DiffIR: Exploring Differences in Ranking Models' Behavior

Kevin Martin Jose
Max Planck Institute for Informatics
Saarbrücken, Germany
kjose@mpi-inf.mpg.de

Thong Nguyen
Max Planck Institute for Informatics
Saarbrücken, Germany
thongnt@mpi-inf.mpg.de

Sean MacAvaney
University of Glasgow
Glasgow, Scotland, UK
sean.macavaney@glasgow.ac.uk

Jeffrey Dalton
University of Glasgow
Glasgow, Scotland, UK
jeff.dalton@glasgow.ac.uk

Andrew Yates
Max Planck Institute for Informatics
Saarbrücken, Germany
ayates@mpi-inf.mpg.de

ABSTRACT

Understanding and comparing the behavior of retrieval models is a fundamental challenge that requires going beyond examining average effectiveness and per-query metrics, because these do not reveal key differences in how ranking models' behavior impacts individual results. DiffIR is a new open-source web tool to assist with qualitative ranking analysis by visually 'diffing' system rankings at the individual result level for queries where behavior significantly diverges. Using one of several configurable similarity measures, it identifies queries for which the rankings of models compared have important differences in individual rankings and provides a visual web interface to compare the rankings side-by-side. DiffIR additionally supports a model-specific visualization approach based on custom term importance weight files. These support studying the behavior of interpretable models, such as neural retrieval methods that produce document scores based on a similarity matrix or based on a single document passage. Observations from this tool can complement neural probing approaches like ABNIRML to generate quantitative tests. We provide an illustrative use case of DiffIR by studying the qualitative differences between recently developed neural ranking models on a standard TREC benchmark dataset.

CCS CONCEPTS

• **Information systems** → **Retrieval effectiveness**.

KEYWORDS

information retrieval, evaluation, analysis

ACM Reference Format:
Kevin Martin Jose, Thong Nguyen, Sean MacAvaney, Jeffrey Dalton, and Andrew Yates. 2021. DiffIR: Exploring Differences in Ranking Models' Behavior. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3462784

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462784

1 INTRODUCTION

Cranfield-style offline experiments with pooled relevance judgments are widely used to compare systems in terms of ranking effectiveness [41]. In this paradigm, a fixed set of queries are issued to different ranking systems, their results are combined into a document pool containing a subset of the document collection, documents in the pool are judged for relevance by assessors, and ranking metrics are calculated over these relevance judgments. Studies show that these metrics positively correlate with user satisfaction [3, 37], though there is disagreement over what constitutes a valid (and useful) ranking metric [11, 12, 36]. Quantitative metrics may identify which systems are more effective overall or for particular queries, but they do not provide insights on *how* and *why* system behavior differs. This task falls to the researcher to dig deeper and inspect the results in greater depth manually. Modern tools to support this type of analysis are important to support advancements in research.

Diagnostic datasets allow researchers to probe for specific model behaviors, such as whether a ranking method satisfies IR axioms or whether text modifications substantially modify its behavior [10, 29, 35]. Despite significant effectiveness gains by neural retrieval methods [27], experiments using axiomatic perturbations suggest that their behavior is not well understood by previous IR axioms [35]. Using a battery of probes involving textual manipulation and transfer learning to analyze the behavior of neural IR models, MacAvaney et al. [29] find that seemingly irrelevant changes like appending a non-relevant sentence to the end of documents can substantially alter results from an effective neural retrieval model. While these existing approaches aid the understanding of complex models, they require that behaviors be specified in advance.

We present DiffIR[1], a newly developed tool to perform qualitative measurement of differences in models' behavior. Given a pair of TREC runs containing rankings for multiple queries, DiffIR identifies *contrasting* queries that have "substantially" different results between two systems and generates a visual side-by-side comparison that illustrates how the key rankings differ. DiffIR supports multiple strategies for ranking comparison, including unsupervised ranking correlations like τ_{AP} [46] and supervised comparison based on existing judgments and ranking metrics. DiffIR's side-by-side comparison shows result snippets from both systems and indicates each result's position in the other system's ranking. In addition to identifying snippets with exact term matching, DiffIR shows

[1] https://github.com/capreolus-ir/diffir

the snippets most relevant to a model's relevance prediction when a term importance file is provided indicating a model's weights over terms in a document. Highly-effective neural models often contain assumptions that can be used to generate such term importance weights. For example, BERT–MaxP [9] is based on the assumption that a document's score can be computed by a single relevant passage, and CEDR [31] uses a similarity matrix indicating the strengths of associations between query and document terms. DiffIR allows researchers to explore *how* systems differ at a result level and identify differences in what terms or passages the methods predict to be most important.

