ESTUCeng at the NTCIR-15 WWW-3 Task: Experimenting with new document quality features

Ahmet Aydın*
Computer Engineering Department
Eskişehir Technical University, Turkey
ahmetaydin@eskisehir.edu.tr

Gökhan Göksel
Computer Engineering Department
Eskişehir Technical University, Turkey
gciplak@eskisehir.edu.tr

Ahmet Arslan
Computer Engineering Department
Eskişehir Technical University, Turkey
aarslan2@eskisehir.edu.tr

Bekir Taner Dinçer
Computer Engineering Department
Muğla Sıtkı Koçman University, Turkey
dtaner@mu.edu.tr

ABSTRACT

The ESTUCeng team participated in the English subtask of the We Want Web Task (WWW-3) [30]. This paper describes our approach to tackle the ad-hoc Web search problem and discusses the results. We used LambdaMART, a learning to rank algorithm, to re-rank samples generated by Language Modeling with Dirichlet smoothing. We extracted several traditional features adopted from the literature as well as a family of new HTML document quality features for the ClueWeb12-B13 dataset. The traditional features are augmented with the newly proposed features. Then, we used feature selection to obtain an optimum subset of features that would produce the highest retrieval effectiveness. The query relevance judgements of the NTCIR-13 We Want Web-1 [21] and the NTCIR-14 We Want Web-2 [25] are used as training data. The novel document quality features (type-D) proved to be useful in achieving a competitive retrieval effectiveness for the ClueWeb12-B13 dataset.

KEYWORDS

Document prior, query independent feature, learning to rank.

TEAM NAME

ESTUCeng

SUBTASKS

We Want Web with CENTRE (WWW-3) (English)

1 INTRODUCTION

The ESTUCeng team participated in the English subtask of the NTCIR-15 WWW-3 Task [30] by applying a learning to rank approach. Learning to rank is the application of machine learning to the document ranking problem. Thus, feature engineering is a key component of learning to rank since instances are represented by features. Therefore, many studies introduce several Query-Dependent (QD) and Query-Independent (QI) features in addition to the standard set of features (the features that the most commonly used datasets contain [28]) to achieve better retrieval effectiveness. We implemented existing features and proposed a new set of QI features. We used Language Modeling with Dirichlet smoothing (LM.DIR) [35] to obtain the sample, which is then re-ranked by LambdaMART [8]. We submitted three runs with different subsets of features. The following sections describe our methodology in detail and discuss our submission results.

*The novel document quality features experimented in this work are stemmed from Ahmet Aydın's ongoing PhD thesis and have not been published before.

2 LEARNING TO RANK METHODOLOGY

In this section, we briefly describe our methodology on building a learning to rank model on the training set and using it to re-rank the test set.

A learning to rank framework requires a sampling stage in both learning a model (training) and applying the learned model (testing) [24]. The sample is obtained by a static term-weighting model such as BM25. The term-weighting model used in sampling is called *sample model* or *reference model* [20].

2.1 Sampling (top-k retrieval)

For learning to rank, usually BM25 is used for sampling [24]. However, in this study for the selection of the sample model we experimented with 8 different term-weighting models: BM25 [29], DFIC [18], DFRee [2], LM.DIR [35], DLH13 [22], DPH [1], LGD [10], PL2 [3]. We use the default values of the hyper-parameters of the models if any.

We also created three indices of the ClueWeb12-B13 dataset for evaluating the effectiveness of two different stemming algorithms: KStem [19] and Porter Stemming [27].

We used top-k=1000 as the sample size. Our training query set is comprised of 180 queries from the previous two NTCIR We Want Web English subtasks [21, 25].

To select one out of 24 combinations we used a specific effectiveness metric as given by Eq. 1 to quantify the fraction of explicitly judged documents retrieved by the minimum of the number of documents retrieved and the value of k. We could use "the total number of judged documents" in the denominator of the equation, however this would not affect the relative ranking of the sample models.

