Contents lists available at ScienceDirect

Information Processing and Management

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

Diversity-aware strategies for static index pruning

Sevgi Yigit-Sert [a,1], Ismail Sengor Altingovde [b,*], Özgür Ulusoy [c]

[a] *Department of Computer Engineering, Ankara University, Ankara, Turkey*
[b] *Computer Engineering Department, Middle East Technical University, Ankara, Turkey*
[c] *Computer Engineering Department, Bilkent University, Ankara, Turkey*

ARTICLE INFO

Keywords:
Query result diversity
Static index pruning
Query processing efficiency

ABSTRACT

Static index pruning aims to remove redundant parts of an index to reduce the file size and query processing time. In this paper, we focus on the impact of index pruning on the topical diversity of query results obtained over these pruned indexes, due to the emergence of diversity as an important metric of quality in modern search systems. We hypothesize that typical index pruning strategies are likely to harm result diversity, as the latter dimension has been vastly overlooked while designing and evaluating such methods. As a remedy, we introduce three novel diversity-aware pruning strategies aimed at maintaining the diversity effectiveness of query results. In addition to other widely used features, our strategies exploit document clustering methods and word-embeddings to assess the possible impact of index elements on the topical diversity, and to guide the pruning process accordingly. Our thorough experimental evaluations verify that typical index pruning strategies lead to a substantial decline (i.e., up to 50% for some metrics) in the diversity of the results obtained over the pruned indexes. Our diversity-aware approaches remedy such losses to a great extent, and yield more diverse query results, for which scores of the various diversity metrics are closer to those obtained over the full index. Specifically, our best-performing strategy provides gains in result diversity reaching up to 2.9%, 3.0%, 7.5%, and 3.9% wrt. the strongest baseline, in terms of the ERR-IA, α-nDCG, P-IA, and ST-Recall metrics (at the cut-off value of 20), respectively. The proposed strategies also yield better scores in terms of an entropy-based fairness metric, confirming the correlation between topical diversity and fairness in this setup.

1. Introduction

In modern information retrieval (IR) systems, ranging from domain-specific search applications to commercial Web search engines, an inverted index takes a central role in computing a ranked list of documents as an answer to a keyword-based query (e.g., see Archer et al. (2019), Jeon et al. (2014), Yin et al. (2016) demonstrating the use of inverted index in large scale search engines such as Bing, Yahoo, and Google, respectively). In essence, an inverted index associates each term with a postings list, which comprises of a *document id* the term has appeared in and a *payload* that contains information about term occurrence, such as term frequency (Lin & Dyer, 2010; Manning et al., 2008).

For the scenarios that require a rather basic text-search functionality, the inverted index is the main component that allows computing the score of the documents in the union (or intersection) of the query terms, using well-known matching functions, such as BM25, as implemented in various open-source IR systems (e.g., Lucene ecosystem). Commercial Web search engines usually

[*] Corresponding author.
E-mail addresses: syigit@ankara.edu.tr (S. Yigit-Sert), altingovde@ceng.metu.edu.tr (I.S. Altingovde), oulusoy@cs.bilkent.edu.tr (Ö. Ulusoy).
[1] This research was conducted while Sevgi Yigit-Sert was at Computer Engineering Department of Middle East Technical University.

https://doi.org/10.1016/j.ipm.2024.103795
Received 7 November 2023; Received in revised form 19 May 2024; Accepted 20 May 2024
Available online 30 May 2024

employ the inverted index for the first-stage retrieval, i.e., to obtain a candidate set of matching documents, which are subsequently re-ranked using machine learnt models (e.g., (Mackenzie et al., 2018; Wang et al., 2016; Yin et al., 2016)). Even with the recent wave of exploiting neural models (such as BERT (Devlin et al., 2019)) for ranking, the inverted index proves to be useful. In such scenarios, an inverted index can be employed to obtain a candidate set of documents to be re-ranked, as in the latter case of first-stage rankers; or to store the neural-based weights, i.e., learned sparse weights, to combine the effectiveness of neural models with the time-tested efficiency of query processing over an inverted index (see Macdonald et al. (2021) for an overview).

The size of an inverted index (or its postings lists) is a crucial parameter for the search efficiency, as it affects both query processing time and storage space (in memory and/or on disk[2]). The longer postings lists would take a longer time to read, decompress, and score,[3] as well as a larger space to store. For commercial search engines, the cost of storage might be considerably high yet affordable, while the time cost of fetching and processing lists remains to be critical (Archer et al., 2019), as search engines aim to optimize not only the mean response time but also the tail latency (Jeon et al., 2014). For other applications, such as e-commerce platforms, digital libraries, or Web archives, for which the search functionality is necessary but may not be the primary goal – in contrast to the case of Web search engines –, even the storage space taken by an index might be an important cost, in addition to the processing time.

A particular approach to cope with the index size is so-called *static index pruning*, which aims to reduce the index size while preserving the retrieval effectiveness for the top-ranked results (Carmel et al., 2001). To this end, most of the earlier works focus on strategies that prune terms, documents, or particular postings from the index; then, evaluate the effectiveness by either computing the overlap between the query results obtained from the full and pruned index files, or by computing the typical relevance-oriented metrics (see Sec. 2.1 for a detailed review).

An orthogonal dimension that is becoming increasingly popular to assess the retrieval effectiveness beyond relevance is the *topical diversity* of the search results. Therefore, various search result diversification methods have been proposed to cover as many of the query aspects as possible in the top-ranked query results, especially for the ambiguous, broad, or under-specified queries, to satisfy users with different query intents (Rodrygo et al., 2015; Yigit-Sert et al., 2020). Furthermore, diversification of search results helps mitigate search bias (Maxwell et al., 2019) and enhance fairness in various search (Karako & Manggala, 2018; McDonald et al., 2019) and recommendation scenarios (Schelenz, 2021). That is, as stated in Gao and Shah (2020), there is a correlation between diversity and fairness, and diversity would improve topical fairness by introducing different aspects of a topic in the top-ranked search results.

To the best of our knowledge, static index pruning and topical diversity have not been explored jointly, and there is no static index pruning method taking diversity into account in the literature. However, index pruning may have serious implications on the result diversity, as illustrated in the following toy scenario. Consider the well-known example of an ambiguous query, 'Java', with the aspects *'Java programming language'* and *'Java island'*, denoted as a_1 and a_2, respectively. In Fig. 1(a), we present the postings list of term 'Java', where documents relevant to aspects a_1 and a_2 are shown in red and black. For the sake of illustration, the postings are sorted in the order of some relevance score (such as BM25). Then, top-5 results of the query would include $\{d_5, d_7, d_1, d_{10}, d_3\}$. Assume that we apply a pruning algorithm, which removes some the postings from the list, and only those shown in Fig. 1(b) remain. Obviously, the top-5 result computed using the pruned list, including $\{d_5, d_7, d_1, d_3, d_2\}$, preserves 80% of documents of the original result, and hence, would yield a good performance in terms of the result similarity metrics employed in the literature (Altingovde et al., 2012; Soner et al., 2020). However, as the new result list covers only one of the aspects, namely a_1, the aspect coverage drops to 50%, and the result diversity is seriously harmed.

1.1. Research questions, novel contributions and key findings

Considering the discussions in the preceding section, there emerges an evident need to take topical diversity into account while tailoring and evaluating methods for static index pruning. In pursuit of this goal, we explore the following research questions in this paper.

- RQ1: How does static index pruning affect the diversity of query results?
- RQ2: How can pruning strategies take into account the topical diversity and preserve index elements, i.e., terms, documents or postings, which are important for maintaining result diversity?
- RQ3: Can we achieve both fair and diverse rankings after pruning?

While seeking to answer these questions, our work makes the following novel contributions:

- For the first time in the literature, we address static index pruning by taking the notion of topical diversity into account. As far as we know, there is only one related study (Pehlivan et al., 2013), which aimed to preserve the temporal diversity of query results (i.e., in terms of the timestamps of the retrieved documents), while we focus on the more challenging topical diversity problem. Furthermore, their evaluation setup did not employ diversity metrics as we do in this work. Therefore, we are first to provide a detailed analysis of the impact of static index pruning on the result diversity in a setup using well-known pruning algorithms and diversity evaluation metrics.

[2] While some most popular terms' postings lists can be cached in main memory, the rest of the index is usually stored on disk, even in the commercial Web search engines (Archer et al., 2019).

[3] Though the relationship may not necessarily be linear due to optimizations like skipping and dynamic pruning.

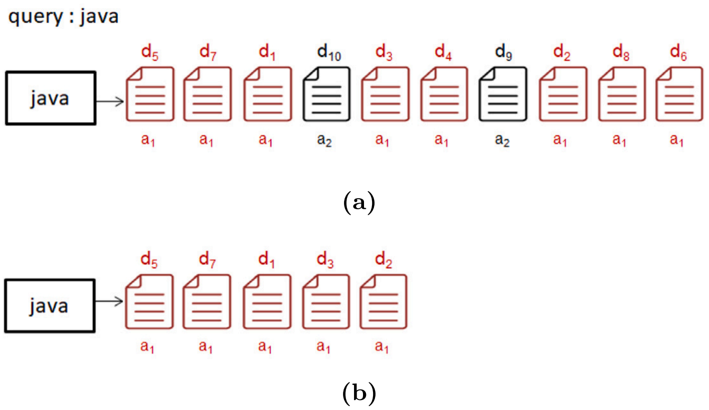


Fig. 1. The impact of pruning on diversity of postings lists for a toy scenario: (a) original, and (b) pruned lists for the term ‘Java’.

- As our second contribution, we propose three diversity-aware pruning strategies. These new strategies make use of the access-counts of the documents for the previous queries, as also employed in some earlier pruning approaches; but extend them by exploiting document clustering methods and word-embeddings to assess the possible impact of a document or posting for the topical diversity.
- We present an experimental setup that allows evaluating the proposed methods for different scenarios. Specifically, we evaluate the diversity of query results obtained over the indexes pruned by our strategies for various rank cut-off values, namely, @20, @50 and @100, using the well-known diversity metrics. The small cut-off value, 20, represents the scenario where we assume that the query results will be directly exposed to the end user. The large cut-off values of 50 and 100 are intended for the scenario where the query results serve as a candidate document set, to be further diversified using one of the algorithms specialized for this purpose (e.g., see Rodrygo et al. (2015)).
- Finally, we evaluate the fairness of query results over the indexes obtained by our diversity-aware pruning strategies, to investigate the interplay of topical diversity and fairness notions in our setup. To this end, we employ a recently proposed entropy-based fairness metric.

Our experiments using typical TREC collections and query sets reveal that baseline static index pruning strategies cause significant reductions (i.e., up to 50% for some metrics) in the diversity of query results for pruning levels ranging from 60% to 90%. Our diversity-aware approaches remedy such losses to a great extent, and yields query results, for which scores of the various diversity metrics are closer to those obtained over the full (unpruned) index. Specifically, our best-performing strategy provides gains in result diversity reaching up to 2.9%, 3.0%, 7.5%, and 3.9% wrt. the strongest baseline, in terms of the ERR-IA, α-nDCG, P-IA, and ST-Recall metrics (at the cut-off value of 20), respectively. Furthermore, we demonstrate that our strategies also yield better scores in terms of a recently proposed fairness metric, confirming the correlation of topical diversity and fairness in our setup.

We believe that the introduced research problem and our contributions addressing this problem are timely and useful for the Information Retrieval community. Our proposed strategies are directly applicable to current retrieval systems ranging from verticals to Web search engines, i.e., for all the scenarios where diversity of results is desirable. Furthermore, our approaches can be either adopted for the learned sparse retrieval scenarios, or can be applied over the indexes that are built for serving first-stage retrieval results to sophisticated re-rankers, such as LLMs, in a dense retrieval scenario. In the latter case, our contribution would be even more impactful, as we allow keeping the efficiency promises of retrieval over an inverted index without sacrificing the result diversity, which would be highly required for subsequent ranking stages.

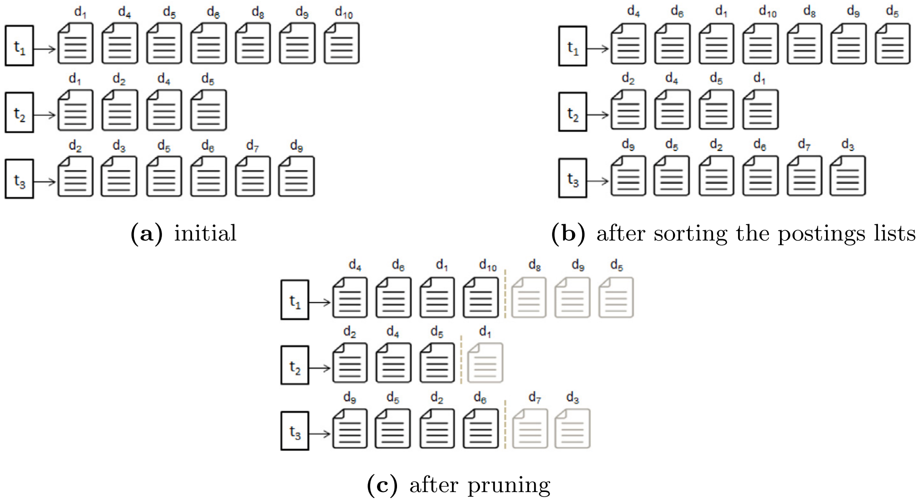
The remainder of this paper is structured as follows: In the next section, we outline existing strategies in the literature relating to static index pruning and also provide a concise overview of diversification in search results. Section 3 introduces our three diversity-aware pruning strategies. We discuss our experimental setup in Section 4. In the following section, we present a comprehensive experimental evaluation, comparing our work with the most competitive strategies in the literature, and discuss broader implications of our findings. Finally, we summarize our main findings and give concluding remarks in Section 6.