Our contributions include:

- An open-source tool for qualitative analysis of ranking models' behavior with both Web and command line interfaces.
- A demonstration of the DiffIR tool available online at https://github.com/capreolus-ir/diffir supporting standard retrieval collections and data formats.
- A case study using the tool to analyze the behavioral differences of recent neural methods on TREC Robust04 [42].

2 RELATED WORK

Significant prior work studies the question of how two sets of document rankings can be compared without relevance judgments in order to determine if they substantially differ [13, 14, 23, 24, 43, 46]. DiffIR identifies such rankings through its query contrast component, that uses τ_{AP} [46] and has a modular design allowing other measures to be incorporated. Similarly, Carterette et al. [5] propose a measure for determining which topics should be judged and apply it to the task of efficiently constructing a minimal test collection. Others investigate how to leverage existing relevance judgments to perform this task [16, 21, 25]. The Neural-IR-Explorer [19] is a tool for exploring a single ranking produced by the TK [20] neural re-ranker, which identifies queries of interest by clustering them based on their mean-contextualized encoding (e.g., queries asking for phone numbers, long and complex questions, or trivia questions), and reports the effectiveness of the model for each cluster.

Several studies illustrate the usefulness of performing side-by-side comparisons when assessing model quality. Thomas and Hawking [38] provide a way to compare two ranking models by replacing the user's usual query interface with their tool and present the user with results from the two models on different panes. The user, who does not know which set of results was generated by which model, is asked for explicit judgements of preference. Since the user's satisfaction depends on the entire resultset presented to the user, this approach allows for holistic comparison of all aspects of the ranking model like coverage and the quality of the document summaries displayed. Side-by-side comparisons are also widely used by Google and other leading search engines when making decisions on search quality and what features to add to search.[2] Carterette et al. [6] use preference judgments in settings where document relevance is difficult to measure by asking an assessor to judge between two pages of results. Similarly, Golbus et al. [15] propose an evaluation system in which the document being judged is paired with a context document on the same topic, and Bailey et al. [4] investigate incorporating the context of the search engine result page when judging document relevance.

[2]https://www.google.com/search/howsearchworks/mission/users/

3 SYSTEM DESCRIPTION

The DiffIR system is composed of two main modules: *Query Contrast* measures and *Term Importance* weights. The former is responsible for measuring the disparity in the performance of two models given a query and two ranking results, and the latter helps to identify which regions a model pays attention to the most in each document. Term Importance weights are also used to generate document snippets in our system.

3.1 Query Contrast Measures

Supervised Measures can only be calculated when users have access to query relevance judgments (qrels). Given the relevance judgments and two ranking results of two models, this module evaluates each model's effectiveness on every query using widely-used information retrieval metrics (e.g., MAP, NDCG). The module then calculates the metric difference for each query and returns the top k queries where the models show the most inconsistency regarding the ground truths. Note that this approach treats all documents of equal relevance level as equivalent; two systems may achieve the same effectiveness with different documents.

Unsupervised measures like rank correlation metrics can be used when relevance judgments are not available Though there are many popular rank correlation statistics, DiffIR currently supports *weightedtau* [40] and τ_{AP} (*tauAP*); both consider the ranking position when calculating the ranking agreement. In both metrics, exchanges of high rank are more influential than disagreements in low rank. A key difference is that *weightedtau* uses the raw ranking score to weight agreement/disagreement document pairs, whereas *tauAP* uses rank position for this purpose. This rank-based weighting is perfectly suitable for ad hoc retrieval tasks where users usually only look at top retrieved results. Since this approach does not use relevance judgments, two systems may achieve comparable effectiveness for a query in terms of relevance metrics while differing in which documents are ranked highly.

