Performance Comparison of Ad-hoc Retrieval Models over Full-text vs. Titles of Documents

Ahmed Saleh[1,2] (✉), Tilman Beck[1], Lukas Galke[1,2], and Ansgar Scherp[1,3]

[1] Kiel University, Kiel, Germany
[2] ZBW – Leibniz Information Centre for Economics, Kiel, Germany
{a.saleh}@zbw.eu
[3] University of Stirling, United Kingdom

Abstract. While there are many studies on information retrieval models using full-text, there are presently no comparison studies of full-text retrieval vs. retrieval only over the titles of documents. On the one hand, the full-text of documents like scientific papers is not always available due to, e. g., copyright policies of academic publishers. On the other hand, conducting a search based on titles alone has strong limitations. Titles are short and therefore may not contain enough information to yield satisfactory search results. In this paper, we compare different retrieval models regarding their search performance on the full-text vs. only titles of documents. We use different datasets, including the three digital library datasets: EconBiz, IREON, and PubMed. The results show that it is possible to build effective title-based retrieval models that provide competitive results comparable to full-text retrieval. The difference between the average evaluation results of the best title-based retrieval models is only 3% less than those of the best full-text-based retrieval models.

Keywords: Information retrieval, Learning to rank, Deep Learning

1 Introduction

Using only titles has shown to be effective for document classification [1] and top-k recommendations [2]. This motivates us to investigate the possibility of building effective retrieval models based only on documents' titles. According to Croft et. al [3], there are four main categories of ranking models: (1) set theoretic models or Boolean models, (2) vector space models (e. g., TF-IDF), (3) probabilistic models (e. g., BM25), and (4) feature-based retrieval (e. g., L2R). Furthermore, there are recent advances in Deep Learning that provide neural network models capable of capturing the semantics of words. We employ representative examples of retrieval models from these categories and compare them regarding their performance over full-text vs. title. For this purpose, we utilize five datasets, out of which three are obtained from digital libraries: PubMed, Econbiz and IREON, and two standard test collections [4]: NTCIR-2 and TREC Disks 4&5.

From the different categories of ranking models, the learning-to-rank model (L2R) outperforms other title-based statistical ranking models. The L2R model only requires a small set of features, which is automatically determined by a correlation-based feature selection method applied on a large set of established IR retrieval features. The average evaluation results over all datasets showed that the best full-text-based retrieval models outperform the best title-based retrieval models by only 3%. Thus, based on our results, we can state that it is possible, given certain constraints, to build an effective titles-based information retrieval model that provides competitive results compared to a retrieval model operating on full-text.

The remainder of the paper is organized as follows: In Section 2, we review the state of the art in the field. The considered retrieval models for our study from the four categories are presented in Section 3. The evaluation approach is described in Section 4 and the results are reported in Section 5. Section 6 discusses the results, before we conclude.

2 Related Studies

There have been a number of retrieval models that specifically attempted to model the structure of documents, including the division of content into title, body, etc. However, to the best of our knowledge, there are no recent studies that use state-of-the-art retrieval techniques to compare ad-hoc retrieval over titles with ad-hoc retrieval over full-text. In this section, we provide a brief account of prior work related to our comparison. Subsequently, we present in detail in Section 3 the retrieval models that have been selected for our comparison study.

Using less text has shown to be more efficient for documents retrieval tasks [5]. The authors showed that Keywords can be searched more quickly than title material. The addition of keywords to titles increases search time by 12%, while the addition of digests increases it by 20%.

In the domain of biomedical literature, Lin compared full-text retrieval with abstract retrieval [6]. Lin used the MEDLINE test collection and two ranking models: BM25 and a modified TF-IDF. The results show that full-text search outperforms abstracts-only search. Hemminger et al. [7] compared full-text retrieval with retrieval based on the metadata provided by the PubMed database, using gene names as queries. In their work, metadata comprise of titles and abstracts. Hemminger et al. concluded that full-text searches yield better results. However, the authors acknowledge that their study is limited on account of the fact that searching by gene name may not be representative of general biomedical literature searches. Furthermore, the authors used only an exact matching retrieval model to search for a small number of gene names in their study. They suggested extending their work by conducting a similar analysis in other domains. In this paper, we use five datasets from different domains and retrieval models from different IR categories in order to compare the full-text vs. title searches.