2. Background and related work

In this section, we first review static index pruning strategies in the literature and then discuss studies that focus on the diversity and fairness of search results.

2.1. Static index pruning

An inverted index correlates index terms and documents with their occurrence. The index terms are referred to as dictionary (a.k.a., lexicon or vocabulary). Each term is associated with a list, called a postings list, which comprises of a *document id* the term


t_1 d_1 d_4 d_5 d_6 d_8 d_9 d_{10}
t_2 d_1 d_2 d_4 d_5
t_3 d_2 d_3 d_5 d_6 d_7 d_9
t_1 d_4 d_6 d_1 d_{10} d_8 d_9 d_5
t_2 d_2 d_4 d_5 d_1
t_3 d_9 d_5 d_2 d_6 d_7 d_3
t_1 d_4 d_6 d_1 d_{10} d_8 d_9 d_5
t_2 d_2 d_4 d_5 d_1
t_3 d_9 d_5 d_2 d_6 d_7 d_3

(a) initial

(b) after sorting the postings lists

(c) after pruning

Fig. 2. The illustration of TCP algorithm.

has appeared in and a *payload* that contains information about term occurrences such as term frequency, the position of every occurrence of the term in the document, etc. (Lin & Dyer, 2010; Manning et al., 2008).

Static index pruning permanently discards some redundant parts of the index to increase disk space utilization and speed up query processing. Static index pruning strategies vary depending on whether pruning goes over terms or documents and are categorized as term-centric and document-centric, respectively.

2.1.1. Term- and document-centric pruning

A pioneering study in static index pruning was conducted by Carmel et al. (2001). In their first approach, called *uniform*, they proposed a fixed threshold for all terms in the vocabulary. All entries in postings lists are sorted by SMART scoring function, and those below the threshold are thrown out of the index. Their next approach, which is more successful, individually defines a threshold value for each term. The documents in the postings list of a term are sorted in descending order, kth highest score of term t is determined as the threshold value and the postings under the threshold are accepted as less important documents and removed from the index. In this study, we refer to the latter algorithm as *term-centric pruning (TCP)*. It is illustrated in Fig. 2.

Chen and Lee (2013) adapted pruning as a convex integer program providing a theoretical foundation. They re-studied uniform pruning with Kullback–Leibler divergence (KLD) and obtained impressive mean average precision (MAP) results, but some technical issues were left unsettled. Their follow-up study (Chen et al., 2015) explored different divergence measures to overcome these issues.

Moura et al. (2005) introduced a new pruning method that takes into account the positional index in addition to the typical (frequency) index to handle conjunctive and phrase queries. In Moura et al. (2008), they improved their method to enable the pruning process to be performed simultaneously at index construction or updating.

Blanco and Barreiro (2007b) proposed to eliminate stop words with their postings lists from the index to reduce the index size. They employed three techniques to identify stopwords: inverse document frequency (IDF), residual inverse document frequency (RIDF), and term discrimination model. Their next study (Blanco & Barreiro, 2010) introduced a novel pruning strategy based on the probability ranking principle (PRP) (Rijsbergen, 1979; Robertson, 1977), which is widely used for document retrieval in IR. They considered every term in the dictionary as a single-term query and decided whether the document would remain in the index depending on three distinct probabilities (i.e., document prior, query likelihood, probability of a term being non-relevant).

In contrast to term-centric strategies that discard some postings of terms, Büttcher and Clarke (2006) introduced *document-centric pruning (DCP)* strategy that removes some terms from documents to reduce the size of documents as it would also reduce the size of the index. This approach is demonstrated in Fig. 3. Inspired by Carpineto et al. (2001), they assessed each term's contribution to the document's content using KLD, and then stored top-scoring terms in the document and disposed of others. They presented two instantiations of the method: (1) they stored top-k (fixed k) terms of each document at the index; (2) the number of terms to be pruned in a document is decided according to the count of distinct terms in the document. The intuition behind this approach is that larger documents are likely to have a wide variety of topics implying more possible query terms. The latter yields better performance. In Altingovde et al. (2012), the authors applied DCP to all terms in the index instead of only the 10^6 most frequent terms as in Büttcher and Clarke (2006) and counted all terms rather than unique terms while computing document size.

Zheng and Cox (2009) represented another document-centric pruning strategy that determines documents to be pruned by assessing the entropy of terms in the document. They argued that as the number of distinctive terms (i.e., scores with low entropy) in a document increases, this document could be considered as more important than the others in the collection. They entirely removed all postings related to the documents that are less important. However, due to the fact that the improvement in performance over different collections is not stable, they suggested a hybrid approach in which 10% of the index is pruned by the entropy-based

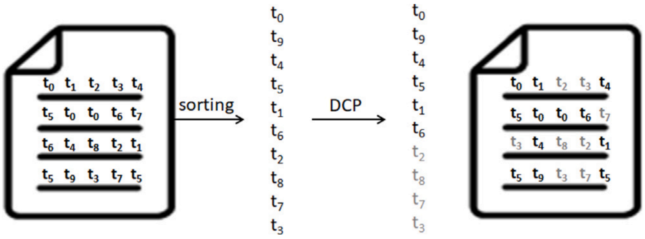


Fig. 3. The illustration of DCP algorithm.

method and the rest by Carmel's method (Carmel et al., 2001). Vishwakarma et al. (2014) also performed a pruning in which, similar to Zheng and Cox (2009), all postings of the document are removed from the index employing a different scoring function based on tf-idf.

Nguyen (2009) discussed pruning in terms of postings by combining the term- and document-centric approaches. Each posting re-defined as <term, document, term frequency> passes through a ranking function that considers both informativeness of the term and the importance of the document among the collection, and some of them are eliminated from the index according to their scores until desired pruning level is reached. The experiments demonstrated that the posting-based approach is better at low pruning levels, but does not outperform the performance of DCP at higher levels.

2.1.2. Popularity and access-based pruning

Here, we provide a detailed explanation of access-based pruning strategies since we construct our diversity-aware pruning algorithms based on them. We also describe the Popularity-based Pruning (PP) method, which is a strong and practical competitor in the literature.

Garcia (2007) attempted to employ the query log of a search engine for static index pruning. Instead of using the relevance score of a document as the scoring function, he utilized the access count of the document, which is defined as the number of times the document occurs in the top-k search results of any query. k can be 10, 100, and 1000. This work achieves 75% improvement in query response time, yet reduces the accuracy (up to %22 in MAP) since a constant number of postings having the top access counts for each term are kept.

Altingovde et al. (2012) enhanced the latter approach, named *access-based term centric pruning (aTCP)*, by keeping a fraction (ϵ) of postings for every term in a dictionary (where ϵ indicates the pruning level). In Altingovde et al. (2012), a novel strategy called *access-based document-centric pruning (aDCP)* was proposed, as well. Access count information is employed from the document-centric pruning perspective in contrast to term-centric, where postings from each term list are pruned. They discarded documents with the lowest access count values from the index until a condition bound to collection length is reached. The collection length, $|C|$, is obtained by adding the number of distinct terms in every document of the collection.

The usage of the popularity of terms for pruning and caching (Baeza-Yates et al., 2007; Ntoulas & Cho, 2007; Skobeltsyn et al., 2008) is a straightforward but powerful method. In this pruning method, namely PP, pruning is performed based on a score assigned for each term using a query log. This score is computed by dividing term frequency by document frequency. Term frequency is the number of queries containing the term in the query log; in other words, it expresses the popularity of the term. Document frequency represents the number of documents that the term appears in the collection. Terms are sorted according to their scores and those with the lowest scores are pruned from the index. The pseudocode of this algorithm (based on Altingovde et al. (2012)) is shown in Algorithm 1.

Altingovde et al. (2012) introduced the concept of query view for static index pruning and incorporated it into the five different methods mentioned above. Query view of a document is the re-expression of the document with all the query words, for which the document appeared in the top-k search results in the query log. They achieved significant improvement in preserving the top-ranked results intact in comparison to the pure pruning methods.

- Term-Centric Pruning with Query Views (TCP-QV): In TCP, a posting in the list of term t is discarded from the index if its score is under the threshold, τ_t. As for TCP-QV, before discarding the posting, it is checked whether the term is part of the query view of the document referring to the posting. If not, it is pruned, otherwise kept in the index.
- Document-Centric Pruning with Query Views (DCP-QV): Query views are employed in the DCP algorithm as another key to be used in sorting. All terms in a document are sorted first by whether the term is in query view of the document, QV_d, and then by their scores. Next, the last $\epsilon \times P_t$ terms are omitted.
- Access-based Term-Centric Pruning with Query Views (aTCP-QV): In aTCP-QV, we have two keys to find the postings to be pruned during sorting postings. The primary sort key is whether or not the term is part of the query view of the document indicating the posting. The secondary key is the access count of the document. While pruning, the last $\epsilon \times P_t$ postings are removed.
- Access-based Document-Centric Pruning (aDCP) with Query Views (aDCP-QV): aDCP algorithm eliminates documents starting from with the lowest access counts until the pruning level is reached. But this time, instead of eliminating the document directly, all terms in the document are examined according to their appearance in the query view, QV_d. If a term is not a member of QV_d, the term is discarded from the document, otherwise preserved.

Algorithm 1: Popularity-based Pruning (PP)

```
Input  :     $I$ : inverted index
           $Pop$ : a list indicating popularity of each term in $I$
      $\epsilon$ : pruning level
Output:    $I_p$ : pruned index
for each term $t$ in $I$ do
    get the postings list $P_t$ from $I$
    calculate $s(t) \leftarrow Pop[t]/|P_t|$
    $isNotPruned_t \leftarrow 0$
end
sort terms w.r.t. scores $s(t)$ in descending order
$totalRemainedPostings \leftarrow 0$
while $totalRemainedPostings < \epsilon \times |I|$ do
    fetch the term $t$ with the highest score $s(t)$
    $isNotPruned_t \leftarrow 1$
    $totalRemainedPostings \leftarrow totalRemainedPostings + |P_t|$
end
for each term $t$ in $I$ do
    if $isNotPruned_t$ then
        set $(t, P_t)$ in $I_p$
    end
end
```

- Popularity-based Pruning (PP) strategy with Query Views (PP-QV): This procedure works as follows: if the desired size of the pruned index is not sufficient to keep all query views, the index is pruned according to the highness of the popularity score. If there is still some available space after insertion of the query views of all terms, the query view of the term is substituted with the corresponding complete postings list, starting from the term with the highest popularity.

2.1.3. Recent index pruning approaches for sparse & dense retrieval

In this section, we discuss the most recent advancements for the static index pruning in the context of traditional (sparse) retrieval and dense retrieval.

Recent advances in index pruning for sparse retrieval: Soner et al. (2020) applied some heuristics at document, sentence, and term level relying on summarization while making pruning decisions without requiring the full index construction. Rodriguez and Suel (2018) evaluated the quality of a posting employing a machine learning algorithm (namely Random Forest) with multiple features and also explored the relationship between pruned index size and query processing cost.

To the best of our knowledge, the design and performance of pruning strategies through the lens of topical diversity has not yet been explored in the literature. The only work that performs index pruning together with diversity is conducted by Pehlivan et al. (2013). While pruning, they considered preserving both retrieval performance and *temporal* diversity. To this end, the next posting to be kept is selected in a greedy manner to optimize a relevance-oriented evaluation metric, i.e., DCG. They used two temporal collections in which documents and queries are associated with temporal aspects. The diversification in their work is time-aware; in other words, they diversify documents based on their time interval. Our work differs from the latter in two ways. First, we focus on a more challenging problem, i.e., preserving the topical diversity of the documents (or, postings), rather than the temporal diversity. Secondly, we evaluate the performance of proposed pruning strategies by diversity-aware metrics (such as α-NDCG (Clarke et al., 2008), Precision-IA (Agrawal et al., 2009), etc.) unlike (Pehlivan et al., 2013) that only used relevance metrics (i.e., NDCG and MAP) for evaluation.

