The Istella22 Dataset: Bridging Traditional and Neural Learning to Rank Evaluation

Domenico Dato
Istella
Italy
domenico@istella.ai

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

Franco Maria Nardini
ISTI-CNR
Italy
francomaria.nardini@isti.cnr.it

Raffaele Perego
ISTI-CNR
Italy
raffaele.perego@isti.cnr.it

Nicola Tonellotto
University of Pisa
Italy
nicola.tonellotto@unipi.it

ABSTRACT

Neural approaches that use pre-trained language models are effective at various ranking tasks, such as question answering and ad-hoc document ranking. However, their effectiveness compared to feature-based Learning-to-Rank (LtR) methods has not yet been well-established. A major reason for this is because present LtR benchmarks that contain query-document feature vectors do not contain the raw query and document text needed for neural models. On the other hand, the benchmarks often used for evaluating neural models, e.g., MS MARCO, TREC Robust, etc., provide text but do *not* provide query-document feature vectors. In this paper, we present Istella22, a new dataset that enables such comparisons by providing both query/document text and strong query-document feature vectors used by an industrial search engine. The dataset consists of a comprehensive corpus of 8.4M web documents, a collection of query-document pairs including 220 hand-crafted features, relevance judgments on a 5-graded scale, and a set of 2,198 textual queries used for testing purposes. Istella22 enables a fair evaluation of traditional learning-to-rank and transfer ranking techniques on the same data. LtR models exploit the feature-based representations of training samples while pre-trained transformer-based neural rankers can be evaluated on the corresponding textual content of queries and documents. Through preliminary experiments on Istella22, we find that neural re-ranking approaches lag behind LtR models in terms of effectiveness. However, LtR models identify the scores from neural models as strong signals.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; **Test collections**.

KEYWORDS

Learning to Rank, Neural Information Retrieval, Evaluation

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531740

ACM Reference Format:
Domenico Dato, Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, and Nicola Tonellotto. 2022. The Istella22 Dataset: Bridging Traditional and Neural Learning to Rank Evaluation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3477495.3531740

1 INTRODUCTION

In recent years, interest in neural Learning-to-Rank (LtR) approaches based on pre-trained language models has grown. These techniques have been demonstrated to be very effective at various ranking tasks, such as question answering and ad-hoc document ranking. The main reason of this success is the ability of deep neural networks to understand complex language patterns and learn to extract features from text that allow them to match queries and documents by abstracting from their lexical representation. In the same time frame, feature-based LtR methods reached maturity, and research on this area focused primarily on specific aspects such as efficiency [10, 39, 45, 46, 74, 80], diversification [70], permutation-invariant models [62, 64]. An investigated topic in feature-based LtR was also how to reduce the performance gap between neural and ensemble-based models [66].

These two research areas progressed almost entirely disjointly and the effectiveness of neural LtR approaches compared to traditional feature-based LtR methods has not yet been well-established. A major reason that left the two areas well separated is the lack of publicly-available datasets enabling a direct comparison: LtR datasets providing query-document feature vectors do not contain the raw query and document text needed for neural models; viceversa, the benchmarks for evaluating neural models provide text but not query-document feature vectors. This resource paper bridges the gap between the two worlds by providing Istella22, the first comprehensive dataset including a common test set of both query/document text and strong query-document feature vectors used by an industrial search engine. Our aim is that the resources provided with Istella22 will help to advance research on the integration of text-based and feature-based LtR methods. In summary, the novel contribution of this paper is the following:

- We contribute the Istella22 dataset, a novel public resource consisting of a comprehensive corpus of 8.4M web documents, a collection of query-document pairs including 220 hand-crafted

features, relevance judgments on a 5-graded scale, and a common set of 2,198 textual queries used for testing purposes.
- We provide a detailed analysis of the resources made available in Istella22 compared to those previously available and used in the LtR research community.
- We report a preliminary comparison of the performance of traditional and neural re-ranking LtR methods applied to the web documents/queries made available in Istella22. Results show that neural re-ranking approaches lag behind traditional LtR models in terms of absolute performance. However, LtR models identify the scores from neural models as strong signals.

The rest of the paper is organized as follows. In Section 2 we discuss the state of the art methods for traditional and neural LtR methods. Section 3 details the Istella22 dataset and reports some statistics about its content, while Section 4 provides an comparison of Istella22 with respect to other publicly-available dataset in the two fields. Moreover, Section 5 discusses the utility and the practical implications of the new dataset and Section 6 presents a preliminary comparison of the performance of traditional and neural LtR techniques. Finally, 7 concludes the work and draws some future lines of investigation.

2 RELATED WORK

Learning-to-Rank (LtR) is a vast research area where several machine learning techniques have been proposed to rank the documents matching a query as established by a large supervised training set. These approaches solve the problem starting from query-document representations based on handcrafted features. More recently, new neural approaches have also shown to be effective in solving the task. In contrast with the older ones, some of these techniques exploit the text of both the query and the document directly to extract meaningful features and compute the relevance of the query w.r.t. to a document, e.g., pre-trained transformers [30]. In the following, we present a brief overview of the two families of techniques: traditional learning-to-rank methods based on handcrafted features and neural text-based approaches.