3 Compared Retrieval Models

Overall, we ensure that we cover well the four dominant categories of retrieval models [3]. We start by discussing the vector space and probabilistic models.

Subsequently, we present learning to rank models. Finally, we present the deep neural networks models.

Vector space and Probabilistic Models As a baseline, we employ the vector space model TF-IDF [8]. TF represents the frequency of occurrence of a term, while the IDF factor of a term is inversely proportional to the number of documents in which the term appears. This means the fewer the term appears in the corpus, the higher the IDF factor and vice versa.

As a concept-based models, we employ the TF-IDF extensions, CF-IDF [9] and HCF-IDF [2]. CF-IDF is an extension of TF-IDF that counts concepts instead of terms. Concepts are terms from a controlled vocabulary (e.g. the term ”Financial crisis” in the economics thesaurus[4]). HCF-IDF [2] is an extension to CF-IDF that considers the hierarchical structure of concepts. The algorithm uses spreading activation and gives less weight to the more general concepts in the hierarchy.

Another retrieval model which utilizes the IDF weighting for ranking the documents is BM25. BM25 has been used as a baseline in TREC Web track [5,6]. It is a combination of BM11 and BM15 scaled by a scaling factor b. BM25CT is an extension to BM25 which uses a combination of terms and concepts that appears.

Learning to Rank (L2R) Models Learning to Rank (L2R) is a family of machine learning techniques that aim at optimizing a loss function regarding a ranking of items. It has been successfully applied in the past for different IR tasks. Chen et al. [10] proposed a learning to rank approach for finding non-factoid answers in an answer sentence retrieval task. They used a combination of Explicit Semantic Analysis (ESA), Word2Vec [11] as semantic text representations, and Metzler and Kanungo’s features (MK). Chen et al. showed that the combination of the semantic features and the MK feature set provides better ranking results than ranking based on MK feature set.

L2R consists of a set of supervised ranking models that are trained with a set of numerical feature vectors in order to retrieve the top-k relevant documents in response to a user’s query. The feature vectors are calculated using the content of the documents and/or the queries. L2R models are generally categorized in pointwise, pairwise, and listwise approaches depending on the way the model performs the optimization task [12]. Pointwise is the category of L2R models where a relevancy degree is generated for every single document regardless of the other documents in the results list of the query. In contrast, the loss function of pairwise approaches considers only one pair of documents at a time. Finally, in the listwise L2R models, the input consists of the entire list of documents associated with a query and the output consists of a ranked list of documents for each query. As a pairwise approach, we use RankNet [13], LambdaMart [14], and RankBoost [15]. Finally, for listwise L2R we use AdaRank [16], Coordinate Ascent [17], and ListNet [18].

Deep Learning Models The recent resurgence of neural networks has also affected the Information Retrieval community. Zhang et al. [19] provided a detailed survey to illustrate the rough evolution of Neural IR research and word embedding approaches to IR. For web search, Huang et al. [20] propose a series of deep structured semantic models (DSSM). The most successful instance of the model uses a multilayer feed-forward neural network to map both the query and the title of a webpage to a common low-dimensional vector space. The similarity between the query-document pairs is computed using cosine similarity. The main novelty is the usage of word-hashing, which dramatically reduces the vocabulary size without neglecting too much information. The reduction in vocabulary size allows the neural network to learn effectively from a large amount of available labeled data. DSSM is composed of four different layers. The first layer is the input layer. It contains the word sequences of the document and the user query. The second layer transforms the word sequences into sub-word units to reduce the large amount of vocabulary size. Subsequently, the sub-word units are used as input for a feed-forward neural network. In order to determine the relevancy of a document, cosine similarity between the query and the documents is computed on the output layer. The documents are ranked with respect to their similarity scores. As an extension to the DSSM model, Shen et al. [21] enhance on that by replacing the feed-forward neural network with a convolutional neural network. Afterwards, they introduced convolutional neural networks with max-pooling in the DSSM architecture (C-DSSM) [22]. The convolutional layer and max-pooling layer are utilized to identify keywords and concepts, in both the query and the document, and project them into a lower-dimensional semantic layer. C-DSSM is claimed to be state-of-the-art in retrieval performance [21].

