Search Result Diversification Using Query Aspects as Bottlenecks

Puxuan Yu
University of Massachusetts Amherst
Amherst, MA, USA
pxyu@cs.umass.edu

Razieh Rahimi
University of Massachusetts Amherst
Amherst, MA, USA
rahimi@cs.umass.edu

Zhiqi Huang
University of Massachusetts Amherst
Amherst, MA, USA
zhiqihuang@cs.umass.edu

James Allan
University of Massachusetts Amherst
Amherst, MA, USA
allan@cs.umass.edu

ABSTRACT

We address some of the limitations of coverage-based search result diversification models, which often consist of separate components and rely on external systems for query aspects. To overcome these challenges, we introduce an end-to-end learning framework called DUB. Our approach preserves the intrinsic interpretability of coverage-based methods while enhancing diversification performance. Drawing inspiration from the information bottleneck method, we propose an aspect extractor that generates query aspect embeddings optimized as information bottlenecks for the task of diversified document re-ranking. Experimental results demonstrate that DUB outperforms state-of-the-art diversification models.

CCS CONCEPTS

• **Information systems** → **Information retrieval diversity**; **Query intent**.

KEYWORDS

Search result diversification; Query aspects; Joint ranking and explanation

ACM Reference Format:
Puxuan Yu, Razieh Rahimi, Zhiqi Huang, and James Allan. 2023. Search Result Diversification Using Query Aspects as Bottlenecks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3583780.3615050

1 INTRODUCTION

Search result diversification (SRD) has long been studied to increase the chance of addressing user information needs [80] in response to their often under-specified search queries [46]. As Spärck Jones et al. [83] suggested, search engines should take into account the relevance of a document in relation to the various potential information needs, also known as query aspects or subtopics, that may be associated with a given query.

There are two broad categories of SRD approaches based on their diversification strategy: *coverage-based* and *novelty-based* [80]. Coverage-based approaches focus on measuring how well a given document covers various aspects of the query. In contrast, novelty-based approaches compare retrieved documents with each other to promote novel information. One advantage of coverage-based approaches over novelty-based approaches is the higher degree of model interpretability and transparency they offer to users [62], because the aspects and how documents cover them can be easily understood. Conversely, novelty-based approaches often rely on measurements such as dissimilarity between document embeddings [84, 85, 97, 98], which are difficult for humans to interpret.

While coverage-based approaches are generally acknowledged for their interpretability, they often rely on external systems for query aspect acquisition, such as (proprietary) Google query suggestion [44, 48, 69, 70] or query completion models trained with query logs [62]. The reliance on such external systems presents several drawbacks. Firstly, the availability of these query aspect acquisition systems cannot be assumed at all times due to factors like high training and/or inference costs, as well as restrictions on extensive usage. Secondly, the acquisition of query aspects is not grounded in the actual documents to be re-ranked, but rather relies on query logs and click data used during the training of the systems. Consequently, there is no guarantee that the provided query aspects are relevant to the candidate documents, which could potentially hinder the re-ranking process. Lastly, query aspect acquisition systems cannot be optimized to align with the downstream objective of search result diversification. Even in the case of IntenT5 [62], which can be considered an open-source alternative to proprietary Google query suggestion, query aspects are represented in plain text, preventing the back-propagation of the diversity-oriented loss to the query suggestion model itself. This lack of joint optimization hinders the simultaneous improvement of query aspect acquisition and diversified re-ranking.

In this research, our main objective is to develop an SRD framework that integrates intrinsic interpretability and effectiveness through end-to-end learning. To achieve this goal, we draw inspiration from the Information Bottleneck Method [86], which focuses on producing a summary of information X that is optimized to predict some other relevant information Y. Motivated by this approach, we introduce DUB (short for **D**iversification **U**sing **B**ottlenecks),

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615050

a coverage-based diversification framework. In DUB, we design a differentiable aspect extraction component that can "summarize" relevant information from candidate documents into latent aspect embeddings, which are optimized to enhance the downstream diversified re-ranking task. This component addresses the need for simultaneous optimization of query aspect acquisition and document re-ranking. We propose two implementations for query aspect extraction. The first approach employs multi-head attention [89], where each head learns the representation of a specific query aspect [19]. The second approach involves clustering document segments (passages), where each cluster represents a distinct query aspect.