Recent advances in index pruning for dense retrieval: Recent advances in pre-trained large language models (LMs) have proven effective in capturing semantic relationships within text. A new approach, called dense retrieval, leverages these LMs to represent both search queries and documents in a low-dimensional space. This space encodes rich information like semantic meaning, sentence structures, language styles, etc. While these neural information retrieval architectures deliver significant improvements in system effectiveness, they come at the cost of high computational expense compared to traditional inverted indexes that utilize simpler ranking functions and benefit from optimized processing algorithms. An alternative approach is known as learned sparse retrieval (LSR). Here, transformer encoders convert queries and documents into sparse vectors. These vectors can then be reduced to a single score for storage within an inverted index. Since static index pruning is a well-established technique for managing index size, it is applied to both dense and sparse retrieval methods. Lassance et al. (2023) explored the effects of various static pruning techniques on inverted indexes constructed using sparse neural retrievers, demonstrating their efficiency through experimental validation with minimal loss. Furthermore, in a follow-up study (Lassance et al., 2024), they introduced a two-stage query processing approach utilizing vectors from a pruned sparse index for initial retrieval. Acquavia et al. (2023) implemented document-centric pruning approaches within dense retrieval index, affirming the insignificance of stopwords and low IDF terms, demonstrating that their embeddings can be safely eliminated from the retrieval process. Liu et al. (2024) presented several document pruning methodologies tailored for late interaction models, aimed at discarding some token embeddings from the dense index.

In our current work, we address the sparse retrieval setup involving traditional inverted index files. However, we believe that our methods and findings in this paper have strong implications for the dense retrieval, too. Specifically, our approaches can be either adopted for learned sparse retrieval scenarios or can be applied for the indexes, over which first-stage retrieval results are obtained and fed to a second-stage dense ranker. We leave exploring these directions as a future work.

2.2. Diversity and fairness in search results

Search result diversification approaches in the literature can be divided into implicit and explicit approaches, depending on how they address query aspects. The implicit approaches employ the inter-similarity of the candidate documents or unsupervised discovery of the underlying query aspects (e.g., Carpineto et al. (2012), Meng et al. (2018), Yu et al. (2018)). The explicit approaches model query aspects explicitly and emphasize the coverage of these aspects in determining the final ranking (e.g., Agrawal et al. (2009), Santos et al. (2010), Yigit-Sert et al. (2020)). An exhaustive survey of diversification methods and metrics for evaluating diversity is provided in Rodrygo et al. (2015).

Diversification methods recently have been studied to tackle search bias (Aktolga & Allan, 2013; Draws et al., 2023; Maxwell et al., 2019) and improve fairness (Karako & Manggala, 2018; McDonald et al., 2019) across search and recommendation results. Diversification in search results aims to bring results with a wide coverage of all possible aspects of a query to improve user satisfaction, and hence, it also encourages topical fairness. As stated in Gao and Shah (2020), there is a correlation between diversity and fairness, and diversity helps improve fairness by introducing different aspects of a topic. In a recent study following this line of research, McDonald et al. (2022) cast search result diversification approaches to generate a fair ranking of users and information sources in academic search.

An earlier study on retrieval bias has also shown that alleviating retrieval bias might improve retrieval performance (Wilkie & Azzopardi, 2014). Static index pruning, however, directly influences the retrievability of documents by removing permanently some postings from the index (Chen et al., 2017). In the light these earlier works that relate diversity and fairness, while evaluating our proposed diversity-aware pruning methods, in addition to typical metrics for diversity effectiveness, we also employ the degree of bias (DB) metric that has been introduced to measure fairness in Gao and Shah (2020).

3. Diversity-aware static index pruning approaches

Index pruning methods in the literature essentially focus on maintaining the relevance effectiveness of top-ranked results obtained over the pruned indexes in comparison to those obtained from the full (unpruned) index. Hence, these methods usually exploit relevance scores based on well-known ranking functions (e.g., BM25) during the pruning, as well as the popularity/access features extracted from previously logged query results, which were again obtained using such functions. We envision that keeping the relevance of results is a necessary but not sufficient principle to guide an index pruning strategy.

Consider the typical search scenario, where user queries are expressed with only few keywords and there may be several *aspects* (a.k.a., subtopics or interpretations) associated with a query topic. An index pruning algorithm that eliminates all index elements (documents and/or postings) related to certain aspects of a query topic would lead a drop in diversity effectiveness and induce accessibility bias (Azzopardi & Vinay, 2008; Lipani, 2019) in top-ranked results, even if all the retrieved results are highly relevant to the query. For instance, if all documents about the ‘Indonesian island Java’ and ‘coffee type Java’ are pruned from the index, the results for the query ‘Java’ can only cover the programming language aspect, implying high effectiveness in terms of relevance, but not diversity. Even worse, applying a diversification algorithm over the retrieved results as a post-processing stage – as typical in the literature – would not be a remedy in this case, as the representatives of other aspects have been permanently removed from the index.

In the light of these discussions, we propose three new diversity-aware index pruning strategies. Our strategies are built upon the access-based methods, namely, aTCP and aDCP (cf. Section 2.1), due to their success shown in an earlier work (Altingovde et al., 2012), and extend them to take diversity into account during the pruning process.[4] Note that, at the time of index pruning, we have no information about the queries and their topical aspects, unlike certain search result diversification approaches (e.g., Rodrygo et al. (2015)) that employ such knowledge. Thus, our strategies exploit the inter-relationships of documents (or, terms) in the index to assess the potential impact of postings on the topical diversity of results and guide the pruning accordingly. In what follows, we discuss the details for each of the proposed pruning strategies.

[4] In case of the unavailability of access-counts and/or query logs, our ideas presented here for aTCP can be directly applied to traditional TCP approach, as well. In contrary, for traditional DCP, our method proposed for aDCP would not be directly applicable and may require extensions, which are not explored in the scope of this work.

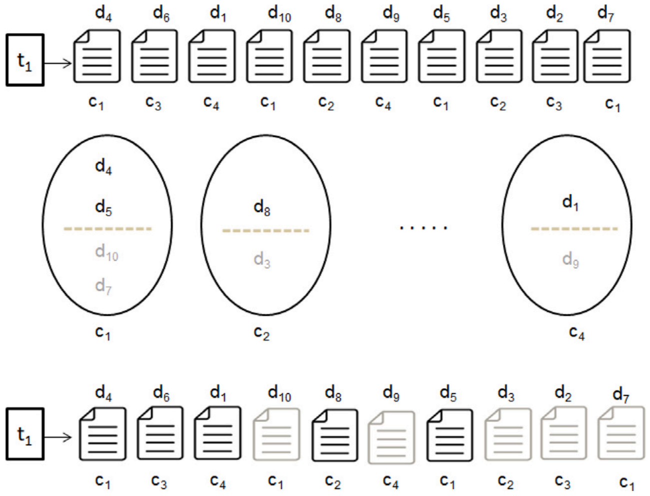

d4 d6 d1 d10 d8 d9 d5 d3 d2 d7
t1
c1 c3 c4 c1 c2 c4 c1 c2 c3 c1
d4
d5
d10
d7
c1
d8
d3
c2
.....
d1
d9
c4
d4 d6 d1 d10 d8 d9 d5 d3 d2 d7
t1
c1 c3 c4 c1 c2 c4 c1 c2 c3 c1

Fig. 4. The illustration of aTCP-Div-Clust algorithm.

3.1. Access-based term-centric diversity-aware pruning with clustering

Our first approach for diversity-aware pruning is built upon the aforementioned aTCP strategy, which sorts each postings list with respect to the access counts of included documents (obtained from a previous query log) and removes those postings with the lowest counts (according to the required pruning level). We extend the latter strategy by considering not only the access count but also the topic of the documents in the postings list during the pruning. In order to do so, we utilize clustering, which is a well-known unsupervised learning approach for the discovery of topics from documents. Clustering algorithms aim to group documents in a way that documents in the same cluster are more similar to each other rather than the documents in other clusters, and each cluster refers to a different topic (Anaya-Sánchez et al., 2010; Gao & Shah, 2020).

We first employ clustering over the entire collection and obtain a general document-cluster mapping. Then, we use this clustering structure to capture the semantics of documents on the term's postings list. Specifically, we create a bucket for each cluster observed in a term's postings list and sort postings in each bucket individually in decreasing order according to their access count, as in the original aTCP strategy. Then, we prune each bucket proportional to the pruning level. This strategy is called as aTCP-Div-Clust, as visualized in Fig. 4. Algorithm 2 gives its pseudo-code.

Algorithm 2: Access-based Term-Centric Diversity-Aware Pruning with Clustering (aTCP-Div-Clust)

```
Input :     $I$ : inverted index
         $AC_D$ : a list indicating an access count value for each document in $D$
            $\epsilon$ : pruning level
          $map$ : the document-cluster mapping
for each term $t$ in $I$ do
    get the postings list $P_t$ from $I$
    create buckets for $P_t$ based on $map$
    for each bucket $b$ do
        sort postings in $b$ w.r.t. access counts, $AC_D$, in descending order
        $NumPrunedPostings \leftarrow 0$
        while $NumPrunedPostings < \epsilon \times |b|$ do
            remove the document $d$ with the lowest access count
            $NumPrunedPostings \leftarrow NumPrunedPostings + 1$
        end
    end
end
```

For our aTCP-Div-Clust strategy, we employ document-clustering mapping generated by the traditional k-means approach with query-driven seeds extracted from a query log (Dai et al., 2016). As an alternative, we also opt to obtain the document-clustering mapping by leveraging URLs. URLs serve as highly informative and valuable sources for acquiring semantic information about the document and are employed in various contexts, including topic classification (Baykan et al., 2011; Souza et al., 2015) and document annotation (Arya & Dwivedi, 2016). Hence, we benefit from URLs of documents to create clusters, rather using the document content (more details are provided in the next section). The variant of our pruning strategy employing URL-based clusters is referred to as aTCP-Div-URL.

3.2. Access-based term-centric diversity-aware pruning with word embeddings

Topical diversity of search queries (and hence, documents) can be examined at different levels, as suggested by the notions of *extrinsic* and *intrinsic* diversity (Raman et al., 2014). Extrinsic diversity arises due to ambiguous or underspecified queries, such as our running example query, ‘Java’, which has substantially different aspects, such as ‘programming language’ or ‘island’. The cluster-based approach presented in the previous section would help such queries, as it is very likely that documents belonging to different aspects of the term ‘Java’ would end up in different clusters and a portion of each cluster would be retained during pruning (as illustrated in Fig. 4). In contrary, intrinsic diversity is required in the scenarios where users require information for the different aspects of the *same* topic (say, for the query ‘java’, aspects may be ‘tutorials’, ‘network programming’, etc.). It would be challenging to obtain such fine-grain clusters where documents must be grouped together based on rather subtle differences. Therefore, we propose an alternative approach, and instead of clustering the entire collection, for each term, we aim to determine the documents in its postings list that might be semantically relevant to the several (possibly, diverse) topical aspects of this term.

Our new strategy has two steps. First, we determine the set of terms E_t that represents the relevant and (potentially) diverse aspects of a given term t (i.e., as if we have a query including only this term t). Next, we compute a score for each document d appearing in the postings list of t, based on whether d also appears in the lists of terms in E_t. For instance, in our running example, for the term ‘java’, the set E_t may include terms like ‘programming’, ‘web’, ‘network’ as well as ‘island’ and ‘coffee’; and hence, documents appearing in a larger number of these lists are more likely to cover one or more aspects of ‘java’.

In order to determine the set of terms E_t, we employ the MMRE method (based on the well-known MMR approach of Carbonell and Goldstein (1998)) introduced by Bouchoucha et al. (2013). MMRE is a query expansion approach (see Azad and Deepak (2019) for a survey) that aims to select terms that are both relevant to the original query and covering its diverse aspects. In our case, we apply MMRE to each term in the index, as we do not have queries during the indexing time. For a given term t, MMRE iteratively selects aspect terms w yielding the highest scores based on the following formula:

$$score_{MMRE}(w) = \lambda Sim(w,t) - (1-\lambda)\max_{w' \in E_t} Sim(w,w') \tag{1}$$

where w is a candidate term in the index, $Sim(w,t)$ is the relevance of w to term t, and E_t is the set of aspect terms already selected. λ is a trade-off parameter that adjusts the balance between relevance and diversity. Basically, in each iteration, MMRE aims to select a term that is similar to term t, but also different from those terms that are already selected in E_t. In Eq. (1), we compute the pairwise similarity of two terms, $Sim(t_i,t_j)$, in terms of the Cosine similarity of the word embedding (WE) vectors of these terms, as in the previous works (e.g., Küçükoğlu (2019), Liu et al. (2014)).

Note that, rather than selecting a fixed number of aspect terms per index term, we employ an adaptive strategy based on a score threshold th. In particular, to include both relevant and diverse terms for t, we select all the terms w with $score_{MMRE}(w) > th$ into E_t. We further enrich E_t with highly relevant terms w such that $Sim(t,w) > th$. The term t itself is also included in E_t for the correctness of the next step of the algorithm.