4 Evaluation

In order to evaluate the effectiveness of title-based retrieval vs. a full-text retrieval, we use five datasets, which are described in Section 4.1. In Section 4.2, we present our evaluation procedures and parameters. In Section 4.3, we explain how we apply the correlation-based feature selection algorithm to sample a subset of features for L2R models. In Section 4.4, we present the metric used for evaluating the retrieval results.

4.1 Datasets

We use labeled datasets which have a full-text and a title. This enables a direct and fair comparison of retrieval performances over the two forms of content. The datasets fall under two categories: (1) standard IR datasets and (2) digital library datasets.

Standard IR Datasets For the standard IR datasets, a document is given a binary classification as either relevant or non-relevant. This decision is referred to as the gold standard or relevance judgments. We used the following two standard IR datasets, namely NTCIR-2 and TREC 4&5, which provide a set of topics and human relevance judgments. Table 1 presents an overview of the datasets characteristics.

(1) *NTCIR-2:* The dataset consists of 49 search topics and 322,059 documents' abstracts. We use the search topics as queries. The documents were extracted from the NACSIS Academic Conference Paper Database, collected between 1997-1999, and NACSIS Grant-in-Aid scientific research database, collected between

Table 1: Overview of the datasets characteristics. |avg| denotes the average number of documents and queries

	NTCIR-2	TREC 4&5	EconBiz	IREON	PubMed	|avg|
# of documents	322,059	507,011	288,344	27,575	646,655	**358,329**
# of queries	66,729	72,270	6,204	7,912	28,470	**36,317**

1988-1997. The documents are from electronics, chemistry, physical sciences, and clinical reports. We use a combination of the titles and abstracts to make up for the missing full-texts. Furthermore, the dataset includes relevance judgments of 66,729 query-document pairs.

(2) *TREC 4&5:* consists of 507,011 English documents from various newspaper or newswire sources (Financial Times, Foreign Broadcast Information Service, Los Angeles Times) and government proceedings (Congressional Record, Federal Register) collected between 1998 and 1994. For our investigation, all data items needed to have a full-text and a title. When examining the files, around 50 thousand documents were missing one of these elements. These documents are mainly from the Federal Register and Los Angeles Times and thus were ignored for our experiments. TREC provides human annotated relevance judgments for some query-document pairs. We use TREC-6 ad-hoc qrels in our experiments. TREC-6 ad-hoc qrels consists of 50 topics and relevance judgments of 72,270 query/document pairs.

Datasets of Digital Libraries In case of the digital library datasets, a hierarchical domain-specific thesaurus that provides topics (or concepts) of the libraries' domain is usually included. Furthermore, many of the digital library documents are manually annotated, by domain experts, with at least one of these concepts. Thus, in our evaluation of the digital library datasets, we consider the document as relevant to a concept if and only if it is annotated with the corresponding concept.

We use the following three digital library datasets Econbiz, IREON, and PubMed which come with a hierarchical thesaurus. This thesaurus provides topics on economic, political, and medical subjects.

(3) *EconBiz:* ZBW, the world's largest economics library, is running a search portal, called EconBiz, for economics' scientific publications. From EconBiz, we obtain 1 million URLs of open access scientific publications and generate a dataset of 288,344 full-text English publications. As user queries, we use the economics thesaurus (STW). The economics thesaurus provides 6,204 economics subjects, i.e, concepts in economics. The thesaurus is developed and maintained by an editorial board of domain experts at ZBW – Leibniz information centre for economics. In this dataset, 203,851 documents are annotated with at least one thesaurus concept.

(4) *IREON:* The German information network 'International Relations and Area Studies' provides us with a dataset of 27,575 full-text politics publications in English. The dataset also contains a politics thesaurus (FIV) with 7,912 political English subjects. Again, the thesaurus subjects are used as queries in

our experiments. In this dataset, 3,936 documents are annotated with at least one thesaurus concept.

(5) *PubMed:* PubMed consists of around 27 million citations for biomedical literature from MEDLINE, life science journals, and online books. Some of the citations include links to full-text content from PubMed Central and publisher websites. From PubMed central, we obtained 646,655 full-text open-access English articles. PubMed is provided by the US national library of medicine. As queries, we use the medical terms from the Medical Subject Headings (MeSH) thesaurus. MeSH consists of 28,470 subjects. In this dataset, 506,802 documents are annotated with at least one thesaurus concept.