Learning to Rank with feature-based representations. Effective LtR algorithms have been proposed in the past to train complex models able to precisely rank the documents matching a query [41]. These algorithms learn a ranking model from a ground truth containing several examples consisting of a feature-based representation of a query-document pair and an associated relevance score to be predicted. RankNet [6] leverages a probabilistic ranking framework based on a pairwise approach to train a neural network. The difference between the predicted scores of two different documents is mapped to a probability by mean of the sigmoid function. Hence, using the cross-entropy loss this probability is compared with the ground truth labels, and Stochastic Gradient Descent (SGD) is used to minimize this loss. FRank [75] exploits a generative additive model and substitutes the cross-entropy loss with the fidelity loss, a distance metric adopted in physics, superior to cross-entropy when applied on top of the aforementioned probabilistic framework since 1) has minimum in zero, 2) is bounded in $[0, 1]$. Neither RankNet or FRank directly optimize a ranking metric, e.g., NDCG, and this discrepancy weakens the power of the model. Since ranking metrics are flat and discontinuous, their optimization within the loss function is troublesome. To overcome this issue, LambdaRank [8] heuristically corrects the RankNet gradients by exploiting the rank position of the document in the overall sorting: it multiplies the RankNet gradient with a term that measure the increase in NDCG when switching the terms, generating the so called λ-gradients. State-of-the-art LtR models include those based on additive ensembles of regression trees learned by Multiple Additive Regression Trees (MART) [31] and λ-MART [7, 78] gradient boosting algorithms. λ-MART [7] combines the successful training methodology provided by λ-gradients with MART. Currently, ensemble of regression trees are the most effective solution among LtR techniques when dealing with handcrafted features. Since such ranking models are made of hundreds of additive regression trees, the tight constraints on query response time require suitable solutions able to provide an optimal trade-off between efficiency and ranking quality [10, 32, 45].

Neural Ranking with text representations. In contrast with feature-based learning-to-rank, neural ranking approaches often encode and compare query and document text directly using neural networks to estimate relevance. Early work used binary character n-gram occurrences [35] and static word embeddings [33] as inputs with some success, but considerable improvements were made once contextualized embeddings (e.g., such as those produced by transformer-based language models like BERT [30]) were introduced (e.g., [55, 59]). The first works in this area focused on jointly modeling the query/document representations and relevance scores, and have been shown to generalize well across datasets/domains (e.g., [50, 81]) and languages (e.g., [54, 71]). Despite the demanding computational costs of these models at query time [34], the costs can still be managed when such systems are deployed at scale [77]. Nonetheless, there has been a strong focus recently to improve the efficiency of these models, mostly by pre-computing representations [53], performing approximate nearest neighbor searches over these representations (e.g., [36, 38]), and through learned query/document rewriting (e.g., [28, 61]).

Neural Ranking with feature-based representations. Some recent work in LtR aims at defining novel neural networks addressing the ranking problem by learning a model from a ground truth based on query-document vectors of handcrafted features. In this line, recent approaches exploit *attention* [76] to learn effective neural networks for ranking. Pasumarthi *et al.* propose ATTN-DIN, a new approach that exploits self-attention item interaction networks for ranking under the multivariate scoring paradigm [64]. ATTN-DIN can automatically learn permutation-equivariant representations, i.e., the scores it produces do not depend from the position of the items in the input, to capture item interactions without any auxiliary information. A second contribution exploiting neural networks and attention is SETRANK by Pang *et al.* [62]. SETRANK is a self-attention network that satisfies the permutation-equivariant requirement. SETRANK is able to capture both the local context information from the cross-item interactions and to learn permutation-equivariant representations for the items. More recently, Qin *et al.* provides a novel architecture that exploits self-attention to model list-wise ranking data as context [66]. Authors also propose to use latent cross [64], i.e., the concatenation between item feature and context feature, to effectively generate the interaction of each item and its list-wise context. Despite their novelty, the above techniques do not

Table 1: Statistics of the Istella22 datasets.

Collection		
	# Documents	8,421,456
Test Set	# Text queries	2,198
	# Relevant Documents	10,693
	# Query-Document Vectors	1,501,704
Train Set	# Queries	44,249
	# Relevant Documents	206,706
	# Query-Document Vectors	8,349,810
Validation Set	# Queries	11,005
	# Relevant Documents	51,362
	# Query-Document Vectors	2,062,424

improve ranking performance when compared with strong λ-MART implementations, as the ones provided by the LightGBM [37] and XGBoost [12] libraries. Recent work in the learning to rank field thus focus on filling the performance gap between state-of-the-art approaches based on ensembles of regression trees and neural networks [66].

Novelty of this work. To the best of our knowledge, this is the first contribution that presents a novel dataset bridging the gap in the evaluation of two parallel research lines, i.e., LtR techniques exploiting feature-based query-document representations and neural LtR approaches exploiting directly the textual content of both query and documents. Istella22 is a novel dataset composed of a document collection, a set of query-document vectors, and a common test set of textual queries with relevance judgements. The three parts together allows for a comprehensive experimental analysis of traditional and neural learning to rank techniques. We exploit the dataset to run a preliminary analysis of the performance of a well-known state-of-the-art LtR technique, namely λ-MART against techniques based on pre-trained transformers. We report on the preliminary results achieved and we outline some possible directions for future research enabled by the provided dataset.

3 THE ISTELLA22 DATASET

The new Istella22 dataset is a novel resource enabling a comprehensive evaluation of the performance of feature-based and text-based techniques for Learning to Rank. It is composed of three parts: i) a collection of multi-lingual web documents, ii) the text and feature-based representations of a common test set of queries and associated relevance judgements over documents belonging to the collection, and iii) the feature-based representation of train and validation sets derived from the collection. Table 1 reports some statistics of the dataset.

3.1 Document Collection

The document collection provided with the Istella22 dataset is produced starting from a bigger collection of 5B documents crawled in the past by the Italian Istella search engine[1]. We employ an available set of queries sampled from the search engine query log to retrieve, for each query, a list of documents from the Istella production backend. The documents collected in this way contribute to the definition of the novel collection of 8.5M multi-lingual unique web documents. Table 2 reports the top-5 languages of the documents. More than 50% of the collection consists of documents written in Italian, while about 30% of it are in English, 1% of it are in German and, 3% of it are in Spanish and French, respectively. Other languages account for about 5% of the collection. The language of the documents have been identified using the `Compact Language Detector` (CLD) v3 library[2]. The CLD library was not able to assign a language for 3% of the documents in the collection.

Table 2: Top-5 languages in the document collection.