In the second step of our proposed method, we compute a relevance score (using, say, tf-idf or BM25) for each document d appearing in the postings list of a given term t. Additionally, if such a document d also appears in the postings lists of the aspect terms in E_t, we compute the relevance score contributions from these terms as well, and add them up. By doing so, we aim to assign higher scores to documents that are related to several possible aspects of t. As access count is shown to be a strong indicator in identifying documents to be pruned in earlier works (e.g., Altingovde et al. (2012)), it is also taken into account for the final score of the document d. Thus, the score of each document (and hence, posting p) in the postings list of term t is computed as follows[5]:

$$Score_{(t,p)} = log(AC[d]) \cdot \left[s(t,d) + \sum_{w \in E_t} s(w,d) \right] \tag{2}$$

where $AC[d]$ represents the access count value (log-smoothed) of the posting p including document d. $s(t,d)$ and $s(w,d)$ are the relevance scores of the document d with respect to term t and word w ($\in E_t$), respectively. Once the scoring of documents in the postings list is completed, they are sorted in descending order and pruned, starting from the lowest score until the desired pruning level is reached. Algorithm 3 presents this pruning strategy, which we call aTCP-Div-WE.

Our choice of employing MMRE term expansion method in the first step of aTCP-Div-WE approach is based on its various advantages. First, being based on the well-known implicit diversification method MMR, MMRE is practical to adapt to our scenario and relatively fast to compute. Secondly, it does not need context from other terms, which is crucial as we have to apply it to each term in the index on its own. Third, it is amenable to be used with term embeddings. Having said that, it is also possible to employ alternative term expansion methods and/or term embeddings in the first step of algorithm, which are not further explored in the scope of this work.

Finally note that, although our algorithm is to be applied offline (i.e., during the index pruning phase), it may be still expensive; as for each term, all the other index terms are considered as candidates while constructing the set E_t. As a remedy, we can restrict the application of this algorithm to a certain subset of terms in the index, such as the terms appearing in the previous query logs or terms whose postings list length is neither too long nor too short (i.e., targeting the terms that are not in the head or tail of the document frequency distribution, respectively). In this paper, we focus on the impact of our proposed algorithm on result diversity, and leave such efficiency optimizations as a future work.

[5] There is potential for improvement in Eq. (2) by employing a weighted combination of scores and through parameter tuning, which we leave as a future direction to explore.

Algorithm 3: Access-based Term-Centric Diversity-Aware Pruning with Word Embeddings (aTCP-Div-WE)

```
Input :   $I$ : inverted index
       $AC_D$ : a list indicating an access count values for each document in $D$
          $\epsilon$ : pruning level
        $exW$ : the set of representing expanded (diverse or similar) words of the terms

for each term $t$ in $I$ do
    get the postings list $P_t$ from $I$
    get the expanded words of $t$ from $exW$, $E_t$
    for each document $d$ in $P_t$ do
        compute query-document score, $s(t, d)$
        for each word $w$ in $E_t$ do
            compute a document score, $s(w, d)$
            $s(t, d) \leftarrow s(t, d) + s(w, d)$
        end
        $s(t, d) \leftarrow s(t, d) \times log(AC[d])$
    end
    sort documents in $P_t$ in descending order w.r.t. $s(t, d)$
    remove the last $\epsilon \times |P_t|$ from $P_t$
end
```

3.3. Access-based document-centric diversity-aware pruning with clustering

Different from the previous term-centric strategies, this strategy is built upon aDCP, a document-centric approach, which sorts the documents in the collection based on their access count and then prunes those with the lowest count values. Here, we again benefit from clustering to keep the topical variety of documents during pruning, in a similar fashion to aTCP-Div-Clust approach.

In our strategy, we place each document in the collection to the corresponding bucket using a document-cluster mapping and then sort each bucket in descending order of their access counts. We eliminate the documents within each bucket entirely from the collection, in the ratio of the pruning level, starting from the document with the lowest access count. Consequently, we retain documents with both high access counts and, potentially, a diverse range of topics. To break the ties among documents with the same access counts, we use document IDs as an intermediate key for sorting. Our new pruning strategy, called aDCP-Div-Clust, is shown in Algorithm 4. In Fig. 5, we illustrate how aDCP-Div-clust operates. Note that, while aDCP-Div-Clust employs the clustering structure generated by the k-means approach of Dai et al. (2016), as in Section 3.1, we also employ URL-based clusters in a variant referred to as aDCP-Div-URL.

Algorithm 4: Access-based Document-Centric Diversity-Aware Pruning with Clustering (aDCP-Div-Clust)

```
Input :   $I$ : inverted index
       $AC_D$ : a list indicating an access count value for each document in $D$
          $\epsilon$ : pruning level
        $map$ : a document-cluster mapping

create buckets based on the $map$
for each bucket $b$ do
    sort documents in $b$ w.r.t. access counts in descending order using $AC_D$
    $NumPrunedPostings \leftarrow 0$
    while $NumPrunedPostings < \epsilon \times |b|$ do
        remove the document $d$ with the lowest access count from the bucket $b$
        $NumPrunedPostings \leftarrow NumPrunedPostings + 1$
    end
end
```

4. Experimental setup

In this study, we used the Category B subset of the ClueWeb09 collection that[6] contains 50 million English pages crawled from the web in 2009. The dataset is indexed by the Zettair IR system.[7] In the course of indexing, stop-words and numbers are involved,

[6] More recent datasets such as ClueWeb22 or MS MARCO lack aspect-level relevance judgments and hence, cannot serve as a testbed for evaluating result diversity.

[7] www.seg.rmit.edu.au/zettair/

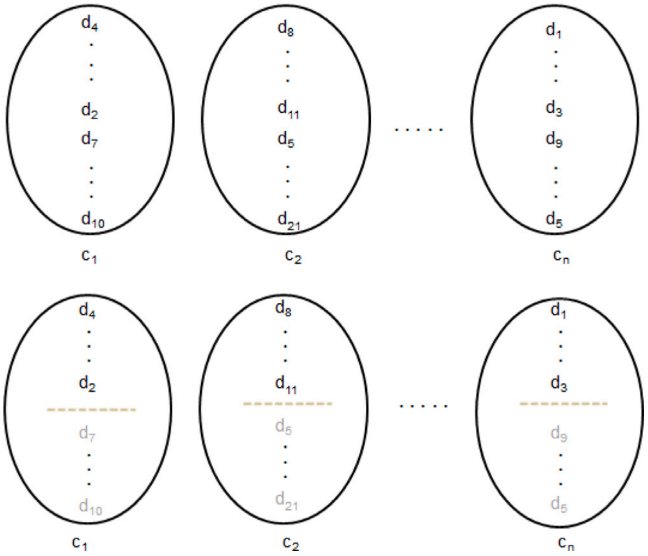


Fig. 5. The illustration of aDCP-Div algorithm.

but no stemming was applied. The resulting index yields a dictionary containing about 160 million terms and occupies 136 GB in disk storage.

The performance of the proposed pruning strategies is assessed through our experiments conducted on this index. We pruned the index at various pruning levels by adjusting the parameter ϵ, which ranges from 60% to 90%. We have not experimented with lower pruning levels since, as indicated in Altingovde et al. (2012), the effectiveness of pruning algorithms becomes more pronounced at higher pruning levels, where the relevance effectiveness of the pruned index is comparatively lower than that of the unpruned index.

We employ query (topic) sets that have been developed for the Diversity Task of the TREC Web Track between 2009 and 2012. Each set includes 50 queries (except for 2010, which has 48) and relevance judgments. The queries are executed in the disjunctive mode on the pruned index, and top-1000 results per query are retrieved by using our own retrieval system implementing the traditional BM25 model. We opt for BM25 because it is still the most widely used IR model, especially for the first-stage retrieval, in various industry-level settings (e.g., Elasticsearch[8] and commercial search engines, such as Yahoo (Yin et al., 2016)) and even in the recently proposed dense retrieval approaches (Macdonald et al., 2021; Mallia et al., 2021). This is due to the fact that BM25 can be computed very efficiently over traditional inverted indexes, and yields a result set with high recall. Furthermore, employing BM25 would ease to replicate our experiments, as it does not require additional features apart from the documents' text. Therefore, we employ BM25 in our proposed aTCP-based algorithm while computing the scores of documents, as well. The parameters of BM25, namely k1 and b, are set experimentally to 1.2 and 0.5, respectively. We also follow the practice in Blanco and Barreiro (2007a) where terms that have document frequency, df, greater than $\frac{N}{2}$ are discarded from the index as they generate negative scores in the BM25's idf formula, $log(\frac{N-df+0.5}{df+0.5})$, where N is the total number of documents in the collection. As shown in the studies conducted on ClueWeb09 collection, spam filtering substantially improves the initial retrieval performance. Thus, we use the spam filtering method in Cormack et al. (2011) and consider documents with a spam score of 60 or less as containing spam, and remove these documents during ranking.

Note that, while computing the access count of documents, we go through the top-1000 search results of a training query set, namely, 1.8 million distinct queries extracted from the AOL query log, as in Altingovde et al. (2012). To break ties for the documents with the same access counts, following the practice in Altingovde et al. (2012), we utilize URLs of documents and sort them in lexicographical order.

Clustering. We use the document-cluster mapping generated by QKLD-QInit, which is found to be the best-performing method in Dai et al. (2016) for the ClueWeb09 Category B dataset. Their method is based on k-means algorithm and to obtain so-called query-driven cluster seeds, it employs Word Embeddings – generated by Continuous Bag-of-Words Model (CBOW) (Mikolov et al., 2013) – of each term extracted from a query log. For the document partitioning phase of k-means, each document is represented by a tf-idf vector. The resulting mapping contains 123 clusters.

For URL-based clustering, we use the second-level domain attribute of a URL as the primary factor for determining the topic of a document. Initially, we obtained nearly 3 million clusters which are hard to extract meaningful and comprehensible information

[8] https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html

Table 1
Diversity effectiveness of the PP algorithm (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. Org.

Pruning level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.1278*(−51.4%)	0.1814*(−49.2%)	0.0783*(−49.7%)	0.3112*(−45.5%)
80%	0.2023*(−23.1%)	0.2807*(−21.3%)	0.1241*(−20.3%)	0.4557*(−20.1%)
70%	0.2162*(−17.8%)	0.3020*(−15.4%)	0.1358*(−12.8%)	0.4941*(−13.4%)
60%	0.2201*(−16.3%)	0.3086*(−13.5%)	0.1353*(−13.2%)	0.5126*(−10.1%)

* Denotes a statistically significant difference from Org at 0.05 level.

to be used in pruning. We address this issue by sorting clusters based on their sizes, selecting the largest 999 clusters, and merging the remaining clusters into a single large cluster labeled as "Others". This procedure yields a total of 1000 clusters.

Baselines. As a baseline, we essentially use the performance of the unpruned index and interpret our results in a way that how we are close to these results. We also employ aTCP and aDCP algorithms to show the improvements of our proposed pruning strategies over them.

Evaluation. We report results in terms of the widely used diversity metrics, namely, ERR-IA (Chapelle et al., 2009), α-nDCG (Clarke et al., 2008), Precision-IA (Agrawal et al., 2009) and ST-recall (Zhai et al., 2003), at the cut-off value of 20. The ndeval software[9] are used to compute the diversity metrics. We use the Student's two-tailed paired t-test (at 95% confidence level) for analyzing statistical significance. To measure fairness, we employ an entropy-based fairness metric, i.e., Degree of Bias (DB) (Gao & Shah, 2020). The degree of bias under the statistical parity constraint assesses the level of exposure of different aspects in the ranking. This evaluation (formulated in Eq. (3)) involves computing entropy and comparing it to the ideal ranking.

$$DB = \frac{H^{ideal} - \sum_j^g p_j \log_2 p_j}{H^{ideal}} \tag{3}$$

where g represents the number of different aspects associated with the query, and p_j is the probability of documents from the aspect j in the current ranking. H^{ideal} indicates the entropy of the ideal ranking that is characterized by an equal number of documents from each aspect, and it yields a value of $log_2 g$ when computing the entropy formula.

5. Experimental results

We conduct our experiments mentioned in this section to seek an answer to the following questions.

- RQ1: How does static index pruning using the most-competitive strategies from the literature, namely, PP, aTCP and aDCP algorithms, affect the diversity of query results?
- RQ2: How does the proposed strategies that take into account the topical diversity perform in terms of diversity effectiveness?
- RQ3: Can we achieve both fair and diverse rankings after pruning?

In the following Sections 5.1, 5.2 and 5.3, we present our findings for each of these questions, respectively.

5.1. Impact of index pruning on the diversity of query results

Since earlier studies evaluate performance of static index pruning using only relevance metrics, we begin with investigating the diversity performance of three strategies, PP, aTCP, and aDCP, which were reported as the most competitive pruning approaches in Altingovde et al. (2012).