4.2 Experimental Procedure

In order to compare the retrieval performance over titles versus full-text, we implemented the retrieval models described in Section 3. The retrieval models generate a ranked list of documents for each query-document pair. In order to evaluate the performance of the retrieval models, we compare the ranked list with the gold standard (see Section 4.4). The procedures for evaluating the standard IR datasets is slightly different from the one of the digital library datasets. In the case of the standard IR datasets, where the human relevance judgments are provided, we generate the lists of the *top-20* documents using the full-text and titles retrieval models, respectively. Afterwards, the lists are compared to the relevance judgments provided as gold standard. Whereas with the digital library datasets, the items of the ranked list are considered as relevant if and only if the search query is included in the document annotations, i.e., binary decision whether a certain annotation that we have queried for is provided with the document's gold standard, or not.

In order to generate the evaluation results for **vector space models and probabilistic retrieval models**, tokenization, stop words removal, and porter stemming are applied. As described in Section 3, the concept-based approaches HCF-IDF and CF-IDF utilize the concepts from STW, FIV, and MeSH, and BM25CT utilize a vector union of the terms and concept features.

For the **L2R models**, following the suggestions of Qin et al. [25] and Minka et al. [26], the documents are sampled in the following way. First, we use BM25 on titles and full-texts to rank all the documents with respect to each query, and then the top 1,000 documents for each query are selected for feature extraction. Subsequently, we extract 29 features for each of the query-document pairs (see details on feature selection in Section 4.3). Therefore, we obtain two feature files, one for the titles and one for the full-texts, for each dataset. We utilize these files together with the gold standard to train the six L2R models selected in Section 3. The LambdaMART model is trained using 1,000 trees with 10 leaves per tree. The learning rate is set to 0.1 and 256 threshold candidates for tree splitting was used. The minimum number of samples for a leaf was set to 1. Early stopping is applied, if there was no improvement for 100 consecutive rounds. The RankNet model is trained using 100 training epochs, one hidden layer, and 10 hidden nodes. The learning rate was set to 0.00005. For the RankBoost model, we use

Table 2: Overview of the L2R features

MK set [23]	Sentence length, Exact match, Term overlap, Synonym overlap, Language Model with dirichlet smoothing
Modified letor [24]	Covered query term number, IDF, Sum/Min/Max/Mean/Variance of TF, Sum/Min/Max/Mean/Variance of length normalized TF, Sum/Min/Max/Mean/Variance of TF-IDF, Language model absolute discounting smoothing, Language model with Bayesian smoothing using dirichlet priors, Language model with jelinek-mercer smoothing
Ranking model features	TF-IDF, BM25, CF-IDF, HCF-IDF, Word2Vec [10]

300 training epochs and 10 threshold candidates. AdaRank is trained in 500 rounds with a learning tolerance of 0.002. The number of epochs for the ListNet model is 1,500 and the learning rate is set to 0.00001. In the case of Coordinate Ascent, we apply 5 random restarts and 25 search iterations per dimension. The performance tolerance is set to 0.001. No regularization was used. In order to ensure that we obtain optimal L2R models that do not over or under-fit, we applied the bias-variance tradeoff method [27].

Finally, we generate the evaluation results for the **semantic models** DSSM and CDSSM. A trained model, with a click-through dataset of 30 million query/clicked-title pairs from Microsoft is used to determine the semantic cosine similarity between each query-document pair in our datasets. Based on the similarity scores, the top twenty documents are passed to the evaluation metric (described in Section 4.4).

4.3 Feature selection for L2R models

A good IR system can retrieve the most important documents in a fast and scalable way using only a limited amount of information about the query and documents. The information is contained in the features of both document and query and therefore a good set of features has to be found. The aim of the feature selection is to find a meaningful subset of features which can still produce sound results. Given a large number of different IR features, we want to find those features which cover diverse information and still contribute the most to the retrieval of the most important documents.

