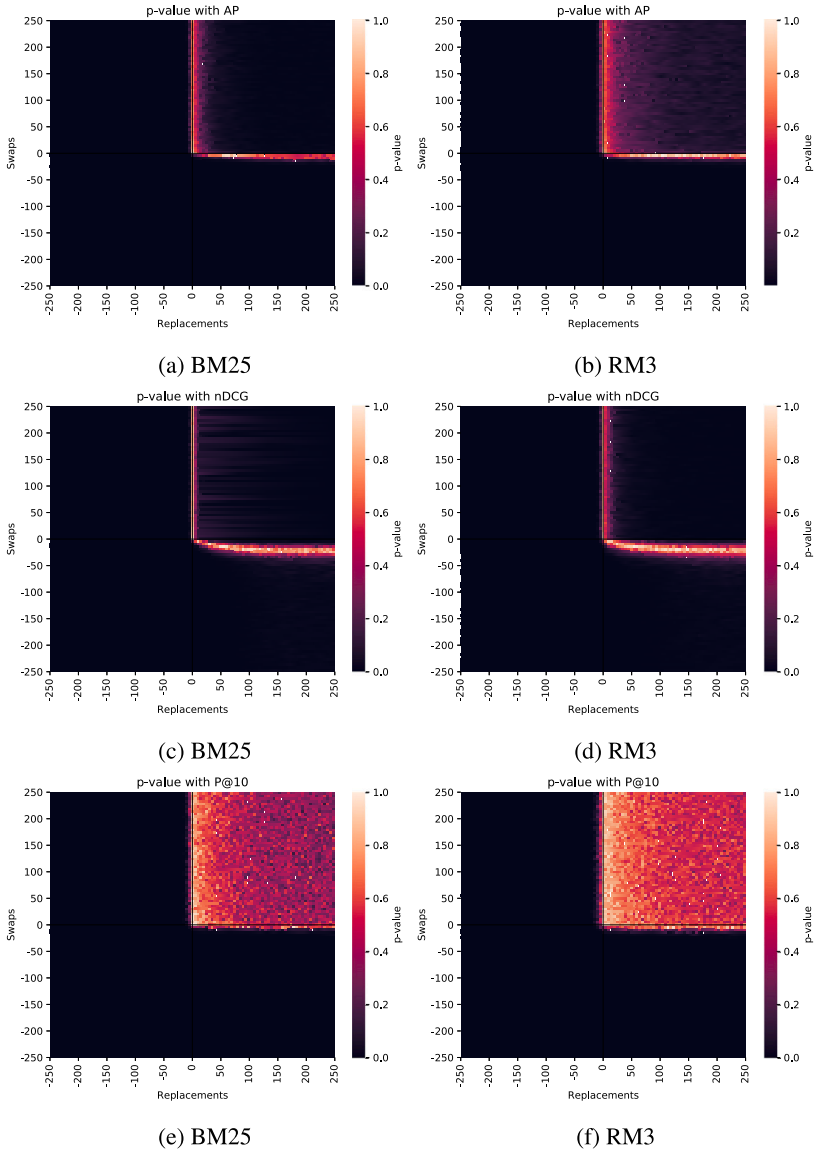


Fig. 18. Scenario 3 - p-value: BM25 and RM3 runs on TREC Common Core 2018. The runs are deteriorated with replacements in $[1, 500]$ and swaps from $[0, 500]$ to $[501, 1000]$.

5. Conclusions, limitations and future work

This paper presents an analysis of IR reproducibility measures. We consider rank based measures, i.e., KTU and RBO, which compare the original and reproduced lists of documents. As effectiveness based measures, we consider RMSE, which compares the difference in terms of IR effectiveness measures, e.g., AP, between the original and reproduced runs. We propose nRMSE, a normalized version of RMSE, to account for the relative difference instead of the absolute one. Finally, we consider statistical tests by looking at the *p*-value.

We propose a deterioration algorithm to deteriorate an input run and generate a reproduced run, where we can control the amount of deterioration. The deterioration algorithm exploits 2 operations: replacement of a retrieved document with a non retrieved one; swaps of two documents in the run.

We compute all reproducibility measures with the output of the deterioration algorithm. We propose 3 experimental scenarios with different types of runs: (1) synthetic runs with a single topic and recall = 1; (2) synthetic runs with a single topic and recall = 0.5; (3) real runs with 50 topics.

Once more, our experiments confirm that reproducibility is not a trivial problem. Not only it is hard to reproduce the results of a published paper, but it is even harder to decide when such results are reproduced to an acceptable extent. For example, given a reproducibility experiment with RMSE score equal to 0.05, we cannot conclude by solely looking at this score that reproducibility

Table 2
Summary and overview of the reproducibility measures. For each measure, the level of specificity and a corresponding use case are given. The experimental findings are summarized once again and examples of possible evaluation criteria are given.

Measure	Level of specificity	Use case
KTU	Document order	KTU should be used for the most specific level of measuring reproducibility. Our experimental results suggest that if the first stage ranking deteriorates, a re-ranker cannot compensate for the deterioration and recover the exact document ranking. **Use this measure for the highest degree of rigor.**
RBO	Document order	Similar to KTU, the RBO is determined by the document ranking. In comparison, it implies a user model and can be parameterized to put higher weights on top-ranked documents. As a result, it is less strict than KTU, i.e., if parameterized accordingly, it will result in higher scores, when there are overlaps between the top-ranked documents in the original and the reproduced run. Similar to KTU, our experiments show that a deteriorated reproduced first stage ranker cannot be compensated by an improved reproduced re-ranker in terms of RBO. **Use this measure when it is important to account for the correlation between the document rankings that likely only have a few overlaps.**
RMSE / nRMSE	Effectiveness	RMSE is determined by the distance between the topic score distributions. It is a document-agnostic reproducibility measure, i.e., a good or perfect reproduction does not depend on particular documents but on the corresponding relevance labels instead. It means that a perfect reproduction (RMSE=0) could be achieved with different documents that comply with the relevance labels in the original rankings. Our experimental results suggest that an improved reproduced re-ranker can indeed compensate a deteriorated first stage ranker as different document can result in the same performance scores at the topic-level. nRMSE is normalized by the worst possible ranking, which results in larger relative errors. **Use these measures, if it is less important to reproduce the exact document ordering but it is more important to reproduce the effectiveness without neglecting the score distribution over the topics.**
p-values	Statistical comparison	At the most general level, the topic score distributions can be compared by paired t-tests. The intuition follows the idea of low p-values indicating a higher probability of failing the reproduction. In turn, higher p-values indicate more similar topic score distributions. Our experimental results suggest that the p-values are very sensitive to deteriorations in the runs. **Use this approach, if it is important to account for the reproduction of topic score distributions beyond RMSE / nRMSE.**

is achieved. This happens because, even if our deterioration algorithm allows to control the amount of noise, we cannot completely disentangle the effects due to the type of run, amount of relevant documents, and the IR effectiveness measure. Reproducibility measures such as RMSE or statistical tests depend on the underlying effectiveness measure and on the performance of the original run. Our normalized version of RMSE can mitigate the variations due to the original run but the dependency on the IR effectiveness measure cannot be removed. Other measures, such as KTU and RBO do not depend on any effectiveness measure but are much stricter, as they require the same exact order of documents for each topic. Moreover, RBO top heaviness can be misleading, as a few changes in the very top of the ranking can cause a severe drop in its score.

Moreover, our reproducibility study is limited by the choice of the experimental collection and the number of topics for real run. It would be beneficial to validate our results with a larger number topics, however publicly available IR collections usually include 50 or fewer assessed documents. One exception is the TREC Million Query Track, which includes 1755 judged topics in 2007 (Allan et al., 2008), 782 in 2008 (Allan, Aslam, Pavlu, Kanoulas, & Carterette, 2009) and 684 in 2009 (Carterette, Pavlu, Fang, & Kanoulas, 2010). However, these collections are not good choices for our experimental set-up because: (1) shallow pools were used so there is only a small number of relevant documents per topic that can be swapped or replaced; (2) statistical approaches were used to select the documents to judge, therefore only special variations of AP should be used to account for the estimated relevance.

The present work is also limited by the number and type of operations considered by our deterioration algorithm. As future work, we plan to extend the deterioration algorithm in different directions: first, we can consider different types of operations to better mimic real reproducibility runs, for example, swaps and replacements with multi-graded relevance; second, we can consider different experimental set-ups, for example, focus the deterioration on smaller intervals at the top of the run instead of considering the first and second half. Moreover, we can extend the present work and consider other measures as Effect Ratio (ER) and other tasks as replicability or generalizability.