Table 1 presents the diversity effectiveness of the PP algorithm at different pruning levels. The "Org" tag in the table denotes the performance when there is no pruning over the index. As expected, when the pruning level increases, the gap between the results of Org and PP widens due to an increase in the number of pruned documents. It can be seen that when using 40% of the original index, the results are substantially lower, reaching up to a 16% drop in the ERR-IA metric. As the pruning level increases, the performance loss reaches around 50 percent. For instance, at 90% pruning, PP yields 0.1278, 0.1814, 0.0783, and 0.3112, whereas the scores of Org are 0.2629, 0.3568, 0.1558, and 0.5705, suggesting a relative decline of 51.4%, 49.2%, 49.7%, and 45.5% for ERR-IA, α-nDCG, P-IA, and ST-Recall, respectively. It is important to note that the observed performance declines in terms of all diversity metrics are statistically significant, that validates our motivation in this work to develop diversity-aware pruning methods.

Table 2 displays the performance of the aTCP algorithm at different pruning levels. The trends observed in Table 2 are similar to those in Table 1, indicating that the decline in diversity performance becomes more emphasized at higher levels of pruning. When we compare the performance of aTCP to Org (i.e., the unpruned index), the relative decline is 6.3%, 6.2%, 10.6%, and 4.8% at the 60% pruning for ERR-IA, α-nDCG, P-IA, and ST-Recall, respectively. However, aTCP outperforms PP for all pruning levels and

[9] http://trec.nist.gov/data/web10.html

Table 2
Diversity effectiveness of the aTCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. Org.

Pruning level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.1612*(−38.7%)	0.2116*(−40.7%)	0.0781*(−49.9%)	0.3492*(−38.8%)
80%	0.2212*(−15.9%)	0.2946*(−17.4%)	0.1152*(−26.1%)	0.4774*(−16.3%)
70%	0.2393*(−9.0%)	0.3228*(−9.5%)	0.1308*(−16.0%)	0.5184*(−9.1%)
60%	0.2464*(−6.3%)	0.3348*(−6.2%)	0.1393*(−10.6%)	0.5432(−4.8%)

* Denotes a statistically significant difference from Org at 0.05 level.

Table 3
Diversity effectiveness of the aDCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. Org.

Pruning level	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	0.2629	0.3568	0.1558	0.5705
90%	0.2165*(−17.6%)	0.2899*(−18.8%)	0.1140*(−26.8%)	0.4619*(−19.0%)
80%	0.2405*(−8.5%)	0.3260*(−8.6%)	0.1360*(−12.7%)	0.5270*(−7.6%)
70%	0.2545(−3.2%)	0.3429*(−3.9%)	0.1439*(−7.6%)	0.5516*(−3.3%)
60%	0.2586(−1.6%)	0.3509(−1.7%)	0.1488*(−4.5%)	0.5651(−0.9%)

* Denotes a statistically significant difference from Org at 0.05 level.

metrics except P-IA. For instance, aTCP yields scores of 0.2116, 0.2946, 0.3228, and 0.3348, while PP can only achieve 0.1814, 0.2807, 0.3020, and 0.3086 for α-nDCG metric at the pruning levels ranging from 90% to 60%.

Table 3 shows the performance of the aDCP pruning algorithm in terms of diversity effectiveness. We again observe that scores of diversity metrics drop when the pruning level increases. Additionally, we notice that aDCP consistently outperforms PP and aTCP across all metrics, and it significantly narrows the gap with the unpruned index especially for less aggressive pruning levels, i.e., at 60% and 70%. Even for the level of 90% pruning, while aTCP causes a performance drop of 38.7%, 40.7%, 49.9%, and 38.8%, aDCP exhibits a significantly lower performance drop with values of 17.6%, 18.8%, 26.8%, and 19.0% in terms of ERR-IA, α-nDCG, P-IA, and ST-Recall metrics, respectively.

Our findings in this section reveal that static index pruning may considerably diminish the diversity of query results, justifying our motivation for developing diversity-aware pruning approaches in this work. As both aTCP and aDCP strategies are found to outperform PP in diversity effectiveness, in the rest of this paper, we only employ aTCP and aDCP as the baselines and do not report any further results for PP.

5.2. Performance of diversity-aware index pruning strategies

To answer our second research question, we evaluate our diversity-aware approaches under different scenarios. In what follows, we first present the diversity effectiveness of the proposed approaches aTCP-Div-Clust, aTCP-Div-WE, aDCP-Div-Clust and aDCP-Div-URL, as discussed in Section 3. Next, for the best performing strategies, we provide evaluation results of their variants enriched with the query views.

Performance of aTCP-Div-Clust and aTCP-Div-WE strategies. Table 4 compares the diversity effectiveness of three cases: (i) when there is no pruning, (ii) pruning with the baseline aTCP, and (iii) pruning based on our proposed diversity-aware aTCP strategies, i.e., aTCP-Div-Clust and aTCP-Div-WE. In the diversity-aware aTCP strategy based on clustering, we utilize two document-clustering mappings: URL-based mapping and QKLD-QInit mapping, as described in Section 4. In Table 4, for the sake of brevity, we prefer to present the results of the latter case, as it consistently yields better performance.

According to Table 4, aTCP-Div-Clust algorithm beats the original aTCP algorithm in most metrics, while aTCP-Div-WE demonstrates superior performance across all metrics. aTCP-Div-WE provides the best gains at the 90% pruning yielding relative improvements of 29.0% (0.1612 → 0.2078), 28.6% (0.2116 → 0.2721), 47.0% (0.0781 → 0.1149), and 24.9% (0.3492 → 0.4362). Even at the 60% pruning, aTCP-Div-WE provides a significant improvement reaching 7.2% (0.2464 → 0.2640) over aTCP. Subsequently, when we compare the performance of our diversity-aware aTCP-Div-WE to the unpruned index ("Org"), we see that the adverse effects of pruning is less emphasized, in comparison to using the baseline aTCP. At 60% pruning, aTCP-Div-WE causes a reduction of 1.4% (0.3568 vs. 0.3519), 6.6% (0.1558 vs. 0.1455), and 3.7% (0.5705 vs. 0.5492) while aTCP's diminishes the effectiveness by 6.2% (0.3568 vs. 0.3348), 10.6% (0.1558 vs. 0.1393), and 4.8% (0.5705 vs. 0.5432) for α-nDCG, P-IA, and ST-Recall metrics, respectively. Note that, for the latter case, aTCP-Div-WE algorithm has no adverse effect in terms of ERR-IA scores (0.2629 vs. 0.2640), while aTCP again causes a relative drop of 1.6% (cf., the last row of Table 2).

Note that, for a few cases, there seems no improvement on some diversity metrics w.r.t. the baseline method. This happens, because what we essentially report is the diversity performance over query results, i.e., rankings obtained over the index; but not a score directly reflecting the diversity of the entire index. Our algorithms are intended to keep the index to cover as diverse topics as possible for a future query stream; but, obviously, this cannot be guaranteed for all possible queries and at every pruning level.

Table 4
Diversity effectiveness of the diversity-aware aTCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. aTCP method.

Pruning level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	–	0.2629	0.3568	0.1558	0.5705
90%	aTCP	0.1612	0.2116	0.0781	0.3492
	aTCP-Div-Clust	0.1823*(13.1%)	0.2373*(12.1%)	0.0886(13.4%)	0.3919*(12.2%)
	aTCP-Div-WE	**0.2078***(29.0%)	**0.2721***(28.6%)	**0.1149***(47.0%)	**0.4362***(24.9%)
80%	aTCP	0.2212	0.2946	0.1152	0.4774
	aTCP-Div-Clust	0.2254(1.9%)	0.2961(0.5%)	0.1147(−0.4%)	0.4784(0.2%)
	aTCP-Div-WE	**0.2331**(5.4%)	**0.3100**(5.2%)	**0.1286**(11.7%)	**0.5044**(5.7%)
70%	aTCP	0.2393	0.3228	0.1308	0.5184
	aTCP-Div-Clust	0.2355(−1.6%)	0.3187(−1.3%)	0.1315(0.5%)	0.5176(−0.2%)
	aTCP-Div-WE	**0.2539**(6.1%)	**0.3375**(4.5%)	**0.1358**(3.8%)	**0.5411**(4.4%)
60%	aTCP	0.2464	0.3348	0.1393	0.5432
	aTCP-Div-Clust	0.2519(2.2%)	0.3386(1.1%)	0.1423(2.1%)	0.5390(−0.8%)
	aTCP-Div-WE	**0.2640***(7.2%)	**0.3519***(5.1%)	**0.1455**(4.4%)	**0.5492**(1.1%)

* Denotes a statistically significant difference from aTCP at 0.05 level.

Table 5
Diversity effectiveness of the diversity-aware aTCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels in terms of ST-Recall and P-IA metrics at ranks 50 and 100. In parentheses, we report the percentage change w.r.t. aTCP method.

Pruning level	Method	P-IA@50	P-IA@100	ST-Recall@50	ST-Recall@100
Org	–	0.1173	0.0811	0.6721	0.7173
90%	aTCP	0.0533	0.0328	0.4132	0.4432
	aTCP-Div-Clust	0.0581(9.1%)	0.0358(9.0%)	0.4546*(10.0%)	0.4775*(7.7%)
	aTCP-Div-WE	**0.0793***(49.0%)	**0.0573***(74.6%)	**0.4932***(19.4%)	**0.5588***(26.1%)
80%	aTCP	0.0810	0.0519	0.5399	0.5758
	aTCP-Div-Clust	0.0776(−4.2%)	0.0528(1.9%)	0.5367(−0.6%)	0.5752(−0.1%)
	aTCP-Div-WE	**0.0884**(9.2%)	**0.0599***(15.5%)	**0.5663**(4.9%)	**0.6120**(6.3%)
70%	aTCP	0.0937	0.0628	0.6026	0.6381
	aTCP-Div-Clust	0.0942(0.6%)	0.0620(−1.3%)	0.6009(−0.3%)	0.6371(−0.2%)
	aTCP-Div-WE	**0.0978**(4.4%)	**0.0669**(6.5%)	**0.6168**(2.4%)	**0.6582**(3.1%)
60%	aTCP	0.0996	0.0683	**0.6417**	**0.6808**
	aTCP-Div-Clust	0.1007(1.1%)	0.0680(−0.4%)	0.6306(−1.7%)	0.6665(−2.1%)
	aTCP-Div-WE	**0.1016**(2.1%)	**0.0704**(3.1%)	0.6263(−2.4%)	0.6789(−0.3%)

* Denotes a statistically significant difference from aTCP at 0.05 level.

Still, our results reported in this section show that the proposed algorithms achieve maintaining (or, sometimes, even improving) the result diversity across several metrics and different scenarios.

Table 5 compares the performance of aTCP and our diversity-aware aTCP algorithms in terms of P-IA and ST-Recall metrics at rank cut-off values of 50 and 100. We choose this experimental setup to investigate the performance when the results sets (or rankings) generated over the pruned indexes are intended to serve as *candidate sets* for a subsequent diversification stage, as typical in the literature (e.g., Rodrygo et al. (2015)). As this scenario requires a large result set, we focus on top-50 and top-100 results; and since only the inclusion of diverse documents in the results lists are important (but not their ranks), we employ P-IA and ST-Recall metrics. The results in Table 5 confirm the superiority of the diversity-aware approaches also for this setup. Our aTCP-Div-WE strategy outperforms the aTCP baseline for almost all pruning levels and metrics (with the exception of ST-recall metric at 60% pruning). For instance, aTCP-Div-WE provides relative improvements 49.0%, 74.6%, 19.4%, and 26.1% over aTCP at 90% pruning for the metrics P-IA@50, P-IA@100, ST-Recall@50 and ST-Recall@100, respectively.

Performance of aDCP-Div-Clust and aDCP-Div-URL strategies. Table 6 shows the effectiveness of our diversity-aware aDCP pruning strategies versus aDCP. For most cases, the proposed aDCP strategies, namely aDCP-Div-Clust and aDCP-Div-URL, outperform aDCP yielding relative improvements up to 1.9%, 2.0%, 2.6%, and 2.5% for ERR-IA, α-nDCG, P-IA, and ST-Recall, respectively. The improvements are moderate compared to diversity-aware aTCP algorithms because aDCP is a strong baseline to beat and already the best-performing baseline method according to Tables 1–3. Despite pruning 60% of the index, aDCP-Div-URL achieves scores that are close to those of the unpruned index. The relative decline in performance compared to the unpruned index is only 2.1% (0.2629 vs. 0.2574), 1.8% (0.3568 vs. 0.3505), 4.3% (0.1558 vs. 0.1491), and 0.4% (0.5705 vs. 0.5685) for ERR-IA, α-nDCG, P-IA, and ST-Recall, respectively. In the comparison of diversity-aware aDCP approaches, the aDCP strategy employing URL-based document-cluster mapping demonstrates better performance than that of employing QKLD-QInit mapping. Lastly, the aDCP-Div-Clust strategy produces higher metric scores in comparison to the best-performing approach discussed in the previous section, i.e., aTCP-Div-WE.

Table 6
Diversity effectiveness of the diversity-aware aDCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. aDCP method.