In line with previous literatures [23,24], we use a set of 29 features (see Table 2) to train our models. The features are the Metzler and Kanungo (MK) [23] set, modified LETOR [24], semantic, and statistical features. The original MK feature set used six features for the query-based summarization task. Due to the difference to our task (comparing query and title), we ignore the sentence location feature because the titles usually consist only of one sentence. Regarding the features Term Overlap and Synonym Overlap, we remove stop-words and perform porter stemming on the queries and titles. The Term Overlap is the fraction of query terms that occur in the document (titles or full-text), while the Synonyms Overlap is the fraction of query terms that either occur or have a synonym in the document. We utilize NLTK[4] to generate synonyms. For the LETOR feature set, we ignore all web-related features (e.g. Sitemap term propagation). The language model parameters are taken from the original work. Additionally, we use the vector representation of words (Word2Vec) to compute the similarity between a query-document pair and use it as an L2R feature. For this purpose, we use Google News, the pre-trained distributional model [28], and gensim framework to generate the similarity scores [29]. Regarding the language model features, we use an Elasticsearch[4] full-text index to generate them. Moreover, we use the scores of the ranking models (BM25, CF-IDF, and HCF-IDF) as L2R features.

We further investigate the possibility of sampling a meaningful subset of features that decreases the error rate of the ranking models. For this purpose, we apply a correlation-based feature selection algorithm (CFS) [30] on each dataset and content modality, i.e., separately for full-text and title. The CFS algorithm computes a score for a subset S of the 29 features containing k features using the following equation (denoted as $score_{CFS}(S)$ in [30]): $\mathrm{score_{CFS(S)}} = \frac{k \cdot \overline{r_{gf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}}$ where $\overline{r_{gf}}$ is the average gold standard(g)-feature correlation and $\overline{r_{ff}}$ represents the average inter-correlation between the features. The formula denotes higher scores to the subsets which have a low 'feature-feature' correlations and high 'gold standard-feature' correlations.

We calculated $score_{CFS}(S)$ for all feature subsets of sizes $|S| = \{1, \ldots, 29\}$, which equals $2^{29} - 1 = 536,870,911$ possible subsets, for each dataset and configuration (full-texts or titles). The best feature sets, in terms of their $score_{CFS}(S)$, are reported in Table 3. We utilize these features in our learning-to-rank models. The results are presented in Table 4.

The CFS results showed that some features, such as BM25, contribute the most to the results. This is consistent with that of Qin et al. [24], who found that using BM25 as a feature in L2R models improves the overall performance of the L2R models.

4.4 Evaluation Metric

We evaluate the retrieval results using normalized discounted cumulative gain ($nDCG$). We assume that users do not look beyond two pages of 10 results. Thus, we limited our evaluation to the top 20 results. The metric $nDCG$ compares the top-20 documents (DCG), with the gold standard and is computed as follows: $\mathrm{nDCG@k} = \frac{\mathrm{DCG@k}}{\mathrm{IDCG@k}}$, where $\mathrm{DCG@k} = \mathrm{rel(1)} + \sum_{i=2}^{k} \frac{\mathrm{rel(i)}}{\mathrm{log(i)}}$ D is a set of documents, $rel(d)$ is a function that returns one if the document is rated relevant, otherwise zero, and $IDCG_k$ is the optimal ranking.

5 Results

In this section, we present the results of the titles vs. full-text retrieval comparison. We observe that the retrieval over titles yields a close nDCG@20 values, or even

Table 3: Best feature subsets (BFS) based on the CFS approach. # is the number of features in the corresponding BFS

Dataset	Content	Best Feature Subset (BFS)	#	$\text{score}_{CFS(BFS)}$
NTCIR-2	**Full-text**	BM25, Exact match	2	0.20
	Titles	BM25, Exact match	2	0.15
TREC	**Full-text**	BM25, Exact match, Sum of length normalized TF	3	0.28
	Titles	BM25, Language model with Dirichlet smoothing, Minimum of TF-IDF, Term overlap, Word2vec	5	0.13
Econbiz	**Full-text**	Language model with absolute discounting smoothing, Language model with bayesian smoothing using dirichlet priors, Min TF-IDF, Var TF-IDF	4	0.41
	Titles	BM25, Exact match, Language model, Synonym overlap, Term overlap, Covered query term number, Max TF-IDF, Mean length norm TF, Mean TF, Mean TF-IDF, Min length norm TF, Min TF, Min TF-IDF, Sum length norm TF, Sum TF, Sum TFIDF	16	0.71
Politics	**Full-text**	Language model with dirichlet smoothing, Language model with absolute discounting smoothing, Language model with Jelinek-Mercer smoothing, Max TF-IDF, Mean TF-IDF, Min TF-IDF, Sum TF, Sum TF-IDF, Var TF-IDF	9	0.41
	Titles	BM25	1	0.54
PubMed	**Full-text**	Language model with Jelinek-Mercer smoothing, Mean TF-IDF	2	0.46
	Titles	Language model with absolute discounting smoothing, IDF	2	0.44

higher metric values in case of the NTCIR-2. Table 4 presents the performance of the title- and full-text-based ranking models on all datasets.