We introduce a novel pretraining task termed *aspect matching*. This task involves the development of a dataset construction approach along with two pretraining objectives. By integrating aspect matching into DUB, we empower it with the capability to effectively use abundant query-aspect relationships found within Wikipedia. This significantly mitigates the issue of data scarcity that commonly hinders advances in search result diversification.

We perform extensive evaluations of DUB in terms of search result diversification. Notably, DUB is able to outperform approaches which get explicit query aspects from Google query suggestions [44] and large language models (e.g., GPT-3.5 [12]). Specifically, DUB demonstrates a significant improvement of 4.3% in α-nDCG@20 performance for the diversity task in the TREC Web tracks [21–24] when compared to the strongest baseline. Furthermore, extrinsic evaluations using latent aspects represented as unigram language models demonstrate that DUB generates aspects that exhibit higher relevance to labeled relevant documents, rendering them more suitable for the purpose of diversified reranking.

2 RELATED WORK AND BACKGROUND

DUB is a search result diversification framework using explicit extraction of latent query aspects from search results. Our work is thus related to *query aspect acquisition* and *search result diversification*.

2.1 Query Aspect Acquisition

Obtaining query aspects (alternatively referred to as subtopics, facets, or intents) for ambiguous queries has proven beneficial in various information retrieval tasks, including query suggestion [3, 8], search result diversification [80], and asking clarifying questions [4, 102]. Query aspects can be acquired from diverse resources include query logs [7, 49, 72, 91, 103], anchor text [27, 54], knowledge bases [10, 47], the entire corpus [8], top-retrieved results, or a combination thereof [34, 42]. Regarding the extraction of query aspects from search results specifically, Dou et al. [35, 36] propose QDMiner that automatically mines query facets by extracting and grouping frequent lists from free text, HTML tags, and repeated regions within top search results. Kong and Allan [52] introduce a learnable graphical model for query facet extraction.

More recently, Transformer based language models [50, 57, 74] have been used to generate query aspects. IntenT5 [62] is trained on query logs [26] to predict succeeding terms of a given query as its intents. Samarinas et al. [77] propose to generate query facets by prompting large language models using a few examples of query-facets pairs as prompts. Their findings suggest that when evaluated using a dedicated facet generation dataset, this method is not as effective as facet generation models that rely on documents [36, 40]. However, when assessed through manual evaluations conducted by humans, it performs equally well. We present an evaluation of using aspects generated by GPT-3.5 for the purpose of search result diversification in Section 5. Rahimi et al. [75] and Yu et al. [100] design text generation models to generate query aspects *per document*. NMIR [40] and its permutation-invariant version PINMIR [41] employ a language generation model to output multiple query intents given a query and its top-retrieved documents. However, their configurations impose a constraint where the combined length of the query and a cluster of documents (for NMIR) or all candidate documents (for PINMIR) must not exceed the maximum input limit of the generation model, such as 1024 tokens of BART [57]. This limitation becomes impractical when extracting query aspects from a larger collection of lengthier documents.