Pruning level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	–	0.2629	0.3568	0.1558	0.5705
90%	aDCP	0.2165	0.2899	0.1140	0.4619
	aDCP-Div-Clust	0.2172(0.3%)	0.2909(0.4%)	**0.1170***(2.6%)	0.4659(0.9%)
	aDCP-Div-URL	**0.2207**(1.9%)	**0.2957**(2.0%)	0.1163*(2.0%)	**0.4735***(2.5%)
80%	aDCP	0.2405	0.3260	0.1360	0.5270
	aDCP-Div-Clust	0.2377(−1.2%)	0.3230(−0.9%)	0.1347(−1.0%)	**0.5279**(0.2%)
	aDCP-Div-URL	**0.2412**(0.3%)	**0.3263**(0.1%)	**0.1367**(0.5%)	0.5254(−0.3%)
70%	aDCP	**0.2545**	**0.3429**	0.1439	**0.5516**
	aDCP-Div-Clust	0.2516(−1.1%)	0.3404(−0.7%)	0.1436(−0.2%)	0.5499(−0.3%)
	aDCP-Div-URL	0.2540(−0.2%)	0.3428(0.0%)	**0.1443**(0.2%)	0.5508(−0.1%)
60%	aDCP	0.2586	0.3509	0.1488	0.5651
	aDCP-Div-Clust	**0.2589**(0.1%)	**0.3515**(0.2%)	**0.1500**(0.8%)	0.5669(0.3%)
	aDCP-Div-URL	0.2574(−0.5%)	0.3505(−0.1%)	0.1491(0.2%)	**0.5685**(0.6%)

* Denotes a statistically significant difference from aDCP at 0.05 level.

Table 7
Diversity effectiveness of the diversity-aware aDCP algorithm (over TREC 2009–2012 topic sets) at different pruning levels in terms of ST-Recall and P-IA metrics at ranks 50 and 100. In parentheses, we report the percentage change w.r.t. aDCP method.

Pruning level	Method	P-IA@50	P-IA@100	ST-Recall@50	ST-Recall@100
Org	–	0.1173	0.0811	0.6721	0.7173
90%	aDCP	0.0752	0.0462	0.5221	0.5493
	aDCP-Div-Clust	**0.0778***(3.4%)	**0.0486***(5.3%)	0.5286(1.2%)	0.5627(2.4%)
	aDCP-Div-URL	0.0763(1.4%)	0.0469*(1.5%)	**0.5340**(2.3%)	**0.5669***(3.2%)
80%	aDCP	**0.0975**	0.0642	0.6234	0.6511
	aDCP-Div-Clust	0.0974(−0.1%)	0.0645(0.5%)	0.6154(−1.3%)	0.6525(0.2%)
	aDCP-Div-URL	**0.0975**(−0.1%)	**0.0649***(1.0%)	**0.6250**(0.2%)	**0.6531**(0.3%)
70%	aDCP	**0.1033**	0.0695	0.6460	0.6841
	aDCP-Div-Clust	0.1029(−0.4%)	**0.0696**(0.2%)	**0.6464**(0.1%)	0.6875(0.5%)
	aDCP-Div-URL	0.1031(−0.3%)	0.0694(−0.1%)	0.6460(0.0%)	**0.6896**(0.8%)
60%	aDCP	0.1083	**0.0741**	0.6549	**0.7026**
	aDCP-Div-Clust	0.1079(−0.4%)	0.0740(−0.1%)	0.6550(0.0%)	0.7008(−0.3%)
	aDCP-Div-URL	**0.1085**(0.2%)	0.0738(−0.4%)	**0.6564**(0.2%)	0.7022(0.0%)

* Denotes a statistically significant difference from aDCP at 0.05 level.

In Table 7, we inspect the diversity performance of aDCP-based pruning algorithms for the same experimental setup used for Table 5, i.e., generating a candidate result set over the pruned indexes. The results show that there is no clear winner between the two diversity-aware pruning algorithms, yet they outperform the aDCP baseline in most cases.

Performance of aTCP-Div-WE and aDCP-Div-URL strategies with query views. In order to enhance the performance of diversity-aware pruning algorithms, we incorporate query views into our approaches, as they are shown to be useful in the context of static index pruning in Altingovde et al. (2012). Recall that the query view of a document consists of the terms that are appeared as queries that have retrieved this document within the top-k results. As described in Section 2, Altingovde et al. (2012) have expanded aTCP and aDCP approaches to avoid pruning the terms that are in the query views of the documents.

The query view idea is also applicable to our diversity-aware algorithms in a similar manner. As aTCP-Div-WE and aDCP-Div-URL are shown to be the best-performing approaches in the previous section, here we focus on the performance of their variants that incorporate query views. For the sake of simpler naming, we refer the variants of the latter two strategies with query views as aTCP-Div-QV and aDCP-Div-QV, respectively, in the rest of discussion. We compare them to the variants of the baseline algorithms that are again extended with query views, i.e., aTCP-QV and aDCP-QV. The query views of documents are obtained by executing the training query log (that is used for obtaining the access counts, as described in Section 4) over the unpruned index.

The performance of the diversity-aware algorithms using query views at various pruning levels is presented in Table 8. The findings demonstrate that incorporating query views enhances the diversity effectiveness of both algorithms. The diversity-aware aTCP-Div-QV provides gains reaching up to 2.9%, 3.0%, 7.5%, and 3.9%, while diversity-aware aDCP-Div-QV yields improvements up to 0.6%, 0.4%, 0.6%, and 0.4% over the corresponding baselines w.r.t. the ERR-IA, α-nDCG, P-IA, and ST-Recall metrics, respectively. We also provide the performance of diversity-aware algorithms with query views in terms of P-IA and ST-Recall at cut-off values of 50 and 100 in Table 9. Again, we show that our methods provide improvements over the baselines. Furthermore, the inclusion of query views helps narrow the difference with scores obtained over the unpruned index. For instance, using only 40% of the index, aTCP-Div-QV yields the results 0.1121, 0.0770, 0.6625, and 0.7072 versus 0.1173, 0.0811, 0.6721, and 0.7173, in

Table 8
Diversity effectiveness of the diversity-aware aTCP and aDCP algorithms using QVs (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. the corresponding pruning method using QV.

Pruning level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	–	0.2629	0.3568	0.1558	0.5705
90%	aTCP-QV	0.2408	0.3263	0.1347	0.5270
	aTCP-Div-QV	**0.2457**(2.0%)	**0.3357**(2.9%)	**0.1449***(7.5%)	**0.5464***(3.7%)
	aDCP-QV	0.2385	0.3239	0.1375	0.5217
	aDCP-Div-QV	0.2398(0.6%)	0.3252(0.4%)	0.1378(0.2%)	0.5229 (0.2%)
80%	aTCP-QV	0.2486	0.3376	0.1430	0.5463
	aTCP-Div-QV	0.2527(1.7%)	**0.3439**(1.9%)	**0.1506***(5.4%)	**0.5575**(2.1%)
	aDCP-QV	**0.2538**	0.3417	0.1433	0.5489
	aDCP-Div-QV	0.2530 (−0.3%)	0.3414(−0.1%)	0.1431(−0.1%)	0.5509(0.4%)
70%	aTCP-QV	0.2518	0.3395	0.1436	0.5429
	aTCP-Div-QV	**0.2548**(1.2%)	**0.3475** (2.4%)	**0.1504***(4.7%)	**0.5642***(3.9%)
	aDCP-QV	0.2533	0.3432	0.1467	0.5528
	aDCP-Div-QV	0.2544(0.4%)	0.3444(0.3%)	0.1475(0.6%)	0.5543(0.3%)
60%	aTCP-QV	0.2539	0.3439	0.1477	0.5521
	aTCP-Div-QV	**0.2613**(2.9%)	**0.3541***(3.0%)	**0.1523**(3.1%)	**0.5641**(2.2%)
	aDCP-QV	0.2575	0.3499	0.1509	0.5605
	aDCP-Div-QV	0.2578(0.1%)	0.3504(0.1%)	0.1516(0.4%)	0.5621(0.3%)

* Denotes a statistically significant difference from the corresponding pruning method using QV at 0.05 level.

Table 9
Diversity effectiveness of the diversity-aware aTCP and aDCP algorithms using QVs (over TREC 2009–2012 topic sets) at different pruning levels in terms of ST-Recall and P-IA metrics at ranks 50 and 100. In parentheses, we report the percentage change w.r.t. the corresponding pruning method using QV.

Pruning level	Method	PI-IA@50	P-IA@100	ST-Recall@50	ST-Recall@100
Org	–	0.1173	0.0811	0.6721	0.7173
90%	aTCP-QV	0.1016	0.0688	0.6171	0.6686
	aTCP-Div-QV	**0.1087***(7.0%)	**0.0750***(9.0%)	**0.6335***(2.7%)	**0.6772***(1.3%)
	aDCP-QV	0.1018	0.0699	0.6211	06 679
	aDCP-Div-QV	0.1017(−0.1%)	0.0699(0.0%)	0.6224(0.2%)	0.6692(0.2%)
80%	aTCP-QV	0.1054	0.0727	0.6402	0.6881
	aTCP-Div-QV	**0.1110***(5.4%)	**0.0753**(3.6%)	**0.6472**(1.1%)	**0.6931**(0.7%)
	aDCP-QV	0.1063	0.0734	0.6466	0.6883
	aDCP-Div-QV	0.1063(0.0%)	0.0735(0.1%)	0.6466(0.0%)	0.6883(0.0%)
70%	aTCP-QV	0.1066	0.0740	0.6494	0.6894
	aTCP-Div-QV	**0.1117**(4.8%)	**0.0761**(2.9%)	0.6581(1.3%)	**0.6998**(1.5%)
	aDCP-QV	0.1085	0.0739	0.6583	0.6981
	aDCP-Div-QV	0.1083(−0.2%)	0.0741(0.2%)	**0.6609**(0.4%)	0.6977(−0.1%)
60%	aTCP-QV	0.1086	0.0747	0.6606	0.7013
	aTCP-Div-QV	**0.1121**(3.2%)	**0.0770**(3.1%)	**0.6625**(0.3%)	**0.7072**(0.8%)
	aDCP-QV	0.1100	0.0760	0.6583	0.7063
	aDCP-Div-QV	0.1103(0.3%)	0.0761(0.2%)	0.6600(0.3%)	0.7070(0.1%)

* Denotes a statistically significant difference from corresponding pruning method using QV at 0.05 level.

terms of P-IA@50, P-IA@100, ST-Recall@50 and ST-Recall@100, respectively. According to Tables 8 and 9, although aDCP-Div-QV also provides improvements, aTCP-Div-QV achieves the overall best performance.

Note that, methods based on aTCP benefits more from the query views in comparison to aDCP methods. This is due to the logic of the algorithms. The original aTCP algorithm processes each postings list on its own and sorts w.r.t. access count, hence, some postings corresponding to documents with the smallest access counts (say, 0) may still remain in the list. In contrary, the original aDCP algorithm sorts documents w.r.t. access counts and removes the document entirely. This implies that aTCP may keep some postings even for documents with the smallest access count, and this results in inferior performance of aTCP. Incorporating query views enforces keeping some postings and hence, the likelihood that aTCP can keep such useless postings in every list decreases, allowing larger improvements for indexes generated by aTCP with query views. That is, the gains are not only due to the postings protected by the query view, but also those useless postings that has to be pruned to leave space for them.

Lastly, we extend our experiments to assess the diversification performance over the candidate set obtained from the pruned index files. To do so, we apply a diversification algorithm to the top-100 documents that are retrieved from the indexes pruned by our diversity-aware methods with query views (namely, aTCP-Div-QV and aDCP-Div-QV). We employ xQuAD (Santos et al., 2010) on the candidate set, as xQuAD is found to be the best-performing diversification method in several TREC campaigns (Rodrygo et al., 2015). According to Table 10, aTCP-Div-QV consistently yields superior results, while aDCP-Div-QV provides comparable

Table 10
Diversification performance of xQuAD with the diversity-aware aTCP and aDCP algorithms (over TREC 2009–2012 topic sets) at different pruning levels. In parentheses, we report the percentage change w.r.t. the corresponding pruning method.