3.2 Term Importance Weights

Since term importance varies from model to model, DiffIR allows the user to provide weight files that indicate the weight given to specific segments of a document. Capreolus [45] can generate these weight files for several neural reranking models by following one of several strategies for producing term importance weights. OpenNIR [28] can currently generate weight files for the EPIC model [30]. We give a brief overview of several weight extraction strategies for common NIR architectures.

3.2.1 Interaction Matrix. Interaction-based models like KNRM [44], CEDR-KNRM [31], and TK [18] construct a similarity matrix that capture the interaction between query and document terms. Though similarity matrices often serve as a feature for subsequent layers (e.g., DRMM [17]), or as a source for convolutional n-grams (e.g., Co-PACRR [22]), they nevertheless provide insights into the relative importance of different document terms in determining the final score of the document. In models where multiple similarity matrices are sometimes employed (e.g., CEDR-KNRM), these can be

condensed to a single weight by changing model hyperparameters (e.g., to use only a single matrix), deriving final weights from the linear layer used to combine the per-matrix weights (if applicable), or making simplifying assumptions. In this work, we consider only a single similarity matrix with CEDR-KNRM and use the maximum similarity between a document term and any query term as the weight of that document term.

3.2.2 Passages. In models that produce a score for each document passage (e.g., BERT–MaxP [9]), these scores are used to assign a uniform weight to all document terms within the passage. This approach can easily be extended to support models that consider the top-k passages in a document (e.g., Birch [2]).

3.2.3 Term Scoring. Some recent neural models, such as EPIC [30] and DeepCT [8], take the approach of predicting scalar importance weights for individual terms in the document. This naturally lends them to visualizing the impact of individual terms. In this demonstration, we include weights for an EPIC model trained with OpenNIR [28], given that its expansion mechanism can lend itself to interesting comparisons. Each term in EPIC produces importance scores for each term in the source lexicon (BERT WordPieces). We produce a weight for each document token using the maximum importance score over terms that appear in the query. For visualization, weights are min-max normalized.

3.2.4 Query Term Matching. When model-specific weight files are not provided, DiffIR will resort to unsupervised query term matching. We find all occurrences of each query term in the document, and all matches are assigned a uniform weight. Even though this approach does not reflect the underlying term matching mechanism of the models involved, the highlighted matches help users to locate document regions that are potentially relevant to the query. We use the Aho–Corasick [1] algorithm to find the matches of all query terms in the document.

3.3 Implementation Details

DiffIR is implemented in Python. It can be used as a command line tool and as a Python package that can be integrated with other tools. DiffIR is model-agnostic; in its most basic setting, it simply accepts TREC-formatted run files and an `ir_datasets` [32] dataset identifier to generate an HTML output. Metrics are calculated using `pytrec_eval` [39] via the `ir_measures`[3] package. To keep the package lightweight, DiffIR offloads the model-dependent term importance weight calculation to other packages like Capreolus and OpenNIR. These weights are supplied as optional inputs.

4 DEMONSTRATION

DiffIR can be installed through the Python package index and exposes a command line interface that receives either one or two runfiles and a configuration as the input. The user can choose to output an HTML file that can be viewed in a browser (Figure 1) or output directly to the console (Figure 2). The former is recommended, though the console output is useful for visualizing a single run file. DiffIR can be run locally using the command:

[3]https://github.com/terrierteam/ir_measures

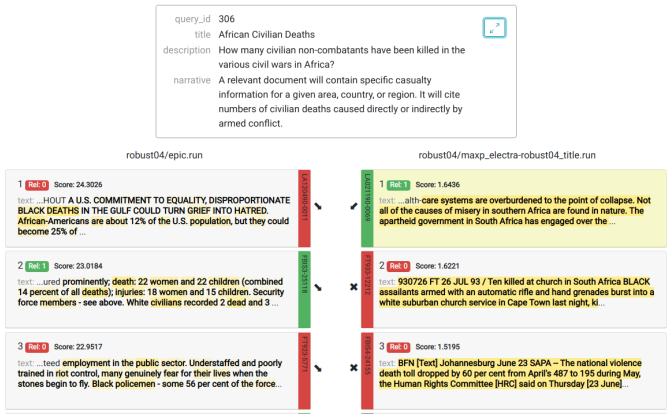

Figure 1: Comparing EPIC and BERT–MaxP for the query "african civilian deaths" on the TREC Robust 2004 dataset