Language	# Documents	%
Italian	4,575,131	54%
English	2,570,253	31%
Spanish	266,660	3%
French	213,129	3%
German	112,916	1%
Other languages	423,608	5%
Unknown	259,759	3%

Available Resources. The document collection is released to the public as a single JSON file. All documents in the Istella22 dataset are already pre-processed. We remove HTML tags by using the Istella internal document processing pipeline. For each document, we release seven fields: `doc_id`, `url`, `title`, `text`, `extra_text`, `lang`, and `lang_pct`. The seven fields provide the following information:

- `doc_id`: The document identifier.
- `url`: The URL of the original document.
- `title`: The title of the document extracted from the corresponding HTML `<title>` tag.
- `text`: The text of the document contained in the HTML `<body>` tag. In details, the text in this field comes from HTML `<p>` tags contained in the body of the document.
- `extra_text`: The text of the document contained in the HTML `<body>` tag but outside HTML `<p>` tags.
- `lang`: The language detected by the CLD library.
- `lang_pct`: The confidence (%) of the language detection tool in assigning the language.

Figure 1 presents the distribution of the number of tokens in the three textual fields. As is typical, titles are relatively short (a median of 9 tokens) and roughly follow a normal distribution. The two text fields can get substantially longer and exhibit a long tail. The medians number of tokens for `text` and `extra_text` are 297 and 275 tokens, respectively. The `text` field has a sizable number of documents with no tokens (5.9%), meaning that they contain no text within paragraph tags.

3.2 Test Queries and Relevance Judgements

The Istella22 dataset provides a common test set of queries for experimenting feature and text-based LtR methods. These queries

[1] https://www.istella.it/en/

[2] https://github.com/google/cld3

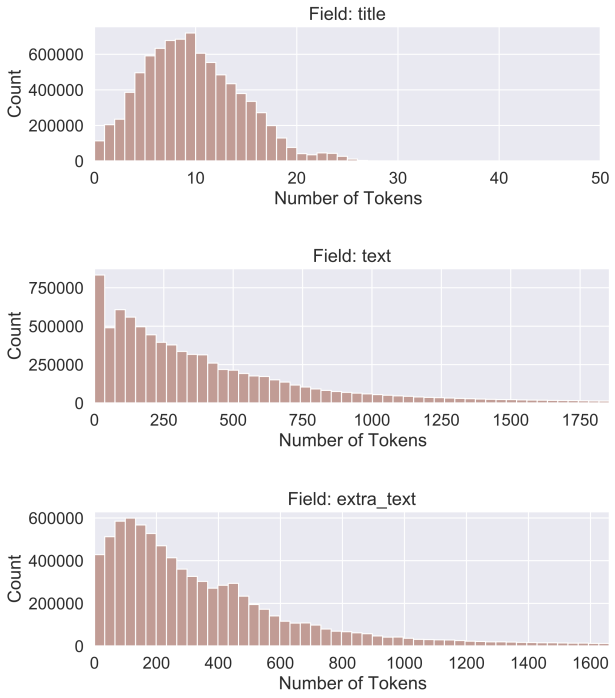

Figure 1: Distribution of the number of tokens in textual fields. For `text` and `extra_text`, we cap the distribution at the 95th percentile due to the long tail of values.

have been randomly sampled from a historical query log of the Istella search engine and come with human-assessed relevant documents judged by the company expert annotators. We remove single-term queries, adult content and sensitive personal information. We explicitly remove one-term queries to favor informational and transactional queries over navigational ones.

The resulting test set consists of 2,198 queries. As for documents, queries are mostly in Italian or English. Figure 2 reports the distribution of queries per number of terms in the query. In detail, queries made up of 2 and 3 terms account for about 70% of the test set, and the percentage raises up to 88.3% if we consider also queries of 4 terms.

The test queries come with a set of 10,693 relevance judgements. On average, each test query is associated with 4.8 relevant documents. Relevance judgements are distributed on a 5-graded scale ranging from 0 (irrelevant) to 4 (perfectly relevant).

Available Resources. Test queries are released in a JSON file, making them easy to use in various retrieval systems. For each query, we provide two fields: `query_id`, `text`. The former is a unique integer identifier of the query, and the latter is the text of the query (converted to lower case and with punctuation removed). Relevance judgements (qrels) are released as a separate TREC-formatted qrels file[3]. Each line in the file maps a query identifier and document identifier to an integer relevance score.

[3]https://trec.nist.gov/data/qrels_eng/

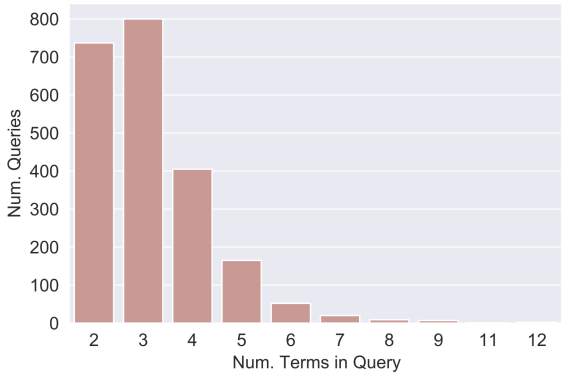

Figure 2: Distributions of the test queries per number of terms in the query.

3.3 Query-Document Feature Vectors

The Istella22 dataset comes with three sets of query-documents feature vectors for training, validation and test, where the feature-based test set exactly overlap with the text-based one. The three sets of query-document vectors are made up of 220 handcrafted features. The features released belong to four main categories: *query* features, *document* features, *query-document* features and *proximity-based* features. An example of a feature belonging to the first category is *query length*, i.e., the number of terms in the query. In the second category we have for example *pagerank*, i.e., the pagerank score associated with the document, while an example in the third category is the *BM25* score [68]. For the latter category, an example is the *longest common subsequence* [3], i.e., the length of the longest subsequence of query terms occurring in the document.

Query-document pairs have been selected starting by train, validation and test queries by using the Istella search engine infrastructure. The query processing pipeline of Istella is made up of two ranking stages where a first, fast and recall-oriented stage retrieves a list of candidate results that are then re-ranked by a second, machine-learned and precision-oriented ranker. Query-document vectors for test queries are generated for the top-1,000 documents in the Istella22 collection returned by the first-stage ranker. For each test query, we also add to this set the feature vectors corresponding to the missing relevant documents for the query if not retrieved by the first-stage ranker. The rationale of this choice is to avoid a possible loss of recall induced by the first-stage ranker. On the other side, feature vectors for train and validation queries are used in the training process of LtR models. For these queries, we generate shorter lists of feature vectors and the generation process employs negative sampling strategies [48].