2.2 Search Result Diversification

Approaches for search result diversification (SRD) can be classified into three categories based on their diversification strategy: coverage-based (estimating document coverage of query aspects), novelty-based (comparing retrieved documents to each other), or a combination of both (also known as hybrid) [58, 69, 70, 79]. Novelty and coverage are related to the notions of extrinsic and intrinsic diversity, respectively, as discussed by Radlinski et al. [71]. Additionally, an alternative categorization that is commonly used in recent literature is the distinction between *implicit* and *explicit* approaches, depending on whether or not the approach uses explicit aspect representations. Explicit approaches are typically considered to be coverage-based or hybrid, as they require explicit query aspects to estimate coverage, while implicit approaches can be thought of as novelty-based. Regarding how documents are scored, SRD methods can also be classified into two main paradigms: *score-and-sort* and *next-document*. In the score-and-sort paradigm [69, 97, 98], candidate documents are scored collectively and subsequently re-ranked. Conversely, the *next-document* paradigm [14, 48, 70, 78, 81, 84, 85] adopts an iterative strategy where at each step a document is greedily selected to maximize a combination of its relevance to the query and its novelty in relation to previously selected documents.

Recent neural network-based approaches for search result diversification tend to adopt implicit methods and do not rely on explicit aspect representations [84, 85, 93, 94, 97, 98, 105]. Instead of directly comparing document representations to compute novelty [14], some of these approaches leverage graph neural networks [51] or document interaction networks [67] for document interactions and representation updates. The updated document representations are then utilized to infer novelty scores [84, 85] or final ranking scores [97, 98]. Some implicit approaches also have the capability to incorporate external knowledge, such as document relation classifiers [84] and entity-related information [85]. On the other hand, explicit and hybrid diversification approaches typically rely on query aspect representations obtained from external sources. Diversified query expansion (DQE) [88] employs multiple external sources, including knowledge bases[10], word embeddings [66], and query logs [59], in addition to the top-retrieved documents [11], to expand the original query. This expansion introduces more diverse terms into the query, resulting in the retrieval of a broader

range of diverse documents. Unsupervised explicit approaches like xQuAD [78], PM2 [29] and HxQuAD/HPM2 [44] use query reformulations or suggestions [44]. Supervised explicit approaches DSSA [48], DESA [69], and GDESA [70] employ similar information as query aspects but can be trained using subtopic-level relevance judgments.

Numerous studies have explored the use of clustering for subtopic mining [90], subtopic retrieval [15, 16] and search result diversification [43, 64, 81] in the context of unsupervised methods. In line with the concept of "query-specific clustering" [43], our proposed framework distinguishes itself by integrating clustering as a fundamental component of an end-to-end learning model.

Note that we do not claim DUB to be more interpretable than other coverage-based approaches. While interpretability is a primary motivation for developing coverage-based SRD models, the evaluation of interpretability involves several factors, such as aspect description readability, aspect diversity, aspect faithfulness, and the trade-off with diversification effectiveness. The examination of interpretability is beyond the scope of our current study, and we leave it to future investigations.

2.3 Background

We briefly introduce Multi-Head Attention and Differentiable K-Means that are required for describing DUB in Section 3.

2.3.1 Multi-Head Attention. Given query, key, and value matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, the attention (Attn) and the multi-head attention (MHA) functions [89] are defined as follows:

$$\text{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}})\boldsymbol{V}, \tag{1}$$

$$\text{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Concat}(\text{head}_1,\cdots,\text{head}_h)\boldsymbol{W}^{\boldsymbol{O}}, \tag{2}$$

$$\text{head}_i = \text{Attn}(\boldsymbol{Q}\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{Q}},\boldsymbol{K}\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{K}},\boldsymbol{V}\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{V}}), \tag{3}$$

where h is the number of attention heads, and d is the embedding dimension. Intuitively, attention calculation is performed in h subspaces of dimension d/h in parallel, and then aggregated back into the main space of dimension d. Projection matrices $\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{Q}}$,$\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{K}}$,$\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{V}}$, and $\boldsymbol{W}^{\boldsymbol{O}}$ are learnable parameters.