For the NTCIR-2 dataset, we observe that the learning to rank model, Coordinate Ascent, with the full set of 29 features attained the *nDCG*@20 value of 0.33, which is 0.01 higher than C-DSSM and BM25 on full-text retrieval. The same metric value of 0.33 has also been attained from the titles retrieval using the DSSM model. Having reduced the features to the best feature set, BM25 and exact matching, coordinate ascent performance slightly improved on full-text retrieval (0.37). However, the titles retrieval of the same model remained at the same value (0.29).

In the case of the TREC dataset, we observe that BM25 achieved the best results on full-text and titles retrieval. BM25 attains a nDCG of 0.41 on full-text compared to 0.23 on titles. The L2R methods, LambdaMart, and Coordinate Ascent using the full feature set and the best feature set attained close results to BM25 on both full-text and titles.

Considering the evaluation results for the EconBiz dataset, we used the STW concepts as queries. We observe that the retrieval over full-text yields higher

Table 4: Average nDCG@20 scores using full-text (FT) vs. titles (T) over the five datasets. The L2R results are calculated using the full feature set (FFS) and the best feature subset (BFS).

		NTCIR-2		**TREC**		**EconBiz**		**IREON**		**PubMed**	
Family	**IR Method**	**T**	**FT**	**T**	**FT**	**T**	**FT**	**T**	**FT**	**T**	**FT**
VSM	**TF-IDF**	0.19	0.18	0.21	0.39	0.26	0.22	0.661	0.36	0.80	0.54
	CF-IDF	0.05	0.05	0.12	0.13	0.13	0.19	0.44	0.32	0.66	0.49
	HCF-IDF	0.23	0.24	0.10	0.12	0.25	0.20	0.659	0.37	0.80	0.54
PM	**BM25**	0.24	0.32	**0.23**	**0.41**	0.25	0.20	0.662	0.37	0.80	0.55
	BM25CT	0.24	0.31	0.20	0.405	0.27	0.19	0.660	0.37	**0.81**	0.56
L2R-FFS	**LambdaMART**	0.25	0.30	0.22	0.39	**0.67**	0.68	0.83	0.69	0.67	0.67
	RankNet	0.28	0.29	0.13	0.10	0.28	0.10	0.20	0.21	0.30	0.30
	RankBoost	0.26	0.32	0.21	0.34	0.52	0.69	0.80	0.59	0.70	0.79
	AdaRank	0.21	0.31	0.19	0.22	0.50	0.67	0.79	0.65	0.56	0.52
	ListNet	0.21	0.24	0.15	0.07	0.28	0.10	0.20	0.20	0.30	0.30
	Coord. Ascent	0.29	0.33	0.22	0.39	0.57	**0.80**	**0.95**	**0.77**	**0.81**	**0.80**
SM	**DSSM**	**0.33**	0.26	0.18	0.23	0.29	0.33	0.41	0.39	0.34	0.33
	C-DSSM	0.32	0.32	0.18	0.20	0.29	0.34	0.42	0.44	0.32	0.35
L2R-BFS	**LambdaMART**	0.20	0.15	0.16	0.33	0.56	0.63	0.70	0.65	0.42	0.65
	RankNet	0.28	0.25	0.05	0.046	0.28	0.10	0.26	0.41	0.59	0.63
	RankBoost	0.26	0.37	0.13	0.38	0.52	0.10	0.80	0.47	0.30	0.72
	AdaRank	0.29	**0.37**	0.18	0.37	0.48	0.49	0.94	0.61	0.59	0.79
	ListNet	0.29	**0.37**	0.19	0.37	0.28	0.28	0.94	0.41	0.39	0.49
	Coord. Ascent	0.29	**0.37**	0.18	0.38	0.53	0.10	0.94	0.69	0.59	0.78

nDCG metric values. The best title-based retrieval model, LambdaMART, attains a nDCG of 0.67 compared to 0.80 of Coordinate Ascent on full-text.