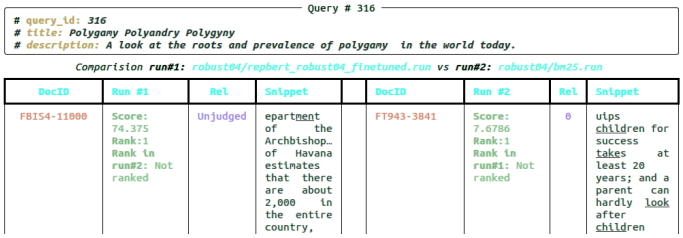

Figure 2: Command line interface for comparing two ranking models

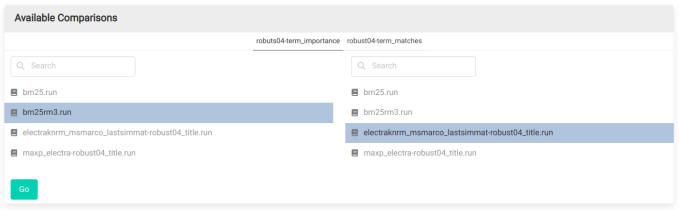

Figure 3: When run in batch mode, DiffIR generates a user interface to interactively compare result file pairs

```
python -m diffir.run <run_1> <run_2> -w \
--measure qrel --topk 10 --metric MAP \
--dataset trec-robust04
```

In the above command, `run_1` and `run_2` are files that contain the document rankings for each query and uses the standard TREC run format. The user must specify a dataset name supported by `ir_datasets` [32]. In the sample command above, DiffIR would select the top ten queries whose mean average precision varies the most between the two run files and renders the content as HTML.

Figure 1 shows a rendered HTML file that compares two run files: the documents on the left are retrieved by EPIC [9] and the ones on the right are from BERT–MaxP for the same query. (Both are using the ELECTRA [7] variant of BERT.) EPIC has ranked a non-relevant document in the top place, while BERT–MaxP has correctly placed a

Table 1: Results on TREC Robust04 for the methods studied.

	MAP	P@20	nDCG@20
BM25	0.2531	0.3631	0.4240
BM25 + RM3	0.3033	0.3974	0.4514
CEDR-KNRM (ELECTRA)	0.3506	0.4594	0.5358
BERT–MaxP (ELECTRA)	0.3239	0.4355	0.5014
EPIC	0.2068	0.3568	0.4124

relevant document at the top position, followed by two non relevant documents. The arrows indicate the relative rank of that document on the other side. DiffIR generates a summary of each document using the supplied weight files and shows the highest weighted span in the document. For BERT–MaxP this is the highest-scoring passage while for EPIC we select the part of the document with the highest cumulative document term weights.

A similar command can be used to generate a HTML file for every pair of runs in a directory, such as is available in our demo:

```
python -m diffir.batchrun <run directory> \
-o <output directory> --measure tauap \
--dataset trec-robust04
```

The weight files required for visualizing document term importance can be obtained through Capreolus for supported models like BERT–MaxP and CEDR-KNRM:

```
python -m capreolus.run rerank.diffirweights  \
with file=<path_to_model_config_file>
```

DiffIR also supports weights from the EPIC [30] model using OpenNIR's interface with the PyTerrier [33] package:

```
epic = onir.pt.epic.reranker
    .from_checkpoint('/path/to/epic/checkpoint')
diffir_weights = epic.explain_diffir(run)
```

A common use case is to train multiple NIR models on multiple datasets for comparison. When combined with a hyperparameter search, this can result in a large number of TREC result files. DiffIR can take a folder containing separate result directories and generate the comparison HTML files in batches. Figure 3 shows the generated "landing page" that makes it easier to view comparisons at scale.

We additionally provide an online notebook to conveniently explore DiffIR without installing the package locally.

5 CASE STUDY

To demonstrate the value of DiffIR, we conduct a case study investigating the performance of several models on the TREC Robust 2004 benchmark [42]. In the study, we compare EPIC, BERT–MaxP, and BM25 with RM3 expansion. The effectiveness of these approaches are shown in in Table 1.