Finally, our study does not consider the user perspective and what is the impact of reproducibility errors on the user experience. We plan to conduct a user study to understand the impact of reproducibility on the final user, for example, what is the impact of replacing or swapping a document in terms of user behavior, e.g., clicks.

As for any evaluation task, the final remark is to use different evaluation measures of different categories, depending on the final goal and/or the application domain. Comparing scores averaged across topics is not enough to conclude that an experiment is successfully reproduced. As our experiments show, even very similar per-topic scores can correspond to different rankings. Therefore, together with the source code, we should consider the release of runs computed on publicly available datasets, following the idea of open runs (Voorhees et al., 2016) and, if possible, annotated in a standardized way (Breuer, Keller, & Schaer, 2022).

CRediT authorship contribution statement

Maria Maistro: Conceptualization, Methodology, Software, Writing, Project administration. **Timo Breuer:** Conceptualization, Methodology, Software, Writing. **Philipp Schaer:** Conceptualization, Methodology, Writing, Supervision. **Nicola Ferro:** Conceptualization, Methodology, Writing, Supervision.

Data availability

The paper contains a link to our code and data. The implementation of all our experiments is publicly available at https://github.com/irgroup/ipm-reproducibility. The GitHub repository includes instructions on how to run the code. The experiments in this paper are conducted only on publicly available data.

Acknowledgments

This paper was partially supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 893667 and the German Research Foundation (DFG) under project no. 407518790.

Appendix

See Tables 3–15.

Table 3
Scenario 1 - Kendall's τ Union.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8111	0.8111	0.8111	0.8111	0.8999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
200	0.8111	0.8111	0.8111	0.8111	0.8960	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	0.8111	0.8111	0.8111	0.8111	0.8970	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	0.8111	0.8111	0.8111	0.8111	0.9080	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
50	0.8111	0.8111	0.8111	0.8111	0.9086	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0	0.8111	0.8111	0.8111	0.8111	0.8945	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	0.8073	0.8128	0.7775	0.7987	0.7767	0.8788	0.8687	0.8979	0.8567	0.8919	0.8883
−100	0.7906	0.7842	0.7617	0.7904	0.7870	0.7600	0.7484	0.7424	0.7601	0.7563	0.7420
−150	0.7805	0.7993	0.7941	0.7849	0.7623	0.7123	0.7658	0.7461	0.7520	0.7504	0.7555
−200	0.7596	0.7828	0.7680	0.7472	0.7452	0.7585	0.7570	0.7528	0.7621	0.7365	0.7641
−250	0.7954	0.7587	0.7841	0.7740	0.7393	0.7559	0.7300	0.7640	0.7393	0.7409	0.7598
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8111	0.8111	0.8111	0.8111	0.8919	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
200	0.8111	0.8111	0.8111	0.8111	0.9063	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	0.8111	0.8111	0.8111	0.8111	0.9010	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	0.8111	0.8111	0.8111	0.8111	0.8980	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
50	0.8111	0.8111	0.8111	0.8111	0.9012	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0	0.8111	0.8111	0.8111	0.8111	0.8965	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	0.7950	0.8016	0.7894	0.7954	0.7980	0.8674	0.8960	0.8679	0.8636	0.8520	0.8708
−100	0.7737	0.7802	0.8003	0.7927	0.7591	0.7664	0.7633	0.7552	0.7466	0.7572	0.7718
−150	0.7645	0.7665	0.7887	0.7906	0.7722	0.7608	0.7590	0.7508	0.7154	0.7459	0.7613
−200	0.7831	0.7691	0.7653	0.7775	0.7669	0.7518	0.7528	0.7787	0.7189	0.7635	0.7370
−250	0.7806	0.7886	0.7418	0.7756	0.7717	0.7413	0.7417	0.7313	0.7297	0.7394	0.7337
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.7742	0.7309	0.7748	0.7315	0.7550	0.7478	0.7344	0.7434	0.7357	0.7680	0.7834
200	0.7844	0.7632	0.7569	0.7313	0.7520	0.7141	0.7537	0.7704	0.7889	0.7692	0.7485
150	0.7520	0.7710	0.7232	0.7365	0.7150	0.7407	0.7535	0.7393	0.7650	0.7487	0.7660
100	0.7470	0.7612	0.7613	0.7794	0.7306	0.7654	0.7411	0.7658	0.7626	0.7188	0.7636
50	0.8880	0.8860	0.8943	0.8582	0.8777	0.8807	0.8622	0.8954	0.8702	0.8724	0.8616
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−100	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−250	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 4
Scenario 2 - Kendall's τ Union.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8111	0.8111	0.8111	0.8111	0.8980	1.0000	0.9314	0.8732	0.8722	0.8666	0.8609
200	0.8111	0.8111	0.8111	0.8111	0.9064	1.0000	0.9440	0.8681	0.8636	0.8711	0.8625
150	0.8111	0.8111	0.8111	0.8111	0.8977	1.0000	0.9277	0.8656	0.8798	0.8773	0.8680
100	0.8111	0.8111	0.8111	0.8111	0.9005	1.0000	0.9279	0.8726	0.8720	0.8688	0.8801
50	0.8111	0.8111	0.8111	0.8111	0.9069	1.0000	0.9370	0.8626	0.8799	0.8636	0.8815
0	0.8111	0.8111	0.8111	0.8111	0.9015	1.0000	0.9304	0.8786	0.8676	0.8753	0.8630
−50	0.8139	0.8001	0.7983	0.7949	0.7714	0.8623	0.8044	0.7571	0.7414	0.7641	0.7504
−100	0.8015	0.7782	0.7750	0.7986	0.7494	0.7608	0.6991	0.6419	0.6315	0.6403	0.6350
−150	0.7984	0.7728	0.7898	0.7638	0.7744	0.7590	0.6755	0.6321	0.6501	0.6626	0.6549
−200	0.8066	0.7846	0.7705	0.7606	0.7419	0.7519	0.7112	0.6341	0.6437	0.6160	0.6303
−250	0.7900	0.7691	0.7613	0.7636	0.7881	0.7590	0.6790	0.6495	0.6395	0.6409	0.6587
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8111	0.8111	0.8111	0.8111	0.8971	1.0000	0.9392	0.8754	0.8754	0.8890	0.8669
200	0.8111	0.8111	0.8111	0.8111	0.9040	1.0000	0.9443	0.8848	0.8813	0.8611	0.8716
150	0.8111	0.8111	0.8111	0.8111	0.8865	1.0000	0.9409	0.8736	0.8695	0.8782	0.8817
100	0.8111	0.8111	0.8111	0.8111	0.8960	1.0000	0.9281	0.8789	0.8709	0.8707	0.8614
50	0.8111	0.8111	0.8111	0.8111	0.9049	1.0000	0.9392	0.8751	0.8741	0.8720	0.8911
0	0.8111	0.8111	0.8111	0.8111	0.9014	1.0000	0.9367	0.8900	0.8749	0.8704	0.8702
−50	0.7904	0.8066	0.8051	0.7825	0.7717	0.8628	0.8107	0.7404	0.7389	0.7578	0.7449
−100	0.7733	0.7880	0.7580	0.7889	0.7812	0.7471	0.7050	0.6265	0.6453	0.6081	0.6622
−150	0.7876	0.7643	0.7872	0.7567	0.7471	0.7265	0.6961	0.6154	0.6384	0.6526	0.6489
−200	0.7946	0.7828	0.7857	0.7710	0.7743	0.7269	0.7127	0.6537	0.6439	0.6539	0.6399
−250	0.7764	0.7755	0.7564	0.7763	0.7668	0.7508	0.6530	0.6034	0.6292	0.6373	0.6312
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.7541	0.7523	0.7639	0.7216	0.7618	0.7456	0.7137	0.6144	0.6638	0.6627	0.6028
200	0.7324	0.7825	0.7722	0.7823	0.7293	0.7719	0.7218	0.6472	0.6217	0.6439	0.6756
150	0.7235	0.7580	0.7264	0.7581	0.7361	0.7146	0.6724	0.6630	0.6202	0.6542	0.6520
100	0.7769	0.7483	0.7540	0.7514	0.7632	0.7547	0.7430	0.6458	0.6263	0.6268	0.6445
50	0.8974	0.8777	0.8777	0.8922	0.8530	0.8766	0.8012	0.7428	0.7727	0.7327	0.7562
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9410	0.8760	0.8736	0.8728	0.8714
−50	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9376	0.8704	0.8879	0.8796	0.8695
−100	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9362	0.8803	0.8679	0.8596	0.8700
−150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9465	0.8823	0.8781	0.8850	0.8835
−200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9460	0.8689	0.8786	0.8799	0.8631
−250	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9451	0.8702	0.8773	0.8630	0.8689

Table 5
Scenario 3 - Kendall's τ Union.