Available Resources. The query-document feature vectors are released as three separate files for train, validation and test sets, respectively. Feature vectors are encoded in SVM-Rank format[4]. In this format, each line of the file represents an assessed query-document pair. The first number of the line is the relevance of

[4]https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

the query-document pair, the second field starting with the `qid:` prefix is the query identifier and the following fields are pairs `index:value`, where `index` is the feature index and `value` is the feature value for the query-document pair.

3.4 Accessing the Istella22 dataset

The Istella22 dataset is made available to researchers[5] according to the conditions detailed in the included license agreement. The dataset is also integrated into the `ir_datasets` library [56], making it easily accessible in a variety of neural ranking toolkits (e.g. PyTerrier [57, 58], OpenNIR [49], and Capreolus [79]) or accessing the data in an *ad hoc* manner. In other words, aside from being accessible through the released data files, the dataset is accessible through Python code as:

```
import ir_datasets
dataset = ir_datasets.load('istella22/test')
for doc in dataset.docs:
  print(doc)
# IstellaDoc(doc_id='07489924', title="...", text="...", ...)
for query in dataset.queries:
  print(query)
# GenericQuery(query_id='263', text='calcio mercato')
for qrel in dataset.qrels:
  print(qrel)
# GenericQrel(query_id='263', doc_id='83436604' relevance=3)
```

4 COMPARISON WITH AVAILABLE DATASETS

Table 3 reports a summary of the web ad-hoc document ranking datasets[6] used in traditional and neural learning to rank. For each of them, we report the availability of text collections, i.e., queries and documents, the size of the corpus, and the main languages of the documents in the corpus. We also report the number of queries, qrels, and the number relevance grades labeled. The analysis shows that, before Istella22, the Information Retrieval community does not have a suitable test collection to compare traditional LtR methods with handcrafted features and text-based neural methods. In fact, before Istella22, the available resources provide only *either* query-document feature vectors or, alternatively, text queries and documents. Among datasets that provide hand-crafted feature vectors, recent and well-known datasets are the Istella ones, i.e., Istella LETOR [29], Istella-S LETOR [43, 44, 47] and Istella-X LETOR [48]. Those datasets are characterized by a significant number of queries and relevance judgements used to build query-document feature vectors. Older datasets in this line are the MSLR-WEB10K, the MSLR-WEB30K and the LETOR 4.0 all released by Microsoft [65]. Another important public resource is the Yahoo LtR Challenge released by Yahoo [11]. All these datasets do not provide textual resources at all.

The datasets that provide query and document text (suitable for neural ranking models) can broadly be classified as those that have many available queries but employed a shallow pooling technique (resulting in a low average number of qrels per query) and those that provide fewer queries but deeper pooling (resuling in a high average number of qrels per query). Among shallowly-pooled datasets, we observe that a heuristic is always employed to determine relevance, rather than direct human assessment. AOL-IA [51, 63] and Sogou-QCL [82] infer relevance based on (potentially noisy) clicks from query logs. MS MARCO [2] infers a document-level label based on assessments at a passage level — resulting in a "maximum passage bias" that can influence which methods are effective at the task [40]. Deeply-pooled datasets are often the result of shared tasks, like TREC and NTCIR, and feature human-assessed relevance at multiple grades. However, they trade off the number of available queries (at most 325, and often far fewer).

Istella22 fills multiple important gaps in available test collections. Most notably, it is the only available dataset that provides both query/document text *and* industrial-grade query-document feature vectors for the same items. This provides an important and missing bridge between these two burgeoning research lines. As far as query-document feature vectors are concerned, Istella22 comes with three sets of query-document feature vectors, i.e., train, validation and test, built starting from 57k real-world queries. In terms of number of queries used to build query-document feature vectors, Istella22 doubles the previous availability of data. Compared with existing text resources, it provides both a large number of available test queries and directly human-assessed relevance labels. Meanwhile, the pooling is deeper than text-based collections of comparable size (i.e., it has on average 4.9 qrels per query, rather than MS MARCO's 1.0). Finally, it contains linguistically-diverse text, featuring a large number of both Italian and English queries/documents (as well as other languages). This is an area of growing interest in the IR community (c.f., the TREC NeuCLIR track).

5 UTILITY AND PREDICTED IMPACT

The Istella22 dataset is a step ahead toward bridging together the evaluation of traditional learning to rank approaches working on handcrafted features with neural information retrieval approaches based on deep pre-trained transformers. We expect the Istella22 dataset being useful for many researchers and IR practitioners working in the application of machine/deep learning techniques to the ranking problem. In recent years, the information retrieval community spent important effort in devising new machine/deep learned strategies for solving the ranking problem. However, a comprehensive comparison of the two parallel research lines is still missing. The Istella22 dataset advances this well-estabilished research area by filling this gap. For this reason, we expect that the Istella22 dataset will impact a large research community as it provides a single common ground of evaluation built on real-world user queries and web documents. Moreover, as neural information retrieval is a recent and hot research area, we expect the Istella22 dataset to collect a significant increasing interest over time.

6 PRELIMINARY EXPERIMENTS

We now propose an evaluation of a state-of-the-art learning to rank technique employing handcrafted features, i.e., λ-MART , against neural solutions based on pre-trained transformers for ranking on

[5] http://quickrank.isti.cnr.it/istella22-dataset/

[6] We limit our exploration to datasets based on web documents because all available feature-based datasets fit this category. We acknowledge that a multitude of datasets are available in other domains — especially domains like news and scientific articles — but to the best of our knowledge, none provide industrial-grade features suitable for learning-to-rank alongside the text.