Unlike the conventional effectiveness measures (e.g., Recall, MAP [7], nDCG [17], ERR [9], etc), this metric takes into account the documents judged as non-relevant by assessors. We choose to optimize this metric for the selection of the term-weighting model because we believe that a learning to rank method may produce better results when the training data include non-relevant documents as well as relevant documents. In other words, *explicitly* judged

Table 1: The effectiveness scores (as measured by the fraction of *explicitly judged* documents in the top-k=1000 documents retrieved) of stemming algorithms and term-weighting models in descending order for the WWW-1 and WWW-2 query sets. The value in boldface, attained by LM.DIR with KStem, presents the highest score.

Sample Model	NoStem	KStem	Porter
LM.DIR (μ=2500)	0.1953	**0.2269**	0.2137
DFIC	0.1800	0.2052	0.1962
LGD (c=1.0)	0.1710	0.1937	0.1845
DPH	0.1695	0.1898	0.1798
DFRee	0.1673	0.1875	0.1779
PL2 (c=1.0)	0.1623	0.1833	0.1773
DLH13	0.1637	0.1823	0.1730
BM25 (k=1.2, b=0.75)	0.1566	0.1750	0.1684

non-relevant documents might be useful in the sample list that is to be used in learning a re-ranking model. Recall that documents that are not explicitly judged in the sample list are assumed as non-relevant during training. However, these documents might well be relevant. This is a known limitation of the reusability of standard information retrieval benchmark datasets [5, 6].

Table 1 presents effectiveness scores of 24 different combinations. The maximum score is attained by LM.DIR with KStem combination; thus, we retrieved our top-k documents by applying LM.DIR to the WWW-1 and WWW-2 English query sets on the ClueWeb12-B13 collection indexed with KStemming employed. To test our approach, we applied the same combination (LM.DIR+KStem) to the WWW-3 English topics on the same collection.

It is interesting to note that although BM25 is usually preferred as the reference model in the literature, it is the worst when our metric is optimized. In other words, LM.DIR returns more explicitly judged (either relevant or non-relevant) documents in the result list than BM25. This observation calls attention to further investigation of the role of sample model in learning to rank settings. What is the ideal criterion for the selection of the sample model: Is it a list-based metric (nDCG@1000, nDCG@20) or a set-based metric (Recall@1000 or P@1000)? Would different learning to rank algorithms benefit from different selection criteria? To our best knowledge, these questions are still unanswered.

$$\text{Effectiveness} = \frac{\#\ Explicitly\ Judged\ Documents\ Retrieved}{\min\{k = 1000,\ \#\ Documents\ Retrieved\}} \quad (1)$$

2.2 Feature Extraction

We examined previous learning to rank studies to gather as many features as possible. Many of the studies experiment on standard datasets such as LETOR [28] and MSLR [28], while some others [4, 37] introduce additional features to improve the retrieval effectiveness. The complete list of features used in this study is listed below. We use the categorization scheme proposed in [20], which categorizes features into three classes: (*i*) query features (Type-Q), (*ii*) query-document pair features (Type-QD) and (*iii*) document features (Type-D). Table 2 presents existing features introduced in the literature. We also introduced novel document quality indicators as Type-D features. Table 3 presents our introduced features.

2.3 Feature Selection

We extracted all features for query-document pairs on training data, which is described in Section 2.1 to obtain the training set. Our learning to rank setup is described as follows. We used 5-fold cross validation on training set by 3 fold for training, 1 fold for testing and 1 fold for validation. Then we used the `JForests`[1] implementation of LambdaMART [8] for training and obtaining prediction scores. We re-ranked our sample by prediction scores and evaluated our results by calculating nDCG@10 with `gdeval`[2].

We applied heuristic feature selection methods and eliminated some of the features on the training set to improve the retrieval effectiveness. First we obtained the ablation scores for each feature in the full feature set (F). A feature ablation score of f_i is the nDCG@10 score of $F - f_i$. Then we applied the following methods to obtain the best subset of features.

1- Add one feature at a time to an empty set by ablation score (ascending sorted) and pick the subset of features with the best nDCG@10 score.

2- Ablate one feature at a time from the full feature set by ablation score (descending sorted). If the current nDCG@10 score of the subset is less than the previous subset then skip the feature.