Though effective for passage ranking, EPIC has yet to be shown to be effective for document ranking. We find that EPIC is sometimes hindered by its maximum token limit; the most important passage identified by BERT–MaxP (ELECTRA) is often outside EPIC's range. This suggests that applying techniques for long documents (e.g., [26]) may benefit this model. Even when the important paragraphs are within this range, however, EPIC can struggle with queries that do not have a clear path to term expansion. For instance, in the query "new fuel sources", it easily identifies specific fuels (often yielding high scores for terms like "ethanol" and "coal"), but is unable to recognize which ones would be considered "new". In this case, BERT–MaxP is able to identify these cases with ease, highlighting limitations of the query term independence assumption [34]. In this case, EPIC's expansion performs favorably compared to RM3 expansion: it ranks two relevant documents in the top 10 results that RM3 ranked at positions 235 and 269, respectively. Another example is with the query "automobile recalls", where EPIC's top-ranked document is about automobile car set recalls, rather than recalls of automobiles themselves. BERT–MaxP also handled this document incorrectly, but ranked it much lower (at position 8), with relevant documents above.

We found that DiffIR can also help quickly characterize unjudged documents. In the case of the query "agoraphobia" (fear of open spaces), half of BERT–MaxP's top 10 results were unjudged. The highlighting and snippets demonstrated that all of these unjudged documents were likely not relevant—at least, not in the maximum passage of the document. Curiously, these were ranked far above documents that used the query term itself. This suggests that BERT–MaxP may have a problem with rare terms. To characterize this further, quantitative probes could be built using a framework like ABNIRML to see if this behavior is systematic. This illustrates how DiffIR complements systematic probing approaches.

We also found that BERT–MaxP sometimes struggles when there are very short passages. For instance, for the query 'quilts, income', two of the top 10 documents included very short passages related to income (e.g., "than being tied down to $40 a day."). Note that this is simply the last segment of text from a much longer document, and it coincidentally is only so short by chance (due to the procedure for generating passages). This suggests that enforcing a minimum passage length may be beneficial for this model.

This case study demonstrated the value of the DiffIR tool. Some of the observations could have an immediate impact on modeling decisions, such as the enforcement of a minimum passage length for BERT–MaxP. Others are more anecdotal, and could be the basis for a more comprehensive quantitative evaluation.

6 CONCLUSION

In this work, we described our demonstration of DiffIR, an open-source tool for exploring the differences in retrieval model's behavior via Web and command line interfaces. Our demonstration illustrates how DiffIR's query contrast component is used to identify queries that substantially differ between two methods and how model-specific term importance weighting (and snippet selection) is used to shed light on what influenced a model's relevance score. Both DiffIR's query contrast component and the term importance weighting component are modular, allowing for the easy integration of different query contrast approaches and term importance weights from new ranking models.

ACKNOWLEDGMENTS

This work was supported in part by the TensorFlow Research Cloud.

REFERENCES

[1] Alfred V Aho and Margaret J Corasick. 1975. Efficient string matching: an aid to bibliographic search. *Commun. ACM* 18, 6 (1975), 333–340.

[2] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 3490–3496.

[3] Azzah Al-Maskari, Mark Sanderson, Paul Clough, and Eija Airio. 2008. The good and the bad system: does the test collection predict users' effectiveness?. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 59–66.

[4] Peter Bailey, Nick Craswell, Ryen W White, Liwei Chen, Ashwin Satyanarayana, and Seyed MM Tahaghoghi. 2010. Evaluating search systems using result page context. In *Proceedings of the third symposium on Information interaction in context*.

[5] Ben Carterette, James Allan, and Ramesh Sitaraman. 2006. Minimal Test Collections for Retrieval Evaluation. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) *(SIGIR '06)*. Association for Computing Machinery, New York, NY, USA, 268–275. https://doi.org/10.1145/1148170.1148219

[6] Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. 2008. Here or There: Preference Judgments for Relevance. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval* (Glasgow, UK) *(ECIR'08)*. Springer-Verlag, Berlin, Heidelberg, 16–27.

[7] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

[8] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. In *arXiv:1910.10687*.

[9] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[10] Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems (TOIS)* 29, 2 (2011), 1–42.

[11] Marco Ferrante, Nicola Ferro, and Norbert Fuhr. 2021. Towards Meaningful Statements in IR Evaluation. Mapping Evaluation Measures to Interval Scales. *arXiv preprint arXiv:2101.02668* (2021).

[12] N. Fuhr. 2018. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum* 51 (2018), 32–41.