Table 3: Comparison between Istella22 and other web ranking datasets. Istella22 fills an important gap in existing datasets by providing multi-lingual document text, production-level LtR features, a test set of textual queries with associated multiple-graded relevance assessments per query and query-document feature vectors for training and validation. A ✓ under Text indicates where text is available for queries or documents. Feats. indicates the number of standard handcrafted features provided by the dataset (– indicates no standard features area available). Langs. specifies the two-digit language code(s) of documents present in the corpus. Corpus, Queries, and Qrels indicate the total number of documents, queries, and relevance assessments, respectively. Grades indicates the number of relevance grades present in the dataset, and Rel. Assessment gives a short description of how the relevance assessments were produced. Values marked with * indicate that data are not currently publicly available (e.g., from a secret held-out test set).

Dataset	Text	Feats.	Langs.	Corpus	Queries	Qrels (avg. per q)	Grades	Rel. Assessment
Istella22 (ours)	✓	220	it,en,+	8.4M				
- Train	–				44k	207k (4.7)	5	Human-assessed
- Validation	–				11k	51k (4.7)	5	Human-assessed
- Test	✓				2.2k	11k (4.9)	5	Human-assessed
Istella LETOR [29]	–	220	–	–	33k	388k (11.7)	5	Human-assessed
Istella-S LETOR [43]	–	220	–	–	33k	388k (11.7)	5	Human-assessed
Istella-X LETOR [48]	–	220	–	–	10k	46k (4.6)	5	Human-assessed
MSLR-WEB10K [65]	–	136	–	–	10k	576k (57.5)	5	Human-assessed
MSLR-WEB30K [65]	–	136	–	–	31k	1.8M (58.6)	5	Human-assessed
LETOR 4.0 [65]	–	46	–	–	2.4k	21k (8.7)	2	Human-assessed
Yahoo LtR Challenge (set1) [11]	–	700	–	–	30k	525k (17.5)	5	Human-assessed
Yahoo LtR Challenge (set2) [11]	–	700	–	–	6k	135k (22.5)	5	Human-assessed
AOL-IA [51, 63]	✓	–	en,+	1.5M	10M	19M (2.0)	1	Inferred from clicks
ClueWeb09 / TREC Web 09-12 [16–19]	✓	–	en	504M	200	84k (422)	3	Human-assessed
ClueWeb12 / TREC Web 13-14 [20, 21]	✓	–	en	733M	100	29k (289)	3	Human-assessed
ClueWeb12-b13 / NTCIR WWW 3 [69]	✓	–	en	52M	160	32k (202)	5	Human-assessed
.GOV2 / TREC TB 04-06 [9, 14, 15]	✓	–	en	25M	150	135k (902)	3	Human-assessed
.GOV / TREC Web 02-04 [22–24]	✓	–	en	1.2M	325	196k (603)	2	Human-assessed
MS MARCO v1 (document) [2]	✓	–	en	3.2M				
- Train	✓				367k	367k (1.0)	1	Inferred from passage
- Dev	✓				5.2k	5.2k (1.0)	1	Inferred from passage
- Eval	✓				5.8k	*	1	Inferred from passage
- TREC DL 19–20 [25, 26]	✓				88	288	4	Human-assessed
MS MARCO v2 (document) [27]	✓	–	en	12M				
- Train	✓				322k	332k (1.0)	1	Inferred from passage
- Dev1	✓				4.6k	4.7k (1.0)	1	Inferred from passage
- Dev2	✓				5.0k	5.2k (1.0)	1	Inferred from passage
- TREC DL 21 [27]	✓				57	13k (229)	4	Human-assessed
NTCIR WWW-4 [13]	✓	–	en	82M	50	*	*	Human-assessed
SogouT-16 / NTCIR WWW 3 [69]	✓	–	zh	1.2B	80	*	*	Human-assessed
Sogou-QCL [82]	✓	–	zh	5.4M	537M	7.7M (14)	5	Inferred from clicks

the novel Istella22 dataset. In detail, we compare the following state-of-the-art methods for ranking.

- Lexical retrieval: BM25 [68] and DPH [1].
- λ-MART: a λ-MART model [78] trained using the LightGBM library[7] [37]. The model is trained using the methodology described in Section 6.1.
- MONOT5 (transfer): MONOT5 [60] models trained on other datasets (testing zero-shot transfer).
- MONOT5 (tuned): MONOT5 [60] models tuned on Istella22 training samples.
- λ-MART$_{\text{MONOT5}}$: to test whether traditional LtR and neural ranking are complementary, this model uses a λ-MART model [78] with both the available 220 features and an additional neural ranking feature: the output of MONOT5 (tuned) is trained using the same methodology described in Section 6.1. We test two versions of the model: one that operates over the doucment's title, URL, and text, and one that omits the text (which is faster to compute).

[7] https://github.com/microsoft/LightGBM

6.1 Experimental Settings

Lexical Retrieval. As a base point of comparison, we test two unsupervised lexical ranking models: BM25 [68] and DPH [1]. Both

models rank lexical matches over a sparse index that includes both title and text contents of the documents. We use the PyTerrier [58] toolkit for performing indexing and retrieval. We test both a version of BM25 using default $k1$ and b parameters, and a version that was tuned using a grid search for NDCG@10 performance on the validation set ($k1$ in $[0.2, 2.0]$ with a step of 0.2; and b in $[0, 1]$ with a step of 0.1). DPH is a parameterless model, so no tuning can be conducted.

λ-MART. The training process of λ-MART andλ-MART$_{\text{MONOT5}}$ is controlled by several hyper-parameters targeting its generalization power and the training speed of the learning phase, while others controlling the shape of the trees. We performed hyper-parameter tuning by exploiting the HyperOpt library [4]. We optimized four learning parameters: `learning_rate` in $[0.0001, 0.3]$, `num_leaves` in $\{15, 510\}$, `min_sum_hessian_in_leaf` in $[1, 100]$, and `min_data_in_leaf` in $\{1, 500\}$. The hyper-parameters of the models are tuned on the validation set according to the NDCG@10 metric. We learn the three models by training 5,000 trees at most. To avoid overfitting, we employ early stopping to stop the training process when there is no improvement on the validation set for 100 consecutive iterations. The optimal λ-MART learned has 1,514 trees, 390 leaves and a learning rate of 0.117, while the optimal λ-MART$_{\text{MONOT5}}$ (using title and URL) learned has 1,951 trees, 375 leaves and a learning rate of 0.051. Moreover, the optimal λ-MART$_{\text{MONOT5}}$ (using title, URL, and text) learned has 1,523 trees, 420 leaves and a learning rate of 0.112.