2.4 Runs and Evaluation

We submitted 3 runs for the English subtask. We extracted the features described and selected the best subsets of features based on methods in Section 2.3. Then we submit our 3 runs, based on the following 3 feature selection methods that produce promising results on the WWW-1 and WWW-2 English tracks.

ESTUCeng-E-CO-NEW-1 : Features are selected based on Method 2 in Section 2.3. Similarity features in Table 3 are calculated by cosine similarity.

ESTUCeng-E-CO-NEW-2 : Features are selected based on Method 2 in Section 2.3. Similarity features in Table 3 are calculated by semantic similarity [32].

ESTUCeng-E-CO-NEW-3 : Features are selected based on Method 1 in Section 2.3. Similarity features in Table 3 are calculated by semantic similarity [32].

Table 4 presents our *offline evaluation* results on the training set. ESTUCeng-E-CO-NEW-1 produces the best results by selecting

[1] https://code.google.com/archive/p/jforests/
[2] https://trec.nist.gov/data/web/12/gdeval.pl

Table 2: Existing Features.

Feature Type	Feature Name
Type-Q	WordCount, Gamma, Omega, AvgPMI, SCS [15, 16]
Type-Q	Mean and Variance of IDF, CTI, Skewness, Kurtosis, SCQ [15, 16, 36]
Type-QD	TF Sum, Min, Max, Mean, Variance for 5 fields [28]
Type-QD	TF-IDF Sum, Min, Max, Mean, Variance for 5 fields [28]
Type-QD	Sum, Min, Max, Mean, Variance of term counts divided by text length [28]
Type-QD	BM25, DFIC, DFRee, DLH13, DPH, LGD, PL2, LM.JM, LM.ABS, LM.DIR for 5 fields [23, 24, 28]
Type-QD	Covered query term number [28]
Type-QD	Covered query term ratio [28]
Type-QD	Minimum coverage, in the title of the document, of all query terms [12]
Type-QD	Minimum coverage, in the body of the document, of all query terms [12]
Type-D	Number of child pages, Inlink count, Outlink count, PageRank [28]
Type-D	SpamScore [11]
Type-D	Entropy, URL Depth, Average term length, Fraction of anchor text, Fraction of table text, Number of title terms, Stopword coverage, Stopword ratio, Text/Document length [4]
Type-D	URLWiki [24]
Type-D	CDD [37]

Table 3: Introduced Features.

Feature Type	Feature Name	Feature Type	Feature Name
Type-D (Boolean)	Contact Info	Type-D	The mean of boolean features
Type-D (Boolean)	Content Length over 1800 words	Type-D	Fraction of images with alt tag text to all images
Type-D (Boolean)	Copyright	Type-D	Number of words in content
Type-D (Boolean)	Description	Type-D	Fraction of number of words in headings to number of words in content
Type-D (Boolean)	Favicon	Type-D	Number of images
Type-D (Boolean)	Https	Type-D	The minimum of the indices of keywords in title
Type-D (Boolean)	Keywords	Type-D	Fraction of (innerlink – outlink) to all links
Type-D (Boolean)	Keywords in Domain	Type-D	URL length
Type-D (Boolean)	Keywords in First 100 words	Type-D	Number of MetaTags
Type-D (Boolean)	Keywords in image alt tag text	Type-D	Fraction of no-follow links to all links
Type-D (Boolean)	Keywords in title	Type-D	Similarity of description and headings
Type-D (Boolean)	Robots.txt	Type-D	Similarity of description and content
Type-D (Boolean)	Social media share	Type-D	Similarity of description and keywords
Type-D (Boolean)	Viewport	Type-D	Similarity of description and title
		Type-D	Similarity of title and content
		Type-D	Similarity of title and headings
		Type-D	Similarity of title and keywords
		Type-D	Similarity of content and headings
		Type-D	Similarity of content and keywords
		Type-D	Similarity of keywords and headings

all features in Table 2 and Table 3 except the variance of Kurtosis. It is interesting that ESTUCeng-E-CO-NEW-3 yields effectiveness scores close to ESTUCeng-E-CO-NEW-1 by eliminating 43 features (41 of them are in Table 2). It should be noted that ESTUCeng-E-CO-NEW-1 and ESTUCeng-E-CO-NEW-3 use different methods (Method 2 and Method 1 in Section 2.3, respectively) on selecting the best feature subsets. It should also be noted that ESTUCeng-E-CO-NEW-1 keeps the initial order of features while ESTUCeng-E-CO-NEW-3 re-orders features by ablation scores.