[13] Ning Gao, Mossaab Bagdouri, and Douglas W Oard. 2016. Pearson rank: a head-weighted gap-sensitive score-based correlation coefficient. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 941–944.

[14] Ning Gao and Douglas Oard. 2015. A head-weighted gap-sensitive correlation coefficient. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 799–802.

[15] Peter B Golbus, Imed Zitouni, Jin Young Kim, Ahmed Hassan, and Fernando Diaz. 2014. Contextual and dimensional relevance judgments for reusable SERP-level evaluation. In *Proceedings of the 23rd international conference on World wide web*.

[16] John Guiver, Stefano Mizzaro, and Stephen Robertson. 2009. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Transactions on Information Systems (TOIS)* 27, 4 (2009), 1–26.

[17] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*. Indianapolis, Indiana, 55–64.

[18] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & Time-Budget-Constrained Contextualization for Re-Ranking. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*.

[19] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2019. Neural-IR-Explorer: A Content-Focused Tool to Explore Neural Re-Ranking Results. arXiv:1912.04713 [cs.IR]

[20] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2019. TU Wien @ TREC Deep Learning '19 – Simple Contextualization for Re-ranking. arXiv:1912.01385 [cs.IR]

[21] Mehdi Hosseini, Ingemar J Cox, Natasa Milic-Frayling, Vishwa Vinay, and Trevor Sweeting. 2011. Selecting a subset of queries for acquisition of further relevance judgements. In *Conference on the theory of information retrieval*. Springer.

[22] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard De Melo. 2018. Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 279–287.

[23] Gabriella Kazai and Homer Sung. 2014. Dissimilarity based query selection for efficient preference based IR evaluation. In *European conference on information retrieval*. Springer, 172–183.

[24] Mucahid Kutlu, Tamer Elsayed, Maram Hasanain, and Matthew Lease. 2018. When rank order isn't enough: New statistical-significance-aware correlation measures. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 397–406.

[25] Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2017. Learning to effectively select topics for information retrieval test collections. *arXiv:1701.07810* (2017).

[26] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093* (2020).

[27] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: Bert and beyond. *arXiv:2010.06467* (2020).

[28] Sean MacAvaney. 2020. OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*. 845–848. https://doi.org/10.1145/3336191.3371864

[29] Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2020. ABNIRML: Analyzing the Behavior of Neural IR Models. *arXiv:2011.00696* (2020).

[30] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via Prediction of Importance with Contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)* (Virtual Event, China). 1573–1576.

[31] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104. https://doi.org/10.1145/3331184.3331317

[32] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3404835.3463254

[33] C. MacDonald and N. Tonellotto. 2020. Declarative Experimentation in Information Retrieval using PyTerrier. *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval* (2020).

[34] Bhaskar Mitra, Corby Rosset, David Hawking, Nick Craswell, Fernando Diaz, and Emine Yilmaz. 2019. Incorporating query term independence assumption for efficient retrieval and ranking using deep neural networks. *arXiv:1907.03693* (2019).

[35] Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An axiomatic approach to diagnosing neural IR models. In *European Conference on Information Retrieval*. Springer, 489–503.

[36] T. Sakai and N. Kando. 2008. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval* 11 (2008), 447–470.

[37] Mark Sanderson, Monica Lestari Paramita, Paul Clough, and Evangelos Kanoulas. 2010. Do user preferences and evaluation measures line up?. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 555–562.

[38] Paul Thomas and David Hawking. 2006. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. 94–101.

[39] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. In *SIGIR*. ACM.

[40] Sebastiano Vigna. 2015. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*. 1166–1176.

[41] E. Voorhees. 2019. The Evolution of Cranfield. In *Information Retrieval Evaluation in a Changing World*.

[42] Ellen M. Voorhees. 2004. Overview of the TREC 2004 Robust Track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*. Gaithersburg, Maryland, 52–69.

[43] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28, 4 (2010), 1–38.

[44] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. Tokyo, Japan, 55–64.

[45] Andrew Yates, Kevin Martin Jose, Xinyu Zhang, and Jimmy Lin. 2020. Flexible IR pipelines with Capreolus. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3181–3188.

[46] Emine Yilmaz, Javed A Aslam, and Stephen Robertson. 2008. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 587–594.