2.3.2 Differentiable K-means. DKM [20] is an attention-based clustering layer that enables end-to-end training through backpropagation, optimizing the loss function of the overall task. In our context, end-to-end training offers the advantage of learning representations that are well-suited for aspect extraction and, consequently, diversified re-ranking. DKM achieves differentiability by replacing the hard clustering assignment of K-means [63], where each instance can only belong to one cluster, with an attention-based soft assignment. This allows each instance to have membership in *all* clusters with different attention weights. However, DKM converges to a trivial solution, resulting in the same K clusters, as all clusters comprise the same set of instances (i.e., all instances). Cho et al. [20] apply early stopping by setting a maximum of five clustering iterations to avoid this undesirable convergence behavior. Although a pre-defined number of clustering iterations can work for clustering similar sets of instances, it is not suitable for clustering dynamically changing sets of passages for different queries. A desirable number of clustering iterations can vary from one query to another.

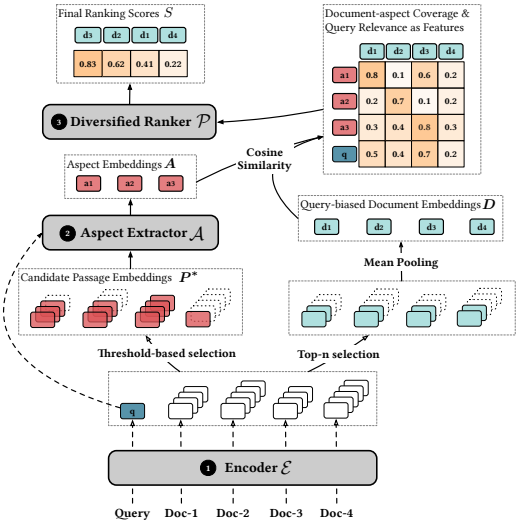

Figure 1: An overview of the DUB framework.

In Section 3.3, we introduce Generalized DKM, which addresses those limitations of DKM when applied to our specific task.

3 METHODOLOGY

3.1 Task Formulation and Framework Overview

Our proposed method is compatible with both sort-and-score and next-document ranking paradigms (Section 2.2). We focus on the **score-and-sort** approach, primarily due to its efficiency. Formally, consider a search query denoted as q and a ranked list of candidate documents represented as R. An SRD model, denoted as $\mathcal{F}$, generates a list of ranking scores, $S = \mathcal{F}(q, R)$. The goal is to obtain a re-ranked list $\pi(R)$ according to S, which is expected to exhibit higher diversity compared to the original ranked list R.

The DUB framework $\mathcal{F}$ in our study consists of three learnable components, as depicted in Figure 1: the text encoder $\mathcal{E}$ (❶), the aspect extractor $\mathcal{A}$ (❷), and the diversified ranker $\mathcal{P}$ (❸). The text encoder $\mathcal{E}$ is responsible for obtaining the query embedding $\boldsymbol{q}$ and passage embeddings $\boldsymbol{P}$. These embeddings are further utilized to select candidate passage embeddings $\boldsymbol{P}^*$ and to build query-biased document embeddings $\boldsymbol{D}$. Subsequently, the aspect extractor $\mathcal{A}$ leverages the candidate passages embeddings $\boldsymbol{P}^*$ and query embeddings $\boldsymbol{q}$ to produce aspect embeddings $\boldsymbol{A}$, which serve as information bottlenecks for the diversification task. Lastly, the diversified ranker $\mathcal{P}$ takes the aspect embeddings $\boldsymbol{A}$ and the query-biased document embeddings $\boldsymbol{D}$, calculates document-aspect coverage as document features, and outputs scalar ranking scores S. We now introduce each component in detail.

3.2 Text Encoder

Most recent works on search result diversification [37, 48, 69, 70, 84, 85, 95, 97, 98] use unsupervised Doc2Vec embeddings [56] to represent queries, aspects, and documents. Instead, inspired by the effective use of contextualized representations across various NLP and IR tasks, we use a shared Transformer language model as the query and document encoder, which allows DUB to be optimized end-to-end with respect to *input texts*.

We segment long documents into overlapping passages and perform passage-level encoding. This approach is motivated by three key factors: (1) the length of documents often exceeds the input limit length of the encoder, such as the 512-token limit of BERT [50]; (2) it may be that only a small portion of a long document is relevant to the query [87]; and (3) a long document may cover multiple query aspects, leading to potential information loss if a single embedding represents the entire document [61].