MONOT5. One of the most prominent neural ranking models is monoT5 [60]. This cross-encoder model jointly encodes query text and passage (or truncated document) text in a pre-trained sequence-to-sequence model, T5 [67]. It is tuned to predict the token `true` or `false`, following a sequencey including the query, document text, and a prompt of `Relevant:`. It uses the probability of the token `true` as the ranking score. We test this type of model in a variety of settings on Istella22. First, we test two *transfer* models, which are trained on alternative datasets: (1) a version trained on MSMARCO [2], using a checkpoint released by [60][8], and (2) a version trained on machine translations of MS MARCO (called mMARCO) to a variety of languages (including Italian), released by [5][9]. Exploring the neural IR models in transfer settings is the primary application of Istella22 to neural IR, considering well-known deficiencies of neural ranking models to transfer domains [73]. Although the Istella22 training query text are proprietary and cannot be released, we also test two T5 models *tuned* the Istella22 train set. These experiments are designed to show the *potential* for neural models to better transfer to this dataset. The first includes both the title and text (truncated to the maximum length supported by the model), which is the same setting as the transfer models. The other version uses the title and URL, a setting of T5 recently observed to be effective on another web ranking task [51]. Both use a version of T5 that was pre-trained on Italian text[10], and perform fine-tuning with a learning rate of 5×10^{-5}, a batch size of 4, and early stopping based on performance on performance on the validation dataset (after 10 consecutive validation steps without an improvement to nDCG).

Evaluation Metrics. We evaluate the performance of the methods using four well-known IR metrics, i.e., Precision (P), NDCG, MRR and MAP made available by the `ir_measures` library [52]. We evaluate precision at three different cut-offs, i.e., $k \in \{1, 5, 10\}$. For what regards NDCG, we report the one employing exponential weighing of the relevance [6] and we evaluate at two different cut-offs, i.e., $k \in \{10, 20\}$.

Available Software. The source code and all the models used in our experiments are made publicly available to allow the reproducibility of the results[11].

6.2 Experimental Results

Table 4 presents the results of our initial experiments on Istella22. Starting with the lexical baseline ranking models, BM25 and DPH, we find that, in an absolute sense, both are able to do a remarkably reasonable job at ranking on this dataset. They are able to rank at least a partially-relevant document at the top position in 43–44% of the time. However, as evidenced by the relatively low performance in terms of NDCG@10 and NDCG@20, these models are not able to effectively distinguish among relevance grades in the top positions. With proper tuning of BM25 parameters, however, relevance grades can be more effectively distinguished.

Moving on to the re-ranking models, we find that λ-MART is able to rank documents much more effectively than the simple lexical retrieval approaches. Indeed, P@1 is over 95%, and NDCG values are in the 80% range. This is likely, at least in part, due to the fact that the same features are used by the ranking models in the Istella engine itself; to an extent, the model is able to learn to mimic the documents that annotators were shown. Despite the high performance, there is still room for further improvement, especially among the most relevant documents.

As expected, the MONOT5 neural re-ranking models are able to improve significantly above simple lexical retrieval models. However, we found that model transfer from either MS MARCO or its multi-language version were not nearly as effective as one that uses in-domain training. This suggests that further work is needed to train highly-effective transfer models to general web search. Mirroring recent observations on the AOL query log [51], we observe that including the URL as additional text for MONOT5 can improve performance. In fact, using only the title and the URL performs marginally better than just the title and the text. However, using all three fields performs better than both other ablations.

Finally, to see whether or not neural ranking features can further enhance λ-MART 's ranking ability, we include the two best-performing variants as additional features (λ-MART$_{\text{MONOT5}}$). Curiously, we see almost no difference in ranking effectiveness when including these features. Only the MAP of λ-MART$_{\text{MONOT5}}$ (when using title, url, and text) exhibits a statistically significant difference in performance (two-sample Fisher's randomization test [42, 72] with a significance level set to $p < 0.05$). However, upon inspecting λ-MART$_{\text{MONOT5}}$'s feature importance, we see that the MONOT5 features are the most important to the model. Specifically, the version

[8]Huggingface model: `castorini/monot5-base-msmarco`
[9]Huggingface model: `unicamp-dl/mt5-base-mmarco-v2`
[10]Huggingface model: `gsarti/it5-base`

[11]https://github.com/hpclab/istella22-experiments

Table 4: Performance of baseline retrieval and re-ranking systems. Re-ranking systems operate over the initial ranked list from Istella, and include LtR, neural-reranking, and hybrid LtR-using-neural-reranking systems.

Method	Feats.	Neural Text	P@1	P@5	P@10	NDCG@10	NDCG@20	MRR	MAP
Retrieval									
BM25 (default)	-	–	0.4331	0.2939	0.2055	0.2280	0.2447	0.5439	0.3649
BM25 (tuned)	-	–	0.4339	0.2947	0.2055	0.3854	0.4207	0.5494	0.3686
DPH	-	–	0.4408	0.2868	0.2020	0.2281	0.2443	0.5479	0.3618
Re-Ranking									
λ-MART	✓	–	0.9559	0.7245	0.4609	0.8188	0.8286	0.9724	0.8891
MONOT5-MSMARCO	–	Ttl+Txt	0.5568	0.3893	0.2699	0.2990	0.3157	0.6675	0.4889
MONOT5-MMARCO	–	Ttl+Txt	0.5868	0.4147	0.2829	0.3175	0.3338	0.6976	0.5203
MONOT5-tuned	–	Ttl+Txt	0.8407	0.5813	0.3792	0.4418	0.4482	0.9005	0.7262
MONOT5-tuned	–	Ttl+Url	0.8412	0.5990	0.3914	0.4402	0.4472	0.9025	0.7396
MONOT5-tuned	–	Ttl+Url+Txt	0.8581	0.5945	0.3910	0.4515	0.4586	0.9132	0.7462
λ-MART$_{\text{MONOT5}}$	✓	Ttl+Url	0.9550	0.7223	0.4597	0.8152	0.8258	0.9716	0.8859
λ-MART$_{\text{MONOT5}}$	✓	Ttl+Url+Txt	0.9509	0.7238	0.4602	0.8153	0.8258	0.9701	0.8849

using just title and URL has an importance of 3.99% among the 221 features, while the version using title, URL, and text has an importance of 4.39%. This suggests that even though we do not observe an increase in performance when using these features, they do provide valuable signals to LtR models. We note that Istella22 is the only resource of its kind available to academic researchers to study interesting and important phenomena like this.