Pruning level	Method	ERR-IA@20	α-nDCG@20	P-IA@20	ST-Recall@20
Org	–	0.3162	0.4170	0.1769	0.6222
90%	aTCP-QV	0.2925	0.3873	0.1598	0.5807
	aTCP-Div-QV	**0.2948**(0.8%)	**0.3914**(1.0%)	**0.1691**(5.8%)	**0.5858**(0.9%)
	aDCP-QV	0.2897	0.3860	0.1615	0.5835
	aDCP-Div-QV	0.2917(0.7%)	0.3878(0.5%)	0.1623(0.5%)	0.5848(0.2%)
80%	aTCP-QV	0.3014	0.3988	0.1687	0.6018
	aTCP-Div-QV	0.3022(0.3%)	0.4029(1.0%)	**0.1780**(5.5%)	**0.6104**(1.4%)
	aDCP-QV	**0.3073**	**0.4035**	0.1690	0.6037
	aDCP-Div-QV	0.3003(−2.3%)	0.3983(−1.3%)	0.1649(−2.4%)	0.6022(−0.3%)
70%	aTCP-QV	0.2970	0.3966	0.1674	0.6027
	aTCP-Div-QV	**0.3189***(7.4%)	**0.4168***(5.1%)	**0.1772***(5.9%)	**0.6117**(1.5%)
	aDCP-QV	0.3000	0.4016	0.1693	0.6106
	aDCP-Div-QV	0.3033(1.1%)	0.4037(0.5%)	0.1687(−0.4%)	0.6114(0.1%)
60%	aTCP-QV	0.3041	0.4043	0.1711	0.6102
	aTCP-Div-QV	**0.3114**(2.4%)	**0.4137**(2.3%)	**0.1811***(5.8%)	**0.6167**(1.1%)
	aDCP-QV	0.3083	0.4103	0.1717	0.6155
	aDCP-Div-QV	0.3071(−0.4%)	0.4089(−0.3%)	0.1774*(3.3%)	0.6165(0.2%)

* Denotes a statistically significant difference from corresponding pruning method at 0.05 level.

Table 11
Average query processing efficiency over Org (Full Index) and index files produced by aTCP-Div-QV. In parentheses, we report the percentage of reduction w.r.t. Org.

Pruning level	Processing time (seconds)	No. of postings
Org	0.30	226682521.5
90%	0.012 (59.3%)	28001462.5 (87.6%)
80%	0.014 (51.7%)	50077139.5 (77.9%)
70%	0.017 (43.2%)	72152814.75 (68.2%)
60%	0.019 (36.4%)	94228480.25 (58.4%)

results to the corresponding baselines. It can be observed that aTCP-Div-QV achieves the best performance and provides a relative improvement up to 7.4%, 5.1%, 5.9%, and 1.5% for ERR-IA, α-nDCG, P-IA, and ST-Recall, respectively.

For our best-performing approach, aTCP-Div-QV, we also evaluate query processing efficiency in terms of the average execution time and number of postings processed (We do not report efficiency results for all of the pruning methods, as they yield similar gains, and the findings are aligned with the literature). As shown in Table 11, as the pruning level increases, gains in processing efficiency also increase. In line with the earlier findings (Altingovde et al., 2012), the gains in processing efficiency, especially for the execution times, are not directly proportional to pruning levels (i.e., gains in the storage space). This is expected, as the postings from very popular terms or documents (appearing in several queries) may be pruned rather moderately (i.e., less than the target index prune level) by our algorithms. Furthermore, for the execution time measurements, given the small number of queries in TREC sets, some overheads in the retrieval code may also have an impact. In contrast, gains in terms of the number of processed postings are more in conformance with the pruning level; e.g., for the pruning level of 90%, the average number of postings processed (per query) is reduced by 86%.

Overall, our experiments yield a positive answer to our second research question: diversity-aware pruning strategies are superior to their traditional counterparts, i.e., aTCP and aDCP (as well as their variants with query views), in preserving diverse documents in the index, which, subsequently, allows retrieving more diverse query results. Furthermore, our strategies provide considerable gains in processing efficiency in addition to the savings in storage space, as anticipated from static index pruning.

5.3. Fairness performance of diversity-aware index pruning strategies

In order to address our last research question, we perform additional experiments measuring the fairness of a ranking by Degree of Bias metric (Gao & Shah, 2020). Table 12 presents the fairness performance of diversity-aware pruning methods versus their corresponding baselines. The results denote that our methods provide better performance except in one case, i.e., the aDCP-Div-QV method at 70% pruning. For all pruning levels, aTCP-Div-QV has the fairest ranking and offers relative improvements of 1.9% (0.6054 vs. 0.5937), 1.3% (0.5949 vs. 0.5874), 1.9% (0.5946 vs. 0.5834), and 1.2% (0.5866 vs. 0.5794) compared to aTCP-QV. Note that, a lower score indicates better fairness performance.

In Table 13, we evaluate the fairness performance of top-100 rankings. aTCP-Div-QV outperforms its counterpart, namely aTCP-QV, yielding a significant improvement of up to 2.6%. We can say that aDCP-Div-QV does not mainly provide any improvement

Table 12
Fairness performance of the diversity-aware aTCP and aDCP algorithms using QVs (over TREC 2009–2012 topic sets) at different pruning levels in terms of Degree of Bias metric at rank 20. In parentheses, we report the percentage change w.r.t. the corresponding pruning method using QV. Lower score indicates better fairness performance.

Pruning level	Method	DB
Org	–	0.5739
90%	aTCP-QV	0.6054
	aTCP-Div-QV	**0.5937**(1.9%)
	aDCP-QV	0.6150
	aDCP-Div-QV	0.6137(0.2%)
80%	aTCP-QV	0.5949
	aTCP-Div-QV	**0.5874**(1.3%)
	aDCP-QV	0.5964
	aDCP-Div-QV	0.5953(0.2%)
70%	aTCP-QV	0.5946
	aTCP-Div-QV	**0.5834**(1.9%)
	aDCP-QV	0.5913
	aDCP-Div-QV	0.6137(−3.8%)
60%	aTCP-QV	0.5866
	aTCP-Div-QV	**0.5794**(1.2%)
	aDCP-QV	0.5814
	aDCP-Div-QV	0.5803(0.2%)

Table 13
Fairness performance of the diversity-aware aTCP and aDCP algorithms using QVs (over TREC 2009–2012 topic sets) at different pruning levels in terms of Degree of Bias metric at rank 100. In parentheses, we report the percentage change w.r.t. the corresponding pruning method using QV. Lower score indicates better fairness performance.

Pruning level	Method	DB
Org	–	0.5963
90%	aTCP-QV	0.6400
	aTCP-Div-QV	**0.6233***(2.6%)
	aDCP-QV	0.6363
	aDCP-Div-QV	0.6360(0.0%)
80%	aTCP-QV	0.6281
	aTCP-Div-QV	**0.6164**(1.9%)
	aDCP-QV	0.6254
	aDCP-Div-QV	0.6251(0.0%)
70%	aTCP-QV	0.6222
	aTCP-Div-QV	**0.6128**(1.5%)
	aDCP-QV	0.6202
	aDCP-Div-QV	0.6194(0.1%)
60%	aTCP-QV	0.6177
	aTCP-Div-QV	**0.6082**(1.5%)
	aDCP-QV	0.6113
	aDCP-Div-QV	0.6120(−0.1%)

* Denotes a statistically significant difference from corresponding pruning method using QV at 0.05 level.

compared to aDCP-QV, but aTCP-Div-QV achieves 2% gain over aDCP-QV. aTCP-Div-QV shows the best performance giving a score of 0.6082 when more than half of the index chops off, which is slightly inferior to the score of using a full index.

The last table, Table 14, denotes the fairness performance of diversity-aware pruning methods when the xQuAD algorithm is applied. From the fairness perspective, all methods perform considerably better than their counterparts in Table 12, which means that employing an explicit diversification algorithm, xQuAD, also helps improving the fairness of results. It can be stated that diversity-aware methods still outperform their corresponding baselines in most cases. While aTCP-Div-QV achieved the best results at 90% and 80% pruning (0.5580 and 0.5462, respectively), aDCP-QV is the best-performing method at 70% and 60% pruning (0.5355 and 0.5307, respectively).

Table 14
Fairness performance of the diversity-aware aTCP and aDCP algorithms using QVs (over TREC 2009–2012 topic sets) at different pruning levels in terms of Degree of Bias metric at rank 20 after the xQuAD algorithm is employed. In parentheses, we report the percentage change w.r.t. the corresponding pruning method using QV. Lower score indicates better fairness performance.

Pruning	Method	DB
90%	aTCP-QV	0.5589
	aTCP-Div-QV	**0.5580**(1.6%)
	aDCP-QV	0.5629
	aDCP-Div-QV	0.5609(0.4%)
80%	aTCP-QV	0.5513
	aTCP-Div-QV	**0.5462**(0.9%)
	aDCP-QV	0.5497
	aDCP-Div-QV	0.5481(0.3%)
70%	aTCP-QV	0.5472
	aTCP-Div-QV	0.5417(1.0%)
	aDCP-QV	**0.5355**
	aDCP-Div-QV	0.5368(−0.2%)
60%	aTCP-QV	0.5374
	aTCP-Div-QV	0.5381(−0.1%)
	aDCP-QV	**0.5307**
	aDCP-Div-QV	0.5320(−0.2%)

5.4. Summary and implications of findings

The findings presented in the previous sections reveal that our diversity-aware pruning strategies are effective in maintaining diversity - and even fairness - of query results obtained over such pruned indexes, and they all beat the baselines; i.e., their traditional counterparts unaware of diversity. A comparison of our proposed strategies under different scenarios leads to the following practical lessons:

- If it is not possible to apply query views idea (e.g., due to lack of query logs on the collection), our aDCP-Div-URL method outperforms its competitors, aDCP-Div-Clust, aTCP-Div-Clust and aTCP-Div-WE, in most of the cases. This means that employing an appropriate clustering of documents (which is based on URLs in our case) provides adequate guidance to the pruning algorithm to keep documents that would maintain the result diversity.
- When query views are incorporated, they improve the performance of all pruning approaches, as also shown in the literature (Altingovde et al., 2012). The best-performing strategy for various cases (i.e., at the cut-off values of 20, 50, and 100; or after applying the xQuAD diversification algorithm) under several diversity metrics and an entropy-based fairness metric is aTCP-Div-QV, which exploits word-embeddings (recall that this strategy is a renamed version of aTCP-Div-WE with query view). This means that employing query views already helps in keeping diversity, and document clustering does not add more; for further improvement, we should employ a strategy that can capture fine-grain contributions of a posting to diversity. The latter is achieved by our word-embedding-based strategy that determines the postings that may be relevant to possible diverse aspects of a given term (cf. Section 3.2).

Our new strategies make it possible to substantially prune the index (for well-known efficiency purposes), while maintaining the diversity of query results. These strategies are directly applicable to current retrieval systems ranging from verticals (e.g., for product or news search) to Web search engines, i.e., for the scenarios where diversity of results is desirable. It is also possible to apply our strategies in combination with other widely-used efficiency optimization methods, such as dynamic query pruning, index quantization, index document reordering, index sharding, and stop word removal, to name a few. A recent work by Mackenzie and Moffat (2020) provides an excellent overview of various combinations of such optimizations, and reports that these optimizations are mostly additive, i.e., may be applied simultaneously with a low possibility of adverse impacts. Therefore, our approaches can also be employed in combination with other efficiency optimizations to further amplify the efficiency gains.

Overall, we envision that inverted indexes will play an important role in the future of search, too; maybe as an efficient first-stage ranker for dense retrieval models. In such a future, our contribution would be even more impactful, as we allow keeping the efficiency promises of retrieval over an inverted index without sacrificing the result diversity, which would be highly required for subsequent ranking stages.

6. Conclusions

The goal of static index pruning is to permanently eliminate redundant parts of an index in order to reduce file size and enhance query processing speed. In this study, we investigated how static index pruning influences the diversity of query results and how the diversity can be maintained during the pruning process. To achieve this, we introduced three novel strategies that consider the topical diversity of index elements (documents or postings) and prioritize the preservation of those elements relevant to different topical aspects during index pruning. These methods re-use some of the ideas, such as the access-count feature and query views, from the best-performing methods in the literature; but further enrich them using clustering and word-embeddings to directly address topical diversity.

Our extensive experiments showed that typical pruning strategies cause a substantial decline (i.e., up to 50% for some metrics) in the diversity of the results obtained over the pruned indexes. Our diversity-aware approaches remedied such losses to a great extent, and aTCP-Div-QV, which exploits word-embeddings to determine and keep diverse postings during the pruning, outperformed all its competitors. Specifically, aTCP-Div-QV provided gains reaching up to 2.9%, 3.0%, 7.5%, and 3.9% w.r.t. the strongest baseline aTCP-QV, in terms of the ERR-IA, α-nDCG, P-IA, and ST-Recall metrics (at a cut-off value of 20), respectively. Furthermore, aTCP-Div-QV yielded scores that are much closer to those obtained over the full index. At pruning levels ranging from 60% to 90%, our method caused a decline of at most 6% (7%) while aTCP-QV causes up to 9% (14%) drop in α-nDCG@20 (P-IA@20) scores, respectively. Our experiments also revealed the correlation between topical diversity and fairness metrics in our setup, as aTCP-Div-QV outperformed the baseline in terms of a recently introduced fairness metric, namely, DB (degree of bias), too.

We envision that the inverted index will still play an important role in the future of search, though its usage might be at different stages of retrieval compared to those it is employed today, due to the growing use of neural models. Therefore, in our future work, we also plan to devise effective and efficient methods to exploit such neural models while deciding whether an index entry should be kept or pruned, to maintain and even improve the result diversity.

CRediT authorship contribution statement

Sevgi Yigit-Sert: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Ismail Sengor Altingovde:** Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Özgür Ulusoy:** Writing – review & editing, Software, Methodology, Conceptualization.