						BM25					
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.9018	0.9023	0.9031	0.9049	0.9159	0.9781	0.9508	0.9439	0.9398	0.9378	0.9396
200	0.9026	0.9025	0.9023	0.9046	0.9145	0.9780	0.9535	0.9441	0.9393	0.9407	0.9402
150	0.9029	0.9027	0.9027	0.9037	0.9144	0.9792	0.9516	0.9421	0.9394	0.9382	0.9407
100	0.9025	0.9024	0.9029	0.9042	0.9147	0.9789	0.9519	0.9421	0.9400	0.9402	0.9400
50	0.9042	0.9044	0.9044	0.9065	0.9162	0.9806	0.9546	0.9448	0.9404	0.9419	0.9403
0	0.9227	0.9227	0.9227	0.9250	0.9361	1.0000	0.9735	0.9644	0.9588	0.9613	0.9604
−50	0.9175	0.9175	0.9146	0.9130	0.9123	0.9132	0.8856	0.8731	0.8753	0.8742	0.8736
−100	0.9142	0.9133	0.9117	0.9098	0.9024	0.8955	0.8746	0.8657	0.8599	0.8568	0.8588
−150	0.9110	0.9073	0.9083	0.9059	0.9042	0.8943	0.8700	0.8623	0.8590	0.8569	0.8584
−200	0.9100	0.9075	0.9069	0.9022	0.9000	0.8931	0.8698	0.8621	0.8553	0.8529	0.8558
−250	0.9068	0.9064	0.9042	0.9014	0.8997	0.8893	0.8696	0.8614	0.8549	0.8560	0.8564
						RM3					
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8976	0.8972	0.8993	0.9022	0.9170	0.9805	0.9599	0.9514	0.9499	0.9477	0.9459
200	0.8987	0.8990	0.8985	0.9034	0.9182	0.9799	0.9585	0.9529	0.9483	0.9467	0.9474
150	0.8974	0.8985	0.8998	0.9030	0.9180	0.9801	0.9567	0.9527	0.9484	0.9484	0.9462
100	0.8978	0.8979	0.8980	0.9022	0.9175	0.9808	0.9596	0.9527	0.9484	0.9479	0.9463
50	0.8980	0.8999	0.9012	0.9029	0.9179	0.9814	0.9601	0.9530	0.9494	0.9489	0.9478
0	0.9158	0.9167	0.9182	0.9215	0.9366	1.0000	0.9779	0.9723	0.9687	0.9669	0.9668
−50	0.9106	0.9089	0.9083	0.9058	0.9052	0.9130	0.8925	0.8869	0.8853	0.8824	0.8798
−100	0.9036	0.9049	0.9021	0.8974	0.8948	0.8942	0.8695	0.8632	0.8618	0.8570	0.8566
−150	0.9040	0.9008	0.8966	0.8938	0.8895	0.8845	0.8663	0.8602	0.8529	0.8537	0.8527
−200	0.8993	0.8990	0.8946	0.8944	0.8866	0.8849	0.8627	0.8565	0.8546	0.8501	0.8497
−250	0.8962	0.8958	0.8930	0.8920	0.8881	0.8821	0.8583	0.8586	0.8492	0.8529	0.8475

Table 6
Scenario 1 - RBO.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0000	0.0000	0.0000	0.0000	0.7455	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
200	0.0000	0.0000	0.0000	0.0000	0.3113	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	0.0000	0.0000	0.0000	0.0000	0.3262	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	0.0000	0.0000	0.0000	0.0000	0.6531	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
50	0.0000	0.0000	0.0000	0.0000	0.3277	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0	0.0000	0.0000	0.0000	0.0000	0.8534	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.5425	0.3060	0.1665	0.2700	0.5632	0.8393
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5636	0.5636	0.5636	0.5636	0.7925	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
200	0.5636	0.5636	0.5636	0.5636	0.6542	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	0.5636	0.5636	0.5636	0.5636	0.6853	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	0.5636	0.5636	0.5636	0.5636	0.9435	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
50	0.5636	0.5636	0.5636	0.5636	0.6645	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0	0.5636	0.5636	0.5636	0.5636	0.8030	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	0.5636	0.5636	0.5636	0.5636	0.5636	0.6498	0.8701	0.8738	0.7951	0.7178	0.7020
−100	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636
−150	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636
−200	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636
−250	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8188	0.8773	0.3387	0.9764	0.4923	0.8383	0.7831	0.7705	0.9413	0.9572	0.8120
200	0.9997	0.7226	0.7720	0.8590	0.5825	0.9749	0.9990	0.8643	0.4903	0.9378	0.9456
150	0.7834	0.9771	0.7591	0.9128	0.8126	0.9801	0.9354	0.7729	0.9420	0.3013	0.6279
100	0.7404	0.3980	0.8966	0.8811	0.7819	0.7901	0.4352	0.9234	0.5900	0.8156	0.9976
50	0.9949	0.8734	0.9999	0.9203	0.9974	0.9507	0.4129	0.9997	0.9804	0.9893	0.9773
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−100	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−250	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 7
Scenario 2 - RBO.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0000	0.0000	0.0000	0.0000	0.2975	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
200	0.0000	0.0000	0.0000	0.0000	0.2957	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	0.0000	0.0000	0.0000	0.0000	0.6763	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	0.0000	0.0000	0.0000	0.0000	0.1627	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
50	0.0000	0.0000	0.0000	0.0000	0.6891	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0	0.0000	0.0000	0.0000	0.0000	0.5992	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.4035	0.3270	0.9357	0.2121	0.2886	0.1245
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5636	0.5636	0.5636	0.5636	0.9444	1.0000	0.9999	0.8776	0.5976	1.0000	0.5969
200	0.5636	0.5636	0.5636	0.5636	0.8307	1.0000	0.5972	0.9993	0.9995	0.9995	0.8770
150	0.5636	0.5636	0.5636	0.5636	0.8918	1.0000	0.9999	0.5972	0.8776	0.9995	0.9999
100	0.5636	0.5636	0.5636	0.5636	0.9394	1.0000	0.5972	0.5970	0.9617	1.0000	0.5970
50	0.5636	0.5636	0.5636	0.5636	0.7027	1.0000	0.9995	0.9623	0.5976	0.9989	1.0000
0	0.5636	0.5636	0.5636	0.5636	0.6544	1.0000	1.0000	0.9618	0.9995	1.0000	0.8770
−50	0.5636	0.5636	0.5636	0.5636	0.5636	0.8806	0.8363	0.6129	0.7744	0.6844	0.3930
−100	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5630	0.4408	0.1235	0.1235	0.4413
−150	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5259	0.5258	0.4413	0.4412
−200	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5253	0.4413	0.0388
−250	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5636	0.5635	0.1607	0.5253	0.1608
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.8776	0.6695	0.7280	0.9956	0.8717	0.9730	0.6580	0.4023	0.8390	0.8775	0.2124
200	0.8158	0.9665	0.3960	0.9657	0.5963	0.9392	0.9366	0.1607	0.1553	0.3721	0.4337
150	0.9837	0.7668	0.9194	0.7405	0.8697	0.3151	0.7288	0.3326	0.5141	0.2673	0.3118
100	0.9565	0.7962	0.9399	0.8605	0.4223	0.5476	0.9291	0.9208	0.9862	0.4508	0.8407
50	0.8161	0.9719	0.9694	0.8661	0.5923	0.5714	0.8412	0.6009	0.8974	0.8458	0.8022
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8774	0.7944	0.8728	0.9198	0.5297
−50	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.8735	0.9184	0.6583	0.9466
−100	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9732	0.8367	0.7724	0.6690	0.9655
−150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.7943	0.7836	0.4961	0.5684	0.5923
−200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9453	0.9137	0.9553	0.9650	0.8105
−250	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9923	0.9456	0.3702	0.9185	0.9215

Table 8
Scenario 3 - RBO.