Figure 1 shows a comparison of ESTUCeng-E-CO-NEW-3 vs. LM.DIR and LTR-with-Standard-Features using the *offline evaluation*. In LTR-with-Standard-Features, we re-rank LM.DIR samples by LambdaMART using the features listed in Table 2. The figure empirically implies that the application of LambdaMART using existing features in the literature improves the retrieval effectiveness of LM.DIR term-weighting model. It also implies that LambdaMART augmented with our proposed query-independent features further improves the effectiveness at all cut-off values of nDCG systematically.

Table 4: The results for the WWW-1 and WWW-2 English query sets.

Run	nDCG@10	nERR@10
ESTUCeng-E-CO-NEW-1	0.35955	0.47492
ESTUCeng-E-CO-NEW-2	0.34136	0.44820
ESTUCeng-E-CO-NEW-3	0.35847	0.47583

Figure 1: The comparison of ESTUCeng-E-CO-NEW-3 vs. LM.DIR and LTR-with-Standard-Features.

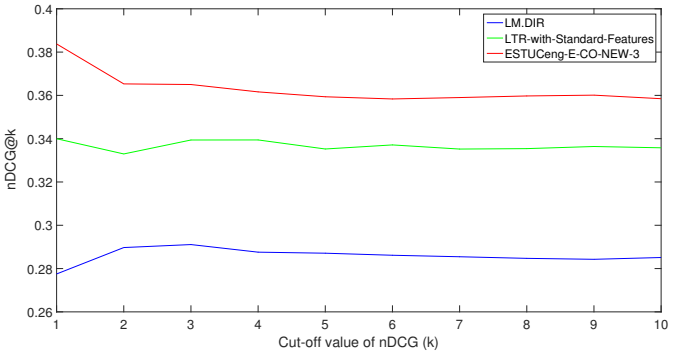
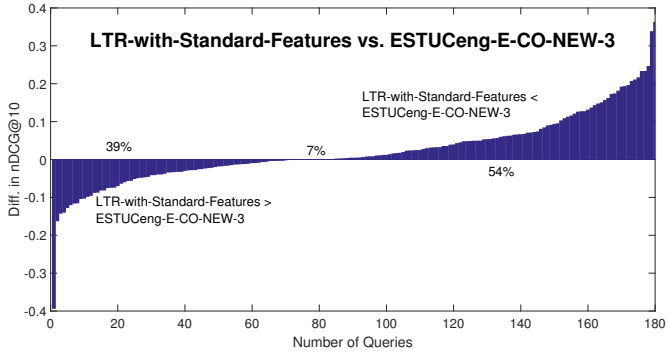


Figure 2 shows the per-query nDCG@10 score differences between LTR-with-Standard-Features and ESTUCeng-E-CO-NEW-3 using the *offline evaluation*. LTR-with-Standard-Features outperforms ESTUCeng-E-CO-NEW-3 on 39% of the total queries, while ESTUCeng-E-CO-NEW-3 outperforms LTR-with-Standard-Features on 54% of the total queries when the cut-off value of nDCG is set to 10. As noted by Peng *et al.* [26] not all queries equally benefit from the application of document prior features.

Figure 2 is also useful for evaluating the robustness of ESTUCeng-E-CO-NEW-3 with respect to LTR-with-Standard-Features. A system can be better than another on the average but it can cause abject failures for some topics. A robust system should satisfy each and every information need at least at a reasonable level [13, 34]. Considering only the delta between two values can hide important details, as illustrated in Figure 2. Although it is common practice to compare retrieval systems by averaging the system-query scores as in Figure 1, per-query comparison of systems provides insight into the reliability [14] of systems.

Figure 2: The per-query score differences between LTR-with-Standard-Features and ESTUCeng-E-CO-NEW-3.