7 CONCLUSIONS AND FUTURE WORK

In this paper we proposed Istella22, a novel dataset bridging traditional and neural learning to rank evaluation. Istella22 includes three main parts, i.e., a multi-lingual web document collection, a test set of textual queries with multiple-graded relevance judgments and query document feature vectors. We detailed the dataset and the methodology employed to build it. We also discussed its possible impact on the information retrieval community as an unique resource allowing us to compare two parallel research lines on the same ground of evaluation. We reported about preliminary experiments conducted showing that, on this dataset, a classical LtR technique based on hand-crafted features outperforms state-of-the-art neural text-based re-ranking solutions, i.e., MONOT5. The analysis conducted and the peculiarities of the provided Istella22 resource suggest several lines of investigation, aimed at understanding and filling this performance gap: we leave such research as future work for us and all the IR community working in the field. For example, interesting open questions remain about the best ways to transfer neural ranking models to web search and how to best utilize signals from neural models in LtR models; Istella22 enables such research.

Acknowledgments. We thank Francesca Fallucchi and Ernesto William De Luca from Marconi University for the suggestions and fruitful discussions.

REFERENCES

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic Models of Information Retrieval based on Measuring the Divergence from Randomness. *ACM Trans. Inf. Sys.* 20, 4 (2002), 357–389.
[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proc. InCoCo@NIPS Workshop*.
[3] Lasse Bergroth, Harri Hakonen, and Timo Raita. 2000. A Survey of Longest Common Subsequence Algorithms. In *Proc. SPIRE*. IEEE, 39–48.
[4] James Bergstra, Daniel Yamins, and David Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proc. ICML*. PMLR, 115–123.
[5] Luiz Henrique Bonifacio, Israel Campiotti, Roberto Lotufo, and Rodrigo Nogueira. 2021. mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset. *arXiv:2108.13897* (2021).
[6] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank using Gradient Descent. In *Proc. ICML*.
[7] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMart: An overview. *Learning* (2010).
[8] Christopher J Burges, Robert Ragno, and Quoc V Le. 2007. Learning to Rank with Nonsmooth Cost Functions. In *Proc. NIPS*.
[9] Stefan Büttcher, Charles L. A. Clarke, and Ian Soboroff. 2006. The TREC 2006 Terabyte Track. In *Proc. TREC*.
[10] Gabriele Capannini, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. 2016. Quality versus Efficiency in Document Scoring with Learning-to-rank Models. *Inf. Proc. Man.* 52, 6 (2016), 1161–1177.
[11] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. In *Proc. the learning to rank challenge*. Proc. PMLR, 1–24.
[12] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proc. SIGKDD*. 785–794.
[13] Zhumin Chu, Yiqun Liu, Chen Nuo, Yujing Li, Junjie Wang, Tetsuya Sakai, Sijie Tao, Nicola Ferro, Maria Maistro, and Ian Soboroff. 2021. NTCIR We Want Web with CENTRE Task. http://sakailab.com/www4/ Task currently running.
[14] Charles L. A. Clark, Falk Scholer, and Ian Soboroff. 2005. The TREC 2005 Terabyte Track. In *Proc. TREC*.
[15] Charles Clarke, Nick Craswell, and Ian Soboroff. 2004. Overview of the TREC 2004 Terabyte Track. In *Proc. TREC*.
[16] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2009. Overview of the TREC 2009 Web Track. In *Proc. TREC*.
[17] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. 2010. Overview of the TREC 2010 Web Track. In *Proc. TREC*.
[18] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Ellen M. Voorhees. 2011. Overview of the TREC 2011 Web Track. In *Proc. TREC*.
[19] Charles L. A. Clarke, Nick Craswell, and Ellen M. Voorhees. 2012. Overview of the TREC 2012 Web Track. In *Proc. TREC*.
[20] Kevyn Collins-Thompson, Paul Bennett, Fernando Diaz, Charles L. A. Clarke, and Ellen M. Voorhees. 2013. TREC 2013 Web Track Overview. In *Proc. TREC*.
[21] Kevyn Collins-Thompson, Craig Macdonald, Paul Bennett, Fernando Diaz, and Ellen M. Voorhees. 2014. TREC 2014 Web Track Overview. In *Proc. TREC*.
[22] Nick Craswell and David Hawking. 2002. Overview of the TREC-2002 Web Track. In *Proc. TREC*.

[23] Nick Craswell and David Hawking. 2004. Overview of the TREC-2004 Web Track. In *Proc. TREC*.

[24] Nick Craswell, David Hawking, Ross Wilkinson, and Mingfang Wu. 2003. Overview of the TREC 2003 Web Track. In *Proc. TREC*.

[25] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 deep learning track. In *Proc. TREC*.

[26] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen Voorhees. 2019. Overview of the TREC 2019 deep learning track. In *Proc. TREC*.

[27] Nick Craswell, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Overview of the TREC 2021 deep learning track. In *Proc. TREC*.

[28] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. *ArXiv* abs/1910.10687 (2019).

[29] Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Fast Ranking with Additive Ensembles of Oblivious and Non-Oblivious Regression Trees. *ACM Trans. Inf. Syst.* 35, 2, Article 15 (2016), 31 pages.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[31] J. H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.