For IREON and PubMed, we used the FIV and MeSH concepts as queries. The titles retrieval was competitive with the full-text retrieval. In both datasets, Coordinate Ascent achieved the best retrieval results. On titles, Coordinate Ascent attained nDCG values of 0.95 and 0.81. These values are 18% and 1% higher than the best full-text retrieval models, respectively.

6 Discussion

The results show that a title-based retrieval over large document corpora is possible. For four of our five datasets, we obtained nDCG@20 scores that are similar or even better than the scores obtained from the best retrieval models over full-text. Figure 1 visualizes the nDCG scores of the best performing retrieval methods for each individual dataset. Below, we discuss the key insights for each dataset.

For NTCIR-2, the best title-based retrieval model DSSM attained the same nDCG metric value as the second best full-text-based retrieval model Coordinate

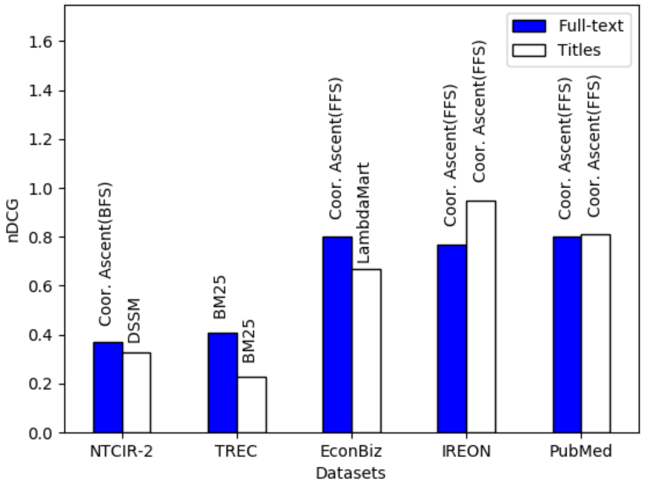

Fig. 1: Average nDCG@20 of the best performing retrieval models

Ascent. In order to generate the L2R results, we used 29 features including Word2Vec and MK feature sets. Applying the CFS method resulted in generating BM25 and the number of exact matches as the best feature subsets. By using solely those two features, the best L2R method on full-text, Coordinate Ascent, gained 4% in terms of nDCG. NTCIR-2 was the only dataset, in which DSSM achieves the best title-based retrieval results. Our study conforms with Cohen et al. [31], who showed that DSSM performs poorly on a traditional dataset for ad-hoc retrieval and argue that the word-hashing method discards too much information.

For the TREC and EconBiz datasets, the titles retrieval results were not competitive with those of full-text retrieval. The nDCG values of the best full-text-based models on TREC and EconBiz were 18%, and 13% better than the best titles-based models datasets respectively.

For the IREON and PubMed datasets, the retrieval over titles results in consistently higher metric values in terms of nDCG.

Aggregating the best nDCG values over all datasets and configurations, the best titles-based retrieval models attain a value of 0.60, whereas the best full-text retrieval models attain a mean score of 0.63 (3% relative improvement). Therefore, we believe that title-based retrieval can be considered providing competitive results comparable with the full-text-based retrieval.

One may consider it as a limitation of our study that thesauri concepts STW, FIV, and MeSH are used for retrieving the results from the digital library datasets EconBiz, IREON, and PubMed, respectively. Considering that these thesauri concepts belong to the same domain as their corresponding datasets, one can consider that finding matching documents is easier. However, the concepts are purposefully chosen as queries as they often resemble topics that users of the scientific digital libraries actually search for.

Reproducibility All the code for reproducing the experiments is publicly available as bitbucket repository[4].

7 Conclusions

We conducted a study to compare title-based with full-text-based ad-hoc retrieval. For this purpose, different retrieval models of different families (probabilistic models, vector space, learning to rank models and semantic models) were compared. We used five datasets, out of which three datasets are obtained from digital libraries: Econbiz, PubMed and IREON, and two standard test collections. Overall, our experiments show that title-based ad-hoc retrieval models can provide close, and sometimes even better, results compared to the full-text ad-hoc retrieval.