	BM25										
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.4545	0.4661	0.4574	0.4709	0.5074	0.9932	0.9743	0.9534	0.9485	0.9491	0.9495
200	0.4677	0.4675	0.4533	0.4613	0.5241	0.9775	0.9718	0.9266	0.9425	0.9205	0.9515
150	0.4606	0.4708	0.4590	0.4716	0.5123	0.9897	0.9626	0.9645	0.9659	0.9501	0.9585
100	0.4596	0.4650	0.4504	0.4849	0.5085	0.9886	0.9817	0.9396	0.9502	0.9493	0.9619
50	0.4590	0.4641	0.4590	0.4714	0.5024	0.9885	0.9740	0.9587	0.9607	0.9417	0.9699
0	0.4713	0.4713	0.4713	0.4741	0.5206	1.0000	0.9829	0.9506	0.9448	0.9715	0.9638
−50	0.4713	0.4713	0.4713	0.4713	0.4817	0.5237	0.5010	0.4685	0.5018	0.4883	0.4926
−100	0.4713	0.4713	0.4713	0.4713	0.4713	0.4845	0.4677	0.4644	0.4341	0.4295	0.4527
−150	0.4713	0.4713	0.4713	0.4713	0.4713	0.4713	0.4406	0.4347	0.4306	0.4109	0.4404
−200	0.4713	0.4713	0.4713	0.4713	0.4713	0.4713	0.4213	0.4398	0.4256	0.4404	0.4201
−250	0.4713	0.4713	0.4713	0.4713	0.4713	0.4713	0.4421	0.4249	0.4489	0.4059	0.4110
	RM3										
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5635	0.5699	0.5578	0.5959	0.6418	0.9928	0.9706	0.9627	0.9481	0.9332	0.9562
200	0.5598	0.5605	0.5642	0.5779	0.6430	0.9884	0.9649	0.9636	0.9426	0.9555	0.9703
150	0.5600	0.5593	0.5714	0.5682	0.6277	0.9940	0.9456	0.9720	0.9515	0.9447	0.9421
100	0.5617	0.5638	0.5571	0.5843	0.6443	0.9980	0.9775	0.9692	0.9536	0.9550	0.9492
50	0.5621	0.5676	0.5621	0.5579	0.6362	0.9830	0.9685	0.9812	0.9540	0.9604	0.9574
0	0.5652	0.5671	0.5798	0.5797	0.6643	1.0000	0.9892	0.9730	0.9692	0.9715	0.9489
−50	0.5652	0.5652	0.5697	0.5673	0.5821	0.6581	0.6426	0.6524	0.5867	0.6088	0.6074
−100	0.5652	0.5652	0.5652	0.5679	0.5738	0.5885	0.5751	0.5643	0.5467	0.5271	0.5232
−150	0.5652	0.5652	0.5652	0.5652	0.5655	0.5700	0.5462	0.5418	0.5469	0.5499	0.5218
−200	0.5652	0.5652	0.5652	0.5652	0.5652	0.5704	0.5389	0.5389	0.5303	0.5294	0.5087
−250	0.5652	0.5652	0.5652	0.5652	0.5652	0.5652	0.5514	0.5188	0.5184	0.5147	0.5220

Table 9
Scenario 1 - RMSE - AP.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	1.0000	1.0000	1.0000	1.0000	0.7254	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
200	1.0000	1.0000	1.0000	1.0000	0.7503	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
150	1.0000	1.0000	1.0000	1.0000	0.7160	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
100	1.0000	1.0000	1.0000	1.0000	0.6961	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	1.0000	1.0000	1.0000	1.0000	0.7262	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	1.0000	1.0000	1.0000	1.0000	0.7179	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.9981	0.9973	0.9960	0.9931	0.9845	0.7123	0.6874	0.7077	0.7154	0.6881	0.6745
−100	0.9945	0.9933	0.9902	0.9846	0.9708	0.9389	0.9356	0.9373	0.9383	0.9387	0.9373
−150	0.9909	0.9887	0.9840	0.9778	0.9658	0.9381	0.9378	0.9377	0.9366	0.9382	0.9381
−200	0.9883	0.9843	0.9801	0.9714	0.9602	0.9369	0.9365	0.9387	0.9385	0.9394	0.9381
−250	0.9840	0.9807	0.9761	0.9692	0.9567	0.9370	0.9398	0.9379	0.9361	0.9382	0.9368
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.7081	0.7081	0.7081	0.7081	0.5415	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
200	0.7081	0.7081	0.7081	0.7081	0.5436	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
150	0.7081	0.7081	0.7081	0.7081	0.5263	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
100	0.7081	0.7081	0.7081	0.7081	0.5053	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	0.7081	0.7081	0.7081	0.7081	0.5310	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	0.7081	0.7081	0.7081	0.7081	0.5355	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.7063	0.7054	0.7041	0.7010	0.6916	0.4828	0.4842	0.4502	0.4925	0.4668	0.4947
−100	0.7029	0.7012	0.6982	0.6929	0.6818	0.6447	0.6459	0.6449	0.6464	0.6451	0.6456
−150	0.6991	0.6964	0.6917	0.6857	0.6727	0.6462	0.6441	0.6474	0.6460	0.6459	0.6454
−200	0.6963	0.6926	0.6877	0.6795	0.6689	0.6455	0.6444	0.6458	0.6441	0.6458	0.6472
−250	0.6929	0.6893	0.6833	0.6765	0.6651	0.6435	0.6447	0.6467	0.6464	0.6456	0.6455
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.1508	0.1528	0.1656	0.1377	0.1981	0.1799	0.1989	0.1733	0.1402	0.1310	0.1512
200	0.1329	0.1600	0.1898	0.1439	0.1658	0.1603	0.1587	0.1701	0.1851	0.1647	0.1493
150	0.1487	0.1277	0.1392	0.1435	0.1577	0.1420	0.1490	0.1712	0.1680	0.1564	0.2059
100	0.1496	0.1741	0.1398	0.1439	0.1770	0.1408	0.1593	0.1383	0.1438	0.1763	0.1731
50	0.0380	0.0473	0.0315	0.0347	0.0363	0.0460	0.0741	0.0381	0.0335	0.0397	0.0440
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 10
Scenario 2 - RMSE - AP.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5000	0.5000	0.5000	0.5000	0.3709	0.0000	0.1352	0.2692	0.2808	0.2722	0.2653
200	0.5000	0.5000	0.5000	0.5000	0.3670	0.0000	0.1050	0.2602	0.2614	0.2804	0.2819
150	0.5000	0.5000	0.5000	0.5000	0.3717	0.0000	0.1114	0.2650	0.2731	0.2739	0.2747
100	0.5000	0.5000	0.5000	0.5000	0.3819	0.0000	0.1186	0.2721	0.2845	0.2861	0.2834
50	0.5000	0.5000	0.5000	0.5000	0.3707	0.0000	0.1122	0.2826	0.2905	0.2764	0.2730
0	0.5000	0.5000	0.5000	0.5000	0.3743	0.0000	0.1192	0.2623	0.2970	0.2879	0.2771
−50	0.4990	0.4986	0.4980	0.4965	0.4918	0.3507	0.2654	0.1085	0.1284	0.1476	0.1411
−100	0.4970	0.4965	0.4948	0.4921	0.4860	0.4691	0.4146	0.3201	0.3268	0.3232	0.3154
−150	0.4955	0.4943	0.4918	0.4881	0.4826	0.4687	0.4158	0.3226	0.3249	0.3279	0.3209
−200	0.4940	0.4921	0.4899	0.4858	0.4799	0.4691	0.4159	0.3283	0.3238	0.3223	0.3252
−250	0.4923	0.4902	0.4881	0.4843	0.4792	0.4686	0.4119	0.3279	0.3232	0.3273	0.3266
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.3541	0.3541	0.3541	0.3541	0.2633	0.0000	0.1287	0.3110	0.2990	0.2919	0.3258
200	0.3541	0.3541	0.3541	0.3541	0.2549	0.0000	0.1495	0.2904	0.2961	0.2972	0.3254
150	0.3541	0.3541	0.3541	0.3541	0.2597	0.0000	0.1287	0.3087	0.3091	0.3136	0.3050
100	0.3541	0.3541	0.3541	0.3541	0.2610	0.0000	0.1592	0.3460	0.3302	0.2807	0.3276
50	0.3541	0.3541	0.3541	0.3541	0.2587	0.0000	0.1356	0.3067	0.3047	0.2917	0.2769
0	0.3541	0.3541	0.3541	0.3541	0.2704	0.0000	0.1247	0.2997	0.3000	0.3102	0.3158
−50	0.3531	0.3527	0.3519	0.3506	0.3460	0.2455	0.1412	0.0410	0.0250	0.0294	0.0325
−100	0.3514	0.3503	0.3489	0.3464	0.3404	0.3235	0.2683	0.1694	0.1671	0.1653	0.1782
−150	0.3494	0.3482	0.3463	0.3425	0.3367	0.3233	0.2688	0.1758	0.1662	0.1664	0.1737
−200	0.3477	0.3460	0.3438	0.3402	0.3339	0.3224	0.2691	0.1822	0.1798	0.1694	0.1680
−250	0.3464	0.3442	0.3420	0.3379	0.3327	0.3231	0.2669	0.1787	0.1748	0.1716	0.1659
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0658	0.0880	0.0762	0.0600	0.0718	0.0657	0.2210	0.4197	0.3361	0.3639	0.4014
200	0.0851	0.0626	0.1092	0.0740	0.0675	0.0760	0.1882	0.4067	0.3932	0.4081	0.3685
150	0.0779	0.0836	0.0734	0.0713	0.0660	0.0816	0.2166	0.4048	0.3835	0.3575	0.4052
100	0.0925	0.0821	0.0888	0.0762	0.0966	0.0850	0.1804	0.3472	0.3809	0.4038	0.4272
50	0.0369	0.0186	0.0232	0.0264	0.0307	0.0251	0.1081	0.2836	0.2470	0.2418	0.2314
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0561	0.1423	0.1536	0.1603	0.1739
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0543	0.1720	0.1443	0.1664	0.1565
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0517	0.1471	0.1662	0.1776	0.1550
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0568	0.1541	0.1578	0.1749	0.1720
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0566	0.1557	0.1477	0.1650	0.1596
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0506	0.1461	0.1739	0.1454	0.1641

Table 11
Scenario 3 - RMSE - AP.