[32] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. 2016. Fast Feature Selection for Learning to Rank. In *Proc. SIGIR*. 167–170.

[33] J. Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. *Proc. CIKM* (2016).

[34] Sebastian Hofstätter and Allan Hanbury. 2019. Let's Measure Run Time! Extending the IR Replicability Infrastructure to Include Performance Aspects. *ArXiv* abs/1907.04614 (2019).

[35] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. *Proc. CIKM* (2013).

[36] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *ArXiv* abs/2004.04906 (2020).

[37] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proc. NeurIPS*. 3146–3154.

[38] O. Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *Proc. SIGIR* (2020).

[39] Francesco Lettich, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2019. Parallel Traversal of Large Ensembles of Decision Trees. *IEEE Trans. Par. Dist. Sys.* 30, 9 (2019).

[40] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *arXiv* abs/2008.09093 (2020). https://arxiv.org/abs/2008.09093

[41] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225–331.

[42] Claudio Lucchese, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, and Salvatore Trani. 2017. RankEval: An Evaluation and Analysis Framework for Learning-to-Rank Solutions. In *Proc. SIGIR* (Tokyo, Japan).

[43] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2016. Post-Learning Optimization of Tree Ensembles for Efficient Ranking. In *Proc. SIGIR*. 949–952.

[44] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2018. X-CLEaVER: Learning Ranking Ensembles by Growing and Pruning Trees. *ACM TIST* 9, 6 (2018), 1–26.

[45] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2015. QuickScorer: A Fast Algorithm to Rank Documents with Additive Ensembles of Regression Trees. In *Proc. SIGIR*.

[46] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Exploiting CPU SIMD Extensions to Speed-up Document Scoring with Tree Ensembles. In *Proc. SIGIR*. 833–836.

[47] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2017. X-DART: Blending Dropout and Pruning for Efficient Learning to Rank. In *Proc. SIGIR*. 1077–1080.

[48] Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Salvatore Orlando, and Salvatore Trani. 2018. Selective Gradient Boosting for Effective Learning to Rank. In *Proc. SIGIR*. 155–164.

[49] Sean MacAvaney. 2020. OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline. In *Proc. WSDM*. 845–848.

[50] Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2020. SLEDGE-Z: A Zero-Shot Baseline for COVID-19 Literature Search. In *Proc. EMNLP*.

[51] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Reproducing Personalised Session Search over the AOL Query Log. In *Proc. ECIR*.

[52] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Streamlining Evaluation with ir-measures. In *Proc. ECIR*.

[53] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient Document Re-Ranking for Transformers by Precomputing Term Representations. *Proc. SIGIR* (2020).

[54] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. 2020. Teaching a New Dog Old Tricks: Resurrecting Multilingual Retrieval Using Zero-shot Learning. In *Proc. ECIR*. 246–254.

[55] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proc. SIGIR*.

[56] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *Proc. SIGIR*.

[57] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation in Information Retrieval Using PyTerrier. In *Proc. ICTIR*. 161–168.

[58] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In *Proc. CIKM*.

[59] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *ArXiv* abs/1901.04085 (2019).

[60] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Fidnings of EMNLP*.

[61] Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *ArXiv* abs/1904.08375 (2019).

[62] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval. In *Proc. SIGIR*. 499–508.

[63] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A Picture of Search. In *Proc. InfoScale*.

[64] Rama Kumar Pasumarthi, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2020. Permutation Equivariant Document Interaction Network for Neural Learning to Rank. In *Proc. ICTIR*. 145–148.

[65] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* abs/1306.2597 (2013). http://arxiv.org/abs/1306.2597

[66] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *Proc. ICLR*.

[67] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv* abs/1910.10683 (2020).

[68] Stephen Robertson and Hugo Zaragoza. 2009. *The Probabilistic Relevance Framework: BM25 and Beyond.* Found. Trends Inf. Retr.

[69] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Y. Liu, Zhicheng Dou, and Ian Soboroff. 2020. Overview of the NTCIR-15 We Want Web with CENTRE (WWW-3) Task.

[70] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2013. Learning to Rank Query Suggestions for Adhoc and Diversity Search. *Inf. Retr.* 16, 4 (2013).

[71] Peng Shi, He Bai, and Jimmy J. Lin. 2020. Cross-Lingual Training of Neural Models for Document Ranking. In *Findings of EMNLP*.

[72] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proc. CIKM*.

[73] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. abs/2104.08663 (2021).

[74] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. 2018. Efficient Query Processing for Scalable Web Search. *Found. Trends Inf. Ret.* 12, 4–5 (2018).

[75] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. 2007. FRank: a Ranking Method with Fidelity Loss. In *Proc. SIGIR*.

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proc. NeurIPS*, Vol. 30. 5998–6008.

[77] Yu Wang, Jinchao Li, Tristan Naumann, Chenyan Xiong, Hao Cheng, Robert Tinn, Cliff Wong, Naoto Usuyama, Richard Rogahn, Zhihong Shen, Yang Qin, Eric Horvitz, Paul Bennett, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Pretraining for Vertical Search: Case Study on Biomedical Literature. In *Proc. SIGKDD*.

[78] Q. Wu, C.J.C. Burges, K.M. Svore, and J. Gao. 2010. Adapting Boosting for Information Retrieval Measures. *Information Retrieval* (2010).

[79] Andrew Yates, Siddhant Arora, Xinyu Zhang, Wei Yang, Kevin Martin Jose, and Jimmy J. Lin. 2020. Capreolus: A Toolkit for End-to-End Neural Ad Hoc Retrieval. *Proc. WSDM* (2020).

[80] Ting Ye, Hucheng Zhou, Will Y. Zou, Bin Gao, and Ruofei Zhang. 2018. RapidScorer: Fast Tree Ensemble Evaluation by Maximizing Compactness in Data Level Parallelization. In *Proc. SIGKDD*. 941–950.

[81] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy J. Lin. 2019. Applying BERT to Document Retrieval with Birch. In *Proc. EMNLP*.

[82] Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-QCL: A New Dataset with Click Relevance Label. In *Proc. SIGIR*.