3 RESULTS

Table 5 presents our English subtask results for the WWW-3 task. ESTUCeng-E-CO-NEW-1 and ESTUCeng-E-CO-NEW-3 results are close to each other while ESTUCeng-E-CO-NEW-2 results are apparently worse. According to the overview paper [30], there is no significant differences between our runs except ESTUCeng-E-CO-NEW-2 which is significantly worse than the rest (Also ESTUCeng-E-CO-NEW-2 results are worse than the BM25-baseline results provided by the organizers). Our best run (ESTUCeng-E-CO-NEW-3) is significantly better than 11 runs, while it outperforms 28 runs by nDCG scores.

4 DISCUSSION

We believe that neither web search nor learning to rank is a solved problem. The real web search employed by commercial search engines (e.g. Google) has many challenges that are difficult to simulate in the test collection based evaluation (i.e. the Cranfield paradigm [33]) of information retrieval systems [31]. Spam scores, anchor text, link graphs and HTML parsing are just the tip of the iceberg. Crawlers and searchers operate in real-life settings deal with crawl frequency, multi language web sites, 404 not found pages, load time, dwell time, content farming, high ad-to-content ratio pages, disavowing manipulative backlinks, generating rich snippets, accelerated mobile pages and so on.

Feature engineering for learning to rank is not solved either. Many features are just derivatives of each other. Especially a feature calculated for a word is aggregated for the query. Recall that a query is usually comprised of more than one word. Simple aggregation methods such as the minimum maximum average variance are used. But their meaning is unclear. What is the meaning of the maximum BM25 score of a query on the title field? This is indeed something difficult to interpret by a human. Furthermore, the mean length of typical Web search queries is 3-5 words. The aggregation statistics average and variance might not be meaningful for such a few data points.

5 CONCLUSIONS

We participated in the English subtask of the NTCIR-15 WWW-3 task [30] as ESTUCeng team. We introduced a novel set of QI features to quantify the HTML document priors, the probability that the document is relevant to *any* query [26], for the ClueWeb12-B13 dataset. The traditional features gathered from the existing learning to rank literature are augmented with the newly proposed features. We used different subsets of features, obtained by different feature selection methods, for our runs. The novel document prior features introduced in this paper proved to be useful in achieving a competitive retrieval effectiveness for the NTCIR-15 WWW-3. Furthermore, results show that a learning to rank system may re-rank

Table 5: Results of the NTCIR-15 WWW-3 according to the overview paper [30].

Run	nDCG@10	Q@10	nERR@10	iRBU@10
ESTUCeng-E-CO-NEW-1	0.6508	0.6638	0.7597	0.9163
ESTUCeng-E-CO-NEW-2	0.4991	0.5051	0.6524	0.8677
ESTUCeng-E-CO-NEW-3	0.6537	0.6644	0.7561	0.9161

documents poorly with the wrong features. Figure 2 reveals that although ESTUCeng-E-CO-NEW-3 outperforms LTR-with-Standard-Features on the average, it produces less effective results on the 39% of total queries. As a future work, we will investigate a selective retrieval approach that predicts the queries that will benefit from the application of our newly proposed document prior features.

REFERENCES

[1] Giambattista Amati. 2006. Frequentist and Bayesian Approach to Information Retrieval. In *Advances in Information Retrieval*. Lecture Notes in Computer Science, Vol. 3936. Springer Berlin Heidelberg, 13–24.

[2] Giambattista Amati. 2009. *Divergence from Randomness Models*. Springer US, Boston, MA, 929–932.

[3] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 357–389.

[4] Michael Bendersky, W Bruce Croft, and Yanlei Diao. 2011. Quality-biased ranking of web documents. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 95–104.

[5] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. 2006. Bias and the Limits of Pooling. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*. ACM, Seattle, Washington, USA, 619–620.

[6] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. 2007. Bias and the limits of pooling for large collections. *Information Retrieval* 10, 6 (26 Dec. 2007), 491–508.

[7] Chris Buckley and Ellen M. Voorhees. 2000. Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*. ACM, Athens, Greece, 33–40.

[8] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[9] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. ACM, Hong Kong, China, 621–630.