						BM25					
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.3126	0.3126	0.3124	0.3110	0.3013	0.0262	0.1012	0.1449	0.1666	0.1697	0.1699
200	0.3127	0.3125	0.3124	0.3108	0.3001	0.0279	0.1034	0.1458	0.1684	0.1768	0.1678
150	0.3125	0.3126	0.3125	0.3112	0.3003	0.0254	0.1107	0.1412	0.1609	0.1677	0.1717
100	0.3126	0.3126	0.3126	0.3109	0.2996	0.0252	0.1047	0.1470	0.1659	0.1685	0.1662
50	0.3125	0.3125	0.3122	0.3112	0.3015	0.0227	0.1017	0.1400	0.1618	0.1616	0.1636
0	0.3135	0.3135	0.3135	0.3126	0.3029	0.0000	0.0803	0.1138	0.1346	0.1340	0.1368
−50	0.3124	0.3121	0.3115	0.3104	0.3062	0.2837	0.2745	0.2785	0.2809	0.2836	0.2839
−100	0.3110	0.3104	0.3092	0.3075	0.3036	0.2928	0.2823	0.2839	0.2884	0.2901	0.2896
−150	0.3097	0.3088	0.3074	0.3053	0.3013	0.2933	0.2832	0.2849	0.2899	0.2896	0.2894
−200	0.3086	0.3074	0.3059	0.3036	0.2998	0.2933	0.2827	0.2845	0.2893	0.2893	0.2902
−250	0.3074	0.3062	0.3047	0.3023	0.2990	0.2933	0.2831	0.2851	0.2881	0.2906	0.2897
						RM3					
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5379	0.5379	0.5372	0.5318	0.4984	0.0144	0.1726	0.2133	0.2345	0.2390	0.2393
200	0.5381	0.5374	0.5358	0.5318	0.4980	0.0165	0.1703	0.2157	0.2367	0.2416	0.2349
150	0.5369	0.5375	0.5368	0.5322	0.4992	0.0161	0.1791	0.2179	0.2304	0.2365	0.2389
100	0.5371	0.5371	0.5374	0.5326	0.4939	0.0142	0.1717	0.2150	0.2341	0.2368	0.2322
50	0.5368	0.5368	0.5355	0.5317	0.5003	0.0129	0.1682	0.2101	0.2333	0.2349	0.2356
0	0.5438	0.5438	0.5438	0.5382	0.5048	0.0000	0.1621	0.2054	0.2257	0.2247	0.2276
−50	0.5039	0.4929	0.4805	0.4601	0.4135	0.2836	0.2618	0.2833	0.2903	0.2955	0.2973
−100	0.4715	0.4614	0.4416	0.4180	0.3778	0.2950	0.2670	0.2821	0.2999	0.3023	0.3002
−150	0.4496	0.4364	0.4170	0.3931	0.3563	0.2963	0.2673	0.2853	0.2999	0.2995	0.2982
−200	0.4327	0.4166	0.4018	0.3776	0.3440	0.2965	0.2687	0.2845	0.2994	0.2989	0.3025
−250	0.4174	0.4034	0.3887	0.3667	0.3366	0.2963	0.2691	0.2873	0.2965	0.3035	0.3002

Table 12
Scenario 1 - nRMSE - AP.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	1.0000	1.0000	1.0000	1.0000	0.7254	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
200	1.0000	1.0000	1.0000	1.0000	0.7503	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
150	1.0000	1.0000	1.0000	1.0000	0.7160	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
100	1.0000	1.0000	1.0000	1.0000	0.6961	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	1.0000	1.0000	1.0000	1.0000	0.7262	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	1.0000	1.0000	1.0000	1.0000	0.7179	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.9981	0.9973	0.9960	0.9931	0.9845	0.7123	0.6874	0.7077	0.7154	0.6881	0.6745
−100	0.9945	0.9933	0.9902	0.9846	0.9708	0.9389	0.9356	0.9373	0.9383	0.9387	0.9373
−150	0.9909	0.9887	0.9840	0.9778	0.9658	0.9381	0.9378	0.9377	0.9366	0.9382	0.9381
−200	0.9883	0.9843	0.9801	0.9714	0.9602	0.9369	0.9365	0.9387	0.9385	0.9394	0.9381
−250	0.9840	0.9807	0.9761	0.9692	0.9567	0.9370	0.9398	0.9379	0.9361	0.9382	0.9368
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	1.0000	1.0000	1.0000	1.0000	0.7647	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
200	1.0000	1.0000	1.0000	1.0000	0.7677	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
150	1.0000	1.0000	1.0000	1.0000	0.7433	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
100	1.0000	1.0000	1.0000	1.0000	0.7136	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	1.0000	1.0000	1.0000	1.0000	0.7499	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	1.0000	1.0000	1.0000	1.0000	0.7563	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.9975	0.9962	0.9943	0.9900	0.9767	0.6817	0.6838	0.6358	0.6955	0.6593	0.6986
−100	0.9927	0.9902	0.9861	0.9785	0.9629	0.9105	0.9122	0.9107	0.9128	0.9111	0.9117
−150	0.9873	0.9835	0.9768	0.9683	0.9501	0.9126	0.9095	0.9143	0.9123	0.9122	0.9115
−200	0.9833	0.9782	0.9712	0.9596	0.9446	0.9115	0.9100	0.9121	0.9096	0.9120	0.9139
−250	0.9785	0.9735	0.9650	0.9554	0.9393	0.9087	0.9105	0.9133	0.9128	0.9118	0.9116
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.1591	0.1612	0.1748	0.1453	0.2090	0.1898	0.2098	0.1828	0.1479	0.1383	0.1595
200	0.1402	0.1688	0.2003	0.1519	0.1749	0.1691	0.1674	0.1794	0.1953	0.1738	0.1576
150	0.1569	0.1347	0.1468	0.1514	0.1663	0.1498	0.1572	0.1806	0.1772	0.1650	0.2172
100	0.1579	0.1837	0.1475	0.1519	0.1867	0.1486	0.1681	0.1459	0.1518	0.1860	0.1827
50	0.0401	0.0499	0.0333	0.0366	0.0383	0.0485	0.0782	0.0402	0.0353	0.0419	0.0464
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 13
Scenario 2 - nRMSE - AP.