Data availability

The authors do not have permission to share data.

Acknowledgments

We thank anonymous reviewers for insightful and constructive comments for enhancing the quality of the article. I.S. Altingovde is partially supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) with the grant no. 117E861 and Middle East Technical University (METU) with the grant no. ADEP-312-2024-11484. S. Yigit-Sert is supported by TÜBİTAK under BİDEB 2211/A program.

References

Acquavia, A., Macdonald, C., & Tonellotto, N. (2023). Static pruning for multi-representation dense retrieval. In *Proceedings of the ACM SIGIR* (pp. 7:1–7:10).

Agrawal, R., Gollapudi, S., Halverson, A., & Ieong, S. (2009). Diversifying search results. In *Proceedings of the ACM WSDM* (pp. 5–14).

Aktolga, E., & Allan, J. (2013). Sentiment diversification with different biases. In *Proceedings of the ACM SIGIR* (pp. 593–602).

Altingovde, I. S., Ozcan, R., & Ulusoy, Ö. (2012). Static index pruning in web search engines: Combining term and document popularities with query views. *ACM Transactions on Information Systems*, *30*(1), 2:1–2:28.

Anaya-Sánchez, H., Pons-Porrata, A., & Berlanga-Llavori, R. (2010). A document clustering algorithm for discovering and describing topics. *Pattern Recognition Letters*, *31*(6), 502–510.

Archer, A., Aydin, K., Bateni, M., Mirrokni, V. S., Schild, A., Yang, R., & Zhuang, R. (2019). Cache-aware load balancing of data center applications. *Proceedings of the VLDB Endowment*, *12*(6), 709–723.

Arya, C., & Dwivedi, S. K. (2016). News web page classification using url content and structure attributes. In *Proceedings of the IEEE NGCT* (pp. 317–322).

Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing & Management (IPM)*, *56*(5), 1698–1735.

Azzopardi, L., & Vinay, V. (2008). Accessibility in information retrieval. In *Proceedings of the ECIR* (pp. 482–489).

Baeza-Yates, R., Gionis, A., Junqueira, F., Murdock, V., Plachouras, V., & Silvestri, F. (2007). The impact of caching on search engines. In *Proceedings of the ACM SIGIR* (pp. 183–190).

Baykan, E., Henzinger, M., Marian, L., & Weber, I. (2011). A comprehensive study of features and algorithms for URL-based topic classification. *ACM Transactions on the Web*, *5*(3).

Blanco, R., & Barreiro, A. (2007a). Boosting static pruning of inverted files. In *Proceedings of the ACM SIGIR* (pp. 777–778).

Blanco, R., & Barreiro, Á. (2007b). Static pruning of terms in inverted files. In *Proceedings of the ECIR* (pp. 64–75).

Blanco, R., & Barreiro, A. (2010). Probabilistic static pruning of inverted files. *ACM Transactions on Information Systems*, *28*(1), 1:1–1:33.

Bouchoucha, A., He, J., & Nie, J.-Y. (2013). Diversified query expansion using conceptnet. In *Proceedings of the ACM CIKM* (pp. 1861–1864).

Büttcher, S., & Clarke, C. L. A. (2006). A document-centric approach to static index pruning in text retrieval systems. In *Proceedings of the ACM CIKM* (pp. 182–189).

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the ACM SIGIR* (pp. 335–336).

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., & Soffer, A. (2001). Static index pruning for information retrieval systems. In *Proceedings of the ACM SIGIR* (pp. 43–50).

Carpineto, C., D'Amico, M., & Romano, G. (2012). Evaluating subtopic retrieval methods: Clustering versus diversification of search results. *Information Processing & Management (IPM)*, *48*(2), 358–373.

Carpineto, C., de Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, *19*(1), 1–27.

Chapelle, O., Metlzer, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the ACM CIKM* (pp. 621–630).

Chen, R.-C., Azzopardi, L., & Scholer, F. (2017). An empirical analysis of pruning techniques: Performance, retrievability and bias. In *Proceedings of the ACM CIKM* (pp. 2023–2026).

Chen, R.-C., & Lee, C.-J. (2013). An information-theoretic account of static index pruning. In *Proceedings of the ACM SIGIR* (pp. 163–172).

Chen, R.-C., Lee, C.-J., & Croft, W. B. (2015). On divergence measures and static index pruning. In *Proceedings of the ACM SIGIR* (pp. 151–160).

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., & MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the ACM SIGIR* (pp. 659–666).

Cormack, G. V., Smucker, M. D., & Clarke, C. L. A. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval Journal*, *14*(5), 441–465.

Dai, Z., Xiong, C., & Callan, J. (2016). Query-biased partitioning for selective search. In *Proceedings of the ACM CIKM* (pp. 1119–1128).

De Moura, E. S., dos Santos, C. F., Fernandes, D. R., Silva, A. S., Calado, P., & Nascimento, M. A. (2005). Improving web search efficiency via a locality based static pruning method. In *Proceedings of the WWW* (pp. 235–244).

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT* (pp. 4171–4186).

Draws, T., Roy, N., Inel, O., Rieger, A., Hada, R., Yalcin, M. O., Timmermans, B., & Tintarev, N. (2023). Viewpoint diversity in search results. In *Proceedings of the ECIR* (pp. 279–297).

Gao, R., & Shah, C. (2020). Toward creating a fairer ranking in search engine results. *Information Processing & Management (IPM)*, *57*(1), 102138.

Garcia, S. (2007). *Search engine optimisation using past queries* (Ph.D. thesis), RMIT University.

Jeon, M., Kim, S., Hwang, S., He, Y., Elnikety, S., Cox, A. L., & Rixner, S. (2014). Predictive parallelization: taming tail latencies in web search. In *Proceedings of the ACM SIGIR* (pp. 253–262).

Karako, C., & Manggala, P. (2018). Using image fairness representations in diversity-based re-ranking for recommendations. In *Proceedings of ACM UMAP* (pp. 23–28).

Küçükoğlu, E. C. (2019). *Search result diversification for selective search* [Master's thesis], Middle East Technical University.

Lassance, C., Déjean, H., Clinchant, S., & Nicola, T. (2024). Two-step SPLADE: simple, efficient and effective approximation of SPLADE. In *Proceedings of the ECIR*.

Lassance, C., Lupart, S., Déjean, H., Clinchant, S., & Tonellotto, N. (2023). A static pruning study on sparse neural retrievers. In *Proceedings of the ACM SIGIR* (pp. 1771–1775).

Lin, J., & Dyer, C. (2010). *Data-intensive text processing with MapReduce*. Morgan and Claypool Publishers.

Lipani, A. (2019). On biases in information retrieval models and evaluation. *ACM SIGIR Forum*, *52*(2), 172–173.

Liu, X., Bouchoucha, A., Sordoni, A., & Nie, J. (2014). Compact aspect embedding for diversified query expansions. In C. E. Brodley, & P. Stone (Eds.), *Proceedings of AAAI* (pp. 115–121).

Liu, Q., Guo, G., Mao, J., Dou, Z., Wen, J.-R., Jiang, H., Zhang, X., & Cao, Z. (2024). An analysis on matching mechanisms and token pruning for late-interaction models. *ACM Transactions on Information Systems*, *42*(5), 1–28.

Macdonald, C., Tonellotto, N., & MacAvaney, S. (2021). IR from bag-of-words to BERT and beyond through practical experiments. In *Proceedings of the ACM CIKM* (p. 4861).

Mackenzie, J. M., Culpepper, J. S., Blanco, R., Crane, M., Clarke, C. L. A., & Lin, J. (2018). Query driven algorithm selection in early stage retrieval. In *Proceedings of the ACM WSDM* (pp. 396–404).

Mackenzie, J. M., & Moffat, A. (2020). Examining the additivity of top-k query processing innovations. In *Proceedings of ACM CIKM* (pp. 1085–1094).

Mallia, A., Khattab, O., Suel, T., & Tonellotto, N. (2021). Learning passage impacts for inverted indexes. In *Proceedings of the ACM SIGIR* (pp. 1723–1727).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. New York, NY, USA: Cambridge University Press.

Maxwell, D., Azzopardi, L., & Moshfeghi, Y. (2019). The impact of result diversification on search behaviour and performance. *Information Retrieval Journal*, *22*(5), 422–446.

McDonald, G., Macdonald, C., & Ounis, I. (2022). Search results diversification for effective fair ranking in academic search. *Information Retrieval Journal*, *25*(1), 1–26.

McDonald, G., Thonet, T., Ounis, I., Renders, J.-M., & Macdonald, C. (2019). University of Glasgow Terrier Team and Naver Labs Europe at TREC 2019 fair ranking track. In *Proceedings of TREC conference*.

Meng, Z., Shen, H., Huang, H., Liu, W., Wang, J., & Sangaiah, A. K. (2018). Search result diversification on attributed networks via nonnegative matrix factorization. *Information Processing & Management (IPM)*, *54*(6), 1277–1291.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the ICLR*.

Moura, E. S. d., Santos, C. F. d., Araujo, B. D. s. d., Silva, A. S. d., Calado, P., & Nascimento, M. A. (2008). Locality-based pruning methods for web search. *ACM Transactions on Information Systems*, *26*(2), 9:1–9:28.

Nguyen, L. T. (2009). Static index pruning for information retrieval systems: A postingbased approach. In *SIGIR 2009 Workshop on Large-Scale Distributed Information Retrieval* (pp. 25–32).

Ntoulas, A., & Cho, J. (2007). Pruning policies for two-tiered inverted index with correctness guarantee. In *Proceedings of the ACM SIGIR* (pp. 191–198).

Pehlivan, Z., Piwowarski, B., & Gançarski, S. (2013). Diversification based static index pruning-application to temporal collections. arXiv preprint arXiv:1308.4839.

Raman, K., Bennett, P. N., & Collins-Thompson, K. (2014). Understanding intrinsic diversity in web search: Improving whole-session relevance. *ACM Transactions on Information Systems*, *32*(4), 20:1–20:45.

Rijsbergen, C. J. V. (1979). *Information retrieval* (2nd ed.) Newton, MA, USA: Butterworth-Heinemann.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, *33*(4), 294–304.

Rodriguez, J., & Suel, T. (2018). Exploring size-speed trade-offs in static index pruning. In *Proceedings of the IEEE Big Data* (pp. 1093–1100).

Rodrygo, L., Macdonald, C., & Ounis, I. (2015). Search result diversification. *Foundations and Trends in Information Retrieval*, *9*(1), 1–90.

Santos, R. L. T., Macdonald, C., & Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of WWW* (pp. 881–890).

Schelenz, L. (2021). Diversity-aware recommendations for social justice? Exploring user diversity and fairness in recommender systems. In *Proceedings of ACM UMAP* (pp. 404–410).

Skobeltsyn, G., Junqueira, F., Plachouras, V., & Baeza-Yates, R. (2008). ResIn: A combination of results caching and index pruning for high-performance web search engines. In *Proceedings of the ACM SIGIR* (pp. 131–138).

Soner, A., Ricardo, B.-Y., & Barla, C. B. (2020). Pre-indexing pruning strategies. In *Proceedings of the SPIRE* (pp. 177–193).

Souza, T., Demidova, E., Risse, T., Holzmann, H., Gossen, G., & Szymanski, J. (2015). Semantic URL analytics to support efficient annotation of large scale web archives. In *Proceedings of the IKC* (pp. 153–166).

Vishwakarma, S. K., Lakhtaria, K. I., Bhatnagar, D., & Sharma, A. K. (2014). An efficient approach for inverted index pruning based on document relevance. In *Proceedings of the CSNT* (pp. 487–490).

Wang, Q., Dimopoulos, C., & Suel, T. (2016). Fast first-phase candidate generation for cascading rankers. In *Proceedings of the ACM SIGIR* (pp. 295–304).

Wilkie, C., & Azzopardi, L. (2014). Best and fairest: An empirical analysis of retrieval system bias. In *Proceedings of the ECIR* (pp. 13–25).

Yigit-Sert, S., Altingovde, I. S., Macdonald, C., Ounis, I., & Ulusoy, Ö. (2020). Supervised approaches for explicit search result diversification. *Information Processing & Management (IPM)*, *57*(6), Article 102356.

Yin, D., Hu, Y., Tang, J., Daly, T., Jr., Zhou, M., Ouyang, H., Chen, J., Kang, C., Deng, H., Nobata, C., Langlois, J., & Chang, Y. (2016). Ranking relevance in yahoo search. In *Proceedings of the ACM SIGKDD* (pp. 323–332).

Yu, H.-T., Jatowt, A., Blanco, R., Joho, H., Jose, J. M., Chen, L., & Yuan, F. (2018). Revisiting the cluster-based paradigm for implicit search result diversification. *Information Processing & Management (IPM)*, *54*(4), 507–528.

Zhai, C. X., Cohen, W. W., & Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the ACM SIGIR* (pp. 10–17).

Zheng, L., & Cox, I. J. (2009). Entropy-based static index pruning. In *Proceedings of the ECIR* (pp. 713–718).