Acknowledgement This work was supported by the EU's Horizon 2020 programme under grant agreement H2020-693092 MOVING.

References

1. Galke, L., Mai, F., Schelten, A., Brunsch, D., Scherp, A.: Using Titles vs. Full-text as Source for Automated Semantic Document Annotation. In: International Conference on Knowledge Capture (K-CAP). (May 2017)
2. Nishioka, C., Scherp, A.: Profiling vs. time vs. content: What does matter for top-k publication recommendation based on twitter profiles? In: Digital Libraries (JCDL), 2016 IEEE/ACM Joint Conference on, IEEE (2016) 171–180
3. Croft, W.B., Metzler, D., Strohman, T.: Search engines: Information retrieval in practice. Volume 283. Addison-Wesley Reading (2010)
4. Christopher, D.M., Prabhakar, R., Hinrich, S.: Introduction to information retrieval. An Introduction To Information Retrieval **151** (2008) 177
5. BARKER, F.H., VEAL, D.C., WYATT, B.K.: Comparative efficiency of searching titles, abstracts, and index terms in a free-text data base. Journal of Documentation **28**(1) (1972) 22–36
6. Lin, J.: Is searching full text more effective than searching abstracts? BMC Bioinformatics **10**(1) (2009) 46
7. Hemminger, B.M., Saelim, B., Sullivan, P.F., Vision, T.J.: Comparison of full-text searching to metadata searching for genes in two biomedical literature cohorts. Journal of the American Society for Information Science and Technology **58**(14) (2007) 2341–2352
8. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18**(11) (1975) 613–620
9. Goossen, F., IJntema, W., Frasincar, F., Hogenboom, F., Kaymak, U.: News personalization using the cf-idf semantic recommender. In: The International Conference on Web Intelligence, Mining and Semantics, ACM (2011)
10. Chen, R.C., Spina, D., Croft, W.B., Sanderson, M., Scholer, F.: Harnessing semantics for answer sentence retrieval. In: Workshop on Exploiting Semantic Annotations in Information Retrieval, ACM (2015) 21–27

[4] https://bitbucket.org/a_saleh/icadl2018

11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 26. Curran Associates, Inc. (2013) 3111–3119
12. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval **3**(3) (2009) 225–331
13. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning, ACM (2005) 89–96
14. Wu, Q., Burges, C.J., Svore, K.M., Gao, J.: Adapting boosting for information retrieval measures. Information Retrieval **13**(3) (2010) 254–270
15. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of machine learning research (2003)
16. Xu, J., Li, H.: Adarank: a boosting algorithm for information retrieval. In: The annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2007) 391–398
17. Metzler, D., Croft, W.B.: Linear feature-based models for information retrieval. Information Retrieval **10**(3) (2007) 257–274
18. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: The 24th international conference on Machine learning, ACM (2007) 129–136
19. Zhang, Y., Rahman, M.M., Braylan, A., Dang, B., Chang, H.L., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., et al.: Neural information retrieval: A literature review. arXiv preprint arXiv:1611.06792 (2016)
20. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. International conference on information and knowledge management (2013)
21. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: The International Conference on World Wide Web, ACM (2014) 373–374
22. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: The International Conference on Information and Knowledge Management, ACM (2014)
23. Metzler, D., Kanungo, T.: Machine learned sentence selection strategies for query-biased summarization. In: SIGIR Learning to Rank Workshop. (2008)
24. Qin, T., Liu, T.Y.: Introducing LETOR 4.0 Datasets. CoRR (2013)
25. Qin, T., Liu, T.Y., Xu, J., Li, H.: How to make LETOR more useful and reliable. In: SIGIR Workshop on Learning to Rank for Information Retrieval. (2008)
26. Minka, T., Robertson, S.: Selection bias in the letor datasets. In: SIGIR Workshop on Learning to Rank for Information Retrieval, Citeseer (2008) 48–51
27. Fortmann-Roe, S.: Understanding the bias-variance tradeoff. (2012)
28. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
29. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: The LREC Workshop on New Challenges for NLP Frameworks. (2010)
30. Hall, M.A.: Correlation-based feature selection of discrete and numeric class machine learning. (2000)
31. Cohen, D., Ai, Q., Croft, W.B.: Adaptability of neural networks on varying granularity IR tasks. arXiv preprint arXiv:1606.07565 (2016)