Perfect ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	1.0000	1.0000	1.0000	1.0000	0.7419	0.0000	0.2704	0.5384	0.5615	0.5444	0.5307
200	1.0000	1.0000	1.0000	1.0000	0.7340	0.0000	0.2101	0.5203	0.5228	0.5607	0.5637
150	1.0000	1.0000	1.0000	1.0000	0.7434	0.0000	0.2228	0.5301	0.5463	0.5477	0.5493
100	1.0000	1.0000	1.0000	1.0000	0.7639	0.0000	0.2372	0.5443	0.5690	0.5722	0.5667
50	1.0000	1.0000	1.0000	1.0000	0.7413	0.0000	0.2244	0.5652	0.5810	0.5527	0.5460
0	1.0000	1.0000	1.0000	1.0000	0.7486	0.0000	0.2384	0.5245	0.5939	0.5759	0.5541
−50	0.9980	0.9973	0.9961	0.9931	0.9836	0.7015	0.5308	0.2170	0.2568	0.2953	0.2821
−100	0.9940	0.9931	0.9895	0.9841	0.9719	0.9382	0.8293	0.6402	0.6537	0.6464	0.6308
−150	0.9909	0.9885	0.9837	0.9762	0.9651	0.9373	0.8316	0.6451	0.6498	0.6558	0.6418
−200	0.9881	0.9842	0.9798	0.9716	0.9599	0.9381	0.8318	0.6567	0.6476	0.6447	0.6504
−250	0.9846	0.9805	0.9762	0.9686	0.9583	0.9372	0.8238	0.6559	0.6465	0.6546	0.6532
Realistic ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.5481	0.5481	0.5481	0.5481	0.4076	0.0000	0.1992	0.4814	0.4629	0.4518	0.5044
200	0.5481	0.5481	0.5481	0.5481	0.3947	0.0000	0.2315	0.4495	0.4583	0.4601	0.5038
150	0.5481	0.5481	0.5481	0.5481	0.4020	0.0000	0.1992	0.4780	0.4785	0.4855	0.4722
100	0.5481	0.5481	0.5481	0.5481	0.4040	0.0000	0.2465	0.5356	0.5112	0.4345	0.5072
50	0.5481	0.5481	0.5481	0.5481	0.4005	0.0000	0.2099	0.4748	0.4717	0.4516	0.4287
0	0.5481	0.5481	0.5481	0.5481	0.4185	0.0000	0.1930	0.4640	0.4645	0.4802	0.4889
−50	0.5466	0.5460	0.5448	0.5428	0.5357	0.3801	0.2186	0.0635	0.0387	0.0454	0.0503
−100	0.5440	0.5424	0.5402	0.5362	0.5270	0.5008	0.4154	0.2622	0.2587	0.2558	0.2759
−150	0.5408	0.5391	0.5361	0.5302	0.5213	0.5006	0.4162	0.2721	0.2573	0.2576	0.2690
−200	0.5383	0.5357	0.5323	0.5267	0.5169	0.4991	0.4166	0.2821	0.2784	0.2623	0.2600
−250	0.5363	0.5329	0.5294	0.5232	0.5150	0.5003	0.4132	0.2766	0.2707	0.2657	0.2568
Reversed ranking											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0675	0.0903	0.0782	0.0616	0.0737	0.0675	0.2270	0.4310	0.3451	0.3737	0.4121
200	0.0874	0.0643	0.1121	0.0760	0.0693	0.0781	0.1932	0.4176	0.4037	0.4191	0.3784
150	0.0800	0.0859	0.0754	0.0732	0.0678	0.0838	0.2224	0.4156	0.3938	0.3671	0.4160
100	0.0950	0.0844	0.0912	0.0783	0.0992	0.0873	0.1853	0.3565	0.3911	0.4147	0.4387
50	0.0379	0.0191	0.0238	0.0271	0.0315	0.0258	0.1109	0.2912	0.2537	0.2483	0.2376
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0576	0.1461	0.1578	0.1646	0.1786
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0557	0.1766	0.1482	0.1708	0.1607
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0531	0.1510	0.1707	0.1824	0.1591
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0583	0.1582	0.1620	0.1796	0.1766
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0581	0.1599	0.1517	0.1694	0.1639
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0520	0.1501	0.1786	0.1493	0.1685

Table 14
Scenario 3 - nRMSE - AP.

BM25											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.3882	0.3882	0.3879	0.3862	0.3741	0.0325	0.1257	0.1799	0.2069	0.2107	0.2110
200	0.3883	0.3881	0.3879	0.3860	0.3727	0.0347	0.1284	0.1810	0.2091	0.2196	0.2084
150	0.3881	0.3882	0.3881	0.3864	0.3729	0.0316	0.1374	0.1753	0.1999	0.2083	0.2133
100	0.3882	0.3882	0.3882	0.3861	0.3720	0.0313	0.1300	0.1826	0.2060	0.2093	0.2063
50	0.3881	0.3881	0.3878	0.3865	0.3744	0.0282	0.1263	0.1738	0.2009	0.2007	0.2032
0	0.3893	0.3893	0.3893	0.3882	0.3761	0.0000	0.0997	0.1414	0.1672	0.1664	0.1699
−50	0.3880	0.3876	0.3868	0.3854	0.3803	0.3524	0.3409	0.3458	0.3488	0.3522	0.3525
−100	0.3863	0.3854	0.3840	0.3818	0.3770	0.3636	0.3506	0.3526	0.3582	0.3603	0.3596
−150	0.3846	0.3835	0.3817	0.3792	0.3742	0.3642	0.3516	0.3538	0.3600	0.3596	0.3594
−200	0.3833	0.3818	0.3799	0.3770	0.3723	0.3642	0.3510	0.3533	0.3593	0.3593	0.3604
−250	0.3817	0.3803	0.3784	0.3754	0.3713	0.3642	0.3516	0.3540	0.3578	0.3609	0.3598
RM3											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.4582	0.4567	0.4530	0.4427	0.4135	0.0343	0.1167	0.1374	0.1630	0.1708	0.1776
200	0.4581	0.4566	0.4523	0.4438	0.4135	0.0363	0.1165	0.1419	0.1601	0.1694	0.1765
150	0.4581	0.4565	0.4526	0.4435	0.4114	0.0306	0.1227	0.1352	0.1653	0.1710	0.1825
100	0.4583	0.4563	0.4523	0.4428	0.4153	0.0339	0.1169	0.1344	0.1625	0.1644	0.1776
50	0.4584	0.4564	0.4529	0.4446	0.4101	0.0365	0.1127	0.1275	0.1597	0.1623	0.1760
0	0.4594	0.4580	0.4536	0.4460	0.4158	0.0000	0.0815	0.1050	0.1277	0.1392	0.1548
−50	0.4575	0.4570	0.4543	0.4497	0.4367	0.3900	0.3786	0.3842	0.3900	0.3899	0.3975
−100	0.4552	0.4540	0.4520	0.4472	0.4376	0.4147	0.4035	0.4066	0.4115	0.4147	0.4186
−150	0.4529	0.4512	0.4488	0.4451	0.4368	0.4210	0.4097	0.4128	0.4181	0.4211	0.4240
−200	0.4507	0.4489	0.4461	0.4420	0.4355	0.4230	0.4127	0.4153	0.4203	0.4225	0.4277
−250	0.4488	0.4468	0.4438	0.4397	0.4338	0.4238	0.4142	0.4177	0.4214	0.4248	0.4286

Table 15
Scenario 3 - p-values - AP.

BM25											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0000	0.0000	0.0000	0.0000	0.0000	0.7578	0.0308	0.0013	0.0021	0.0083	0.0030
200	0.0000	0.0000	0.0000	0.0000	0.0000	0.6860	0.0572	0.0088	0.0096	0.0031	0.0064
150	0.0000	0.0000	0.0000	0.0000	0.0000	0.7157	0.0526	0.0088	0.0124	0.0105	0.0051
100	0.0000	0.0000	0.0000	0.0000	0.0000	0.6861	0.0443	0.0069	0.0046	0.0132	0.0025
50	0.0000	0.0000	0.0000	0.0000	0.0000	0.7343	0.0501	0.0171	0.0082	0.0036	0.0060
0	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.1148	0.0095	0.0411	0.0389	0.0389
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RM3											
Swaps \ Replacements	−250	−200	−150	−100	−50	0	50	100	150	200	250
250	0.0000	0.0000	0.0000	0.0000	0.0000	0.7396	0.1037	0.1164	0.0996	0.0635	0.0940
200	0.0000	0.0000	0.0000	0.0000	0.0000	0.7496	0.1946	0.1264	0.0733	0.0737	0.0815
150	0.0000	0.0000	0.0000	0.0000	0.0000	0.7698	0.1896	0.1583	0.0787	0.0519	0.0430
100	0.0000	0.0000	0.0000	0.0000	0.0000	0.7702	0.2478	0.1263	0.0945	0.0462	0.0498
50	0.0000	0.0000	0.0000	0.0000	0.0000	0.7325	0.1986	0.1450	0.0850	0.0943	0.0916
0	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.3217	0.2514	0.1759	0.1777	0.1234
−50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
−250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

References

Agosti, M., Di Nunzio, G. M., Ferro, N., & Silvello, G. (2019). An Innovative Approach to Data Management and Curation of Experimental Data Generated through IR Test Collections. In N. Ferro, & C. Peters (Eds.), *The Information Retrieval Series: 41*, *Information retrieval evaluation in a changing world – lessons learned from 20 years of CLEF* (pp. 105–122). Springer International Publishing, Germany.

Agosti, M., Ferro, N., & Thanos, C. (2012). DESIRE 2011 Workshop on Data infrastructurEs for Supporting Information Retrieval Evaluation. *SIGIR Forum*, *46*(1), 51–55.