[10] Stéphane Clinchant and Éric Gaussier. 2011. Retrieval Constraints and Word Frequency Distributions a Log-logistic Model for IR. *Information Retrieval* 14, 1 (Feb. 2011), 5–25.

[11] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval* 14, 5 (2011), 441–465.

[12] Boris Dadachev. 2015. *On the Helmholtz principle for text mining*. Ph.D. Dissertation. Cardiff University.

[13] B. Taner Dinçer, Craig Macdonald, and Iadh Ounis. 2014. Hypothesis Testing for the Risk-sensitive Evaluation of Retrieval Systems. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '14)*. ACM, Gold Coast, Queensland, Australia, 23–32.

[14] Donna Harman and Chris Buckley. 2009. Overview of the Reliable Information Access Workshop. *Information Retrieval* 12, 6 (Dec. 2009), 615–641.

[15] Claudia Hauff, Leif Azzopardi, and Djoerd Hiemstra. 2009. The combination and evaluation of query performance prediction methods. In *European Conference on Information Retrieval*. Springer, 301–312.

[16] Ben He and Iadh Ounis. 2006. Query performance prediction. *Information Systems* 31, 7 (2006), 585–594.

[17] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446.

[18] İlker Kocabaş, Bekir Taner Dinçer, and Bahar Karaoğlan. 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information retrieval* 17, 2 (2014), 153–176.

[19] Robert Krovetz. 1993. Viewing Morphology As an Inference Process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*. ACM, Pittsburgh, USA, 191–202.

[20] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[21] Cheng Luo, Tetsuya Sakai, Yiqun Liu, Zhicheng Dou, Chenyan Xiong, and Jingfang Xu. 2017. Overview of the NTCIR-13 We Want Web Task. In *Proceedings of NTCIR-13*. 394–401. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings13/pdf/ntcir/01-NTCIR13-OV-WWW-LuoC.pdf

[22] Craig Macdonald, Ben He, Vassilis Plachouras, and Iadh Ounis. 2005. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *TREC*.

[23] Craig Macdonald, Rodrygo LT Santos, Iadh Ounis, and Ben He. 2013. About learning models with multiple query-dependent features. *ACM Transactions on Information Systems (TOIS)* 31, 3 (2013), 1–39.

[24] Craig Macdonald, Rodrygo L. T. Santos, and Iadh Ounis. 2013. The whens and hows of learning to rank for web search. *Information Retrieval* 16, 5 (2013), 584–628.

[25] Jiaxin Mao, Tetsuya Sakai, Cheng Luo, Peng Xiao, Yiqun Liu, and Zhicheng Dou. 2019. Overview of the NTCIR-14 We Want Web Task. In *Proceedings of NTCIR-14 (NTCIR-14)*. Tokyo, Japan, 455–467. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/01-NTCIR14-OV-WWW-MaoJ.pdf

[26] Jie Peng, Craig Macdonald, and Iadh Ounis. 2008. Automatic Document Prior Feature Selection for Web Retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. Association for Computing Machinery, Singapore, Singapore, 761–762.

[27] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

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

[29] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (April 2009), 333–389.

[30] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, and Ian Soboroff. 2020. Overview of the NTCIR-15 We Want Web with CENTRE (WWW-3) Task. In *Proceedings of NTCIR-15*. to appear.

[31] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends® in Information Retrieval* 4, 4 (2010), 247–375.

[32] Hideki Shima. 2014. WS4J WordNet Similarity for Java. https://github.com/mhjabreel/ws4j

[33] Ellen M. Voorhees. 2007. TREC: Continuing Information Retrieval's Tradition of Experimentation. *Commun. ACM* 50, 11 (Nov. 2007), 51–54.

[34] Lidan Wang, Paul N. Bennett, and Kevyn Collins-Thompson. 2012. Robust Ranking Models via Risk-sensitive Optimization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, Portland, Oregon, USA, 761–770.

[35] C. Zhai and J. Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)* 22, 2 (2004), 179–214.

[36] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *European conference on information retrieval*. Springer, 52–64.

[37] Yun Zhou and W Bruce Croft. 2005. Document quality models for web ad hoc retrieval. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. 331–332.