Allan, J., Aslam, J. A., Pavlu, V., Kanoulas, E., & Carterette, B. (2009). Million Query Track 2008 Overview. In *NIST Special Publication: 500-277*, *Proc. 17th text retrieval conference (TREC 2008)*. National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Allan, J., Carterette, B., Dachev, B., Aslam, J. A., Pavlu, V., & Kanoulas, E. (2008). Million Query Track 2007 Overview. In *NIST Special Publication: 500-274*, *Proc. 16th text retrieval conference (TREC 2007)*. National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Arampatzis, A., & van Hameren, A. (2001). The Score Distributional Threshold Optimization for Adaptive Binary Classification Tasks. In D. H. Kraft, W. B. Croft, D. J. Harper, & J. Zobel (Eds.), *Proc. 24th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2001)* (pp. 285–293). ACM Press, New York, USA.

Arampatzis, A., & Kamps, J. (2009). A Signal-to-Noise Approach to Score Normalization. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, & J. J. Lin (Eds.), *Proc. 18th International conference on information and knowledge management (CIKM 2009)* (pp. 797–806). ACM Press, New York, USA.

Arampatzis, A., & Robertson, S. E. (2011). Modeling Score Distributions in Information Retrieval. *Information Retrieval*, *14*(1), 26–46.

Arguello, J., Crane, M., Diaz, F., Lin, J., & Trotman, A. (2015). Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). *SIGIR Forum*, *49*(2), 107–116.

Armstrong, T. G., Moffat, A., Webber, W., & Zobel, J. (2009). Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, & J. J. Lin (Eds.), *Proc. 18th International conference on information and knowledge management (CIKM 2009)* (pp. 601–610). ACM Press, New York, USA.

Baker, M. (2016). 1,500 Scientists Lift the Lid on Reproducibility. *Nature*, *533*, 452–454.

Breuer, T., Ferro, N., Fuhr, N., Maistro, M., Sakai, T., Schaer, P., et al. (2020). How to Measure the Reproducibility of System-oriented IR Experiments. In Y. Chang, X. Cheng, J. Huang, Y. Lu, J. Kamps, V. Murdock, J.-R. Wen, A. Diriye, J. Guo, O. Kurland (Eds.), *Proc. 43rd Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2020)* (pp. 349–358). ACM Press, New York, USA.

Breuer, T., Ferro, N., Maistro, M., & Schaer, P. (2021). repro_eval: A Python Interface to Reproducibility Measures of System-Oriented IR Experiments. In D. Hiemstra, M. Moens, J. Mothe, R. Perego, M. Potthast, & F. Sebastiani (Eds.), *Lecture Notes in Computer Science: vol. 12657*, *Advances in information retrieval - 43rd european conference on IR Research (ECIR 2021), Proceedings, Part II* (pp. 481–486). Springer.

Breuer, T., Keller, J., & Schaer, P. (2022). Ir_metadata: An extensible metadata schema for IR experiments. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, & G. Kazai (Eds.), *Proc. 45th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2022)* (pp. 3078–3089). ACM Press, New York, USA.

Breuer, T., & Schaer, P. (2021). A Living Lab Architecture for Reproducible Shared Task Experimentation. In C. Wolff, & T. Schmidt (Eds.), *Information between data and knowledge: information science and its neighbors from data science to digital humanities - Proc. 16th international symposium of information science (ISI 2021)* (pp. 348–362). Werner Hülsbusch.

Carterette, B., Pavlu, V., Fang, H., & Kanoulas, E. (2010). Million Query Track 2009 Overview. In *NIST Special Publication: 500-278*, *Proc. 18th text retrieval conference (TREC 2009)*. National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Clancy, R., Ferro, N., Hauff, C., Sakai, T., & Wu, Z. (2019). The SIGIR 2019 Open-Source IR Replicability Challenge (OSIRRC 2019). In *Proc. 42nd Annual International ACM SIGIR Conference on research and development in information retrieval (SIGIR 2019)* (pp. 1432–1434). ACM Press, New York, USA, editor=Piwowarski, B. and Chevalier, M. and Gaussier, E. and Maarek, Y. and Nie, J.-Y. and Scholer, F.,.

Cleverdon, C. W. (1962). *Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems*. Aslib Cranfield Research Project, College of Aeronautics, Cranfield, UK.

Cleverdon, C. W. (1967). The Cranfield Tests on Index Languages Devices. *Aslib Proceedings*, *19*(6), 173–194.

Cummins, R. (2014). Document Score Distribution Models for Query Performance Inference and Prediction. *ACM Transactions on Information System (TOIS)*, *32*(1), 2:1–2:28.

Dacrema, M. F., Boglio, S., Cremonesi, P., & Jannach, D. (2021). A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Transactions on Information Systems (TOIS)*, *39*(2), 20:1–20:49.

Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are we Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In T. Bogers, A. Said, P. Brusilovsky, & D. Tikk (Eds.), *Proc. 13th ACM conference on recommender systems (RecSys 2019)* (pp. 101–109). ACM Press, New York, USA.

De Roure, D. (2014). The Future of Scholarly Communications. *Insights*, *27*(3), 233–238.

Ferrante, M., Ferro, N., & Maistro, M. (2015). Towards a Formal Framework for Utility-oriented Measurements of Retrieval Effectiveness. In J. Allan, W. B. Croft, A. P. de Vries, C. Zhai, N. Fuhr, & Y. Zhang (Eds.), *Proc. 1st ACM SIGIR International conference on the theory of information retrieval (ICTIR 2015)* (pp. 21–30). ACM Press, New York, USA.

Ferro, N. (2017). Reproducibility Challenges in Information Retrieval Evaluation. *ACM Journal of Data and Information Quality (JDIQ)*, *8*(2), 8:1–8:4.

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". *SIGIR Forum*, *50*(1), 68–82.

Ferro, N., Fuhr, N., Maistro, M., Sakai, T., & Soboroff, I. (2019a). CENTRE@CLEF2019: Overview of the Replicability and Reproducibility Tasks. In L. Cappellato, N. Ferro, D. E. Losada, & H. Müller (Eds.), *CLEF 2019 Working notes*. CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073.

Ferro, N., Fuhr, N., Maistro, M., Sakai, T., & Soboroff, I. (2019b). Overview of CENTRE@CLEF 2019: Sequel in the Systematic Reproducibility Realm. In F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. E. Losada, G. Heinatz Bürki, L. Cappellato, & N. Ferro (Eds.), *Experimental IR Meets multilinguality, multimodality, and interaction. proceedings of the tenth international conference of the CLEF association (CLEF 2019)* (pp. 287–300). Lecture Notes in Computer Science (LNCS) 11696, Springer, Heidelberg, Germany.

Ferro, N., & Kelly, D. (2018). SIGIR Initiative to Implement ACM Artifact Review and Badging. *SIGIR Forum*, *52*(1), 4–10.

Ferro, N., Maistro, M., Sakai, T., & Soboroff, I. (2018). Overview of CENTRE@CLEF 2018: a First Tale in the Systematic Reproducibility Realm. In P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J.-Y. Nie, L. Soulier, E. SanJuan, L. Cappellato, & N. Ferro (Eds.), *Experimental IR Meets multilinguality, multimodality, and interaction. proceedings of the nineth international conference of the CLEF association (CLEF 2018)* (pp. 239–246). Lecture Notes in Computer Science (LNCS) 11018, Springer, Heidelberg, Germany.

Freire, J., Fuhr, N., & Rauber, A. (Eds.), (2016). Report from Dagstuhl Seminar 16041: Reproducibility of Data-Oriented Experiments in e-Science. In *Dagstuhl Reports, Volume 6, Number 1*, *Report from dagstuhl seminar 16041: reproducibility of data-oriented experiments in e-science*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Germany.

Gibney, E. (2020). This AI Researcher is Trying to Ward off a Reproducibility Crisis. *Nature*, *577*, 14.

Gollub, T., Stein, B., & Burrows, S. (2012). Ousting Ivory Tower Research: Towards a Web Framework for Providing Experiments as a Service. In W. Hersh, J. Callan, Y. Maarek, & M. Sanderson (Eds.), *Proc. 35th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2012)* (pp. 1125–1126). ACM Press, New York, USA.

Gollub, T., Stein, B., Burrows, S., & Hoppe, D. (2012). TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. In A. Hameurlain, A. M. Tjoa, & R. R. Wagner (Eds.), *Proc. 23rd International workshop on database and expert systems applications (DEXA 2012)* (pp. 151–155). IEEE Computer Society.

Hopfgartner, F., Hanbury, A., Müller, H., Kando, N., Mercer, S., Kalpathy-Cramer, J., et al. (2015). Report on the Evaluation-as-a-Service (EaaS) Expert Workshop. *SIGIR Forum*, *49*(1), 57–65.

Ivie, P., & Thain, D. (2018). Reproducibility in Scientific Computing. *ACM Computing Surveys*, *51*(3), 63:1–63:36.

Jaleel, N. A., Allan, J., Croft, W. B., Diaz, F., Larkey, L. S., Li, X., et al. (2004). UMass at TREC 2004: Novelty and HARD. In *NIST Special Publication*: *500–261*, *Proc. 13th text retrieval conference (TREC 2004)*. National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446.

Kendall, M. G. (1945). The Treatment of Ties in Ranking Problems. *Biometrika*, *33*(3), 239–251.

Lavrenko, V., & Croft, W. B. (2001). Relevance-Based Language Models. In D. H. Kraft, W. B. Croft, D. J. Harper, & J. Zobel (Eds.), *Proc. 24th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2001)* (pp. 120–127). ACM Press, New York, USA.

Lin, J. (2018). The Neural Hype and Comparisons against Weak Baselines. *SIGIR Forum*, *52*(2), 40–51.

Lin, J., Ma, X., Lin, S., Yang, J., Pradeep, R., & Nogueira, R. (2021). Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, & T. Sakai (Eds.), *Proc. 44th Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2021)* (pp. 2356–2362). ACM Press, New York, USA.

Lucic, A., Bleeker, M. J. R., de Rijke, M., Sinha, K., Jullien, S., & Stojnic, R. (2022). Towards Reproducible Machine Learning Research in Information Retrieval. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, & G. Kazai (Eds.), *Proc. 45th Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2022)* (pp. 3459–3461). ACM Press, New York, USA.

Lv, Q., Ding, M., Liu, Q., Chen, Y., Feng, W., He, S., et al. (2021). Are we Really Making Much Progress?: Revisiting, Benchmarking and Refining Heterogeneous Graph Neural Networks. In F. Zhu, B. C. Ooi, C. Miao (Eds.), *Proc. 27th ACM SIGKDD Conference on knowledge discovery and data mining (KDD 2021)* (pp. 1150–1160). ACM Press, New York, USA.

MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., & Goharian, N. (2021). Simplified Data Wrangling with ir_datasets. In *Proc. 44th Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2021)* (pp. 2429–2436). ACM Press, New York, USA.

Macdonald, C., & Tonellotto, N. (2020). Declarative Experimentation in Information Retrieval using PyTerrier. In K. Balog, V. Setty, C. Lioma, Y. Liu, M. Zhang, & K. Berberich (Eds.), *Proc. 2020 ACM SIGIR International conference on the theory of information retrieval (ICTIR 2020)* (pp. 161–168). ACM Press, New York, USA.

Manmatha, R., Rath, T., & Feng, F. (2001). Modelling Score Distributions for Combining the Outputs of Search Engines. In D. H. Kraft, W. B. Croft, D. J. Harper, & J. Zobel (Eds.), *Proc. 24th Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2001)* (pp. 267–275). ACM Press, New York, USA.

Marchesin, S., Purpura, A., & Silvello, G. (2020). Focal Elements of Neural Information Retrieval Models. An Outlook through a Reproducibility Study. *Information Processing & Management*, *57*(6), Article 102109.

National Academies of Sciences, Engineering, and Medicine (2019). *Reproducibility and Replicability in Science*. The National Academies Press, Washington, USA.

Open Science Collaboration (2015). Estimating the Reproducibility of Psychological Science. *Science*, *349*(6251), 943–952.

Parapar, J., Losada, D. E., Presedo Quindimil, M. A., & Barreiro, A. (2020). Using Score Distributions to Compare Statistical Significance Tests for Information Retrieval Evaluation. *Journal of the Association for Information Science and Technology (JASIST)*, *71*(1), 98–113.

Parapar, J., & Radlinski, F. (2021). Towards Unified Metrics for Accuracy and Diversity for Recommender Systems. In H. J. C. Pampín, M. A. Larson, M. C. Willemsen, J. A. Konstan, J. J. McAuley, J. Garcia-Gathright, B. Huurnink, & E. Oldridge (Eds.), *Proc. 15th ACM Conference on recommender systems (RecSys 2021)* (pp. 75–84). ACM Press, New York, USA.

Pashler, H., & Wagenmakers, E.-J. (2012). Editors' Introduction to the Special Section on Replicability in Psychological Science: A Crisis of Confidence? *Perspectives on Psychological Science*, *7*(6), 528–530.

Plesser, H. E. (2017). Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers Neuroinformatics*, *11*, 76.

Potthast, M., Gollub, T., Wiegmann, M., & Stein, B. (2019). TIRA Integrated Research Architecture. In N. Ferro, & C. Peters (Eds.), *The Information Retrieval Series*: *41*, *Information retrieval evaluation in a changing world – lessons learned from 20 Years of CLEF*. Springer International Publishing, Germany.

Robertson, S. E. (2001). On Score Distributions and Relevance. In D. H. Kraft, W. B. Croft, D. J. Harper, & J. Zobel (Eds.), *Proc. 24th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2001)* (pp. 40–51). ACM Press, New York, USA.

Robertson, S. E., & Kanoulas, E. (2012). On Per-topic Variance in IR Evaluation. In W. Hersh, J. Callan, Y. Maarek, & M. Sanderson (Eds.), *Proc. 35th Annual international ACM SIGIR Conference on research and development in information retrieval (SIGIR 2012)* (pp. 891–900). ACM Press, New York, USA.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1994). Okapi at TREC–3. In *NIST Special Publication*: *500-225*, *Proc. 3rd text retrieval conference (TREC-3)* (pp. 109–126). National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Robertson, S. E., & Zaragoza, U. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval (FnTIR)*, *3*(4), 333–389.

Sakai, T., Ferro, N., Soboroff, I., Zeng, Z., Xiao, P., & Maistro, M. (2019). Overview of the NTCIR-14 CENTRE Task. In E. Ishita, N. Kando, M. P. Kato, & Y. Liu (Eds.), *Proc. 14th NTCIR Conference on evaluation of information access technologies (NTCIR-14)* (pp. 494–509). National Institute of Informatics, Tokyo, Japan.

Schaer, P., Breuer, T., Castro, L. J., Wolff, B., Schaible, J., & Tavakolpoursaleh, N. (2021). Overview of LiLAS 2021 - Living Labs for Academic Search (Extended Overview). In G. Faggioli, N. Ferro, A. Joly, M. Maistro, & F. Piroi (Eds.), *CLEF 2021 Working notes*. CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073.

Soboroff, I., Ferro, N., Maistro, M., & Sakai, T. (2019). Overview of the TREC 2018 CENTRE Track. In E. M. Voorhees, & A. Ellis (Eds.), *NIST Special Publication*: *500-331*, *Proc. 27th text retrieval conference (TREC 2018)*. National Institute of Standards and Technology (NIST), Gaithesburg, Maryland, USA.

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005). Indri: A Language Model-based Search Engine for Complex Queries. *Proceedings of the International Conference on Intelligent Analysis*, *2*(6), 2–6.

Swets, J. A. (1963). Information Retrieval Systems. *Science*, *141*(3577), 245–250.

Trotman, A., Clarke, C. L. A., Ounis, I., Culpepper, J. S., Cartright, M.-A., & Geva, S. (2012). Open Source Information Retrieval: a Report on the SIGIR 2012 Workshop. *ACM SIGIR Forum*, *46*(2), 95–101.

Voorhees, E. M. (1998). Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), *Proc. 21st Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 1998)* (pp. 315–323). ACM Press, New York, USA.

Voorhees, E. M. (2001). Evaluation by Highly Relevant Documents. In D. H. Kraft, W. B. Croft, D. J. Harper, & J. Zobel (Eds.), *Proc. 24th Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2001)* (pp. 74–82). ACM Press, New York, USA.

Voorhees, E. M., Rajput, S., & Soboroff, I. (2016). Promoting Repeatability Through Open Runs. In E. Yilmaz, & C. L. A. Clarke (Eds.), *Proc. 7th International workshop on evaluating information access (EVIA 2016)* (pp. 17–20). National Institute of Informatics, Tokyo, Japan.

Webber, W., Moffat, A., & Zobel, J. (2010). A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems (TOIS)*, *4*(28), 20:1–20:38.

Yang, P., Fang, H., & Lin, J. (2018). Anserini: Reproducible Ranking Baselines Using Lucene. *ACM Journal of Data and Information Quality (JDIQ)*, *10*(4), 16:1–16:20.

Yang, W., Lu, K., Yang, P., & Lin, J. (2019). Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In B. Piwowarski, M. Chevalier, E. Gaussier, Y. Maarek, J. Nie, & F. Scholer (Eds.), *Proc. 42nd Annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2019)* (pp. 1129–1132). ACM Press, New York, USA.