We take the mean of token embeddings from the last encoder layer to obtain the query embedding ($\boldsymbol{q}$) and passage embeddings ($\boldsymbol{P}$). After encoding all passages from all documents to be re-ranked, we filter out less related passages and derive two types of representations – candidate passage embeddings for aspect extraction and query-biased document embeddings for document scoring. For candidate passage embeddings, passages whose similarity to the query is greater than a preset threshold θ are selected. The embeddings of the retained passages from *all* candidate documents are denoted as $\boldsymbol{P^*} = \{\boldsymbol{p} | \cos(\boldsymbol{q}, \boldsymbol{p}) \geq \theta, \boldsymbol{p} \in \boldsymbol{P}\}$ and are used for aspect extraction. For query-biased document embedding from its passage representations, we select the top-n passages from the document that are most similar to the query embedding and average them to construct a query-biased embedding for the document, denoted by $\boldsymbol{d}$. Query-biased document embeddings for all candidate documents with respect to a query is denoted as $\boldsymbol{D}$.

Both selection methods aim to filter out irrelevant contexts from the model. The adoption of two different selection criteria is mainly for handling irrelevant documents in the candidate set, some of which may lack suitable passages for building the query-biased document embedding $\boldsymbol{D}$ if threshold-based selection alone is used.

3.3 Aspect Extractor

After obtaining the query embedding $\boldsymbol{q}$ and candidate passage embeddings $\boldsymbol{P^*}$, the aspect extractor $\mathcal{A}$ is employed to generate a fixed number (K) of aspect embeddings $\boldsymbol{A} = \mathcal{A}(\boldsymbol{q}, \boldsymbol{P^*})$, where $\boldsymbol{A} \in \mathbb{R}^{K \times d}$. Each aspect embedding aims to capture one specific aspect of the query-relevant information that is covered by the retrieved documents. We investigate two different approaches for extracting query aspects.

Aspect Extractor Using Multi-Head Attention. The first design of the DUB aspect extractor is based on multi-head attention (MHA) to construct query aspects from similar passages. However, we introduce a modification to the original MHA [89] and its application in aspect-based dense retrieval [53]. Specifically, we consider the output of each attention head as the latent representation of a query aspect, akin to the intent modeling approach proposed by Chen et al. [19]. In this aspect extractor, the outputs of $h = K$ attention heads are not combined into a single output, as expressed in Eq. 2, but are preserved separately as K aspect embeddings. Hence, the formal implementation of $\mathcal{A}$ can be defined as follows:

$$\boldsymbol{A} = \mathcal{A}(\boldsymbol{q}, \boldsymbol{P^*}) = \text{MHA}(\boldsymbol{q}, \boldsymbol{P^*}, \boldsymbol{P^*}) = \{\text{head}_1, \cdots, \text{head}_K\}, \quad (4)$$

$$\boldsymbol{a_i} = \text{head}_i = \text{Attn}(\boldsymbol{q}\boldsymbol{W_i^Q}, \boldsymbol{P^*}\boldsymbol{W_i^K}, \boldsymbol{P^*}\boldsymbol{W_i^V}), \quad (5)$$

In this formulation, the query embedding $\boldsymbol{q}$ serves as the query matrix of the self-attention module, while the candidate passage embeddings $\boldsymbol{P^*}$ are treated as the key and value matrices. Notably, the input projection matrices $\boldsymbol{W_i^Q}$, $\boldsymbol{W_i^K}$, and $\boldsymbol{W_i^V}$ in DUB's aspect extractor have dimensions of $\mathbb{R}^{d \times d}$, differing from the matrices in Eq. 3 with dimensions of $\mathbb{R}^{d \times d/h}$. This modification ensures that the output of each head in the aspect extractor has a dimension of d to represent one query aspect.

Algorithm 1: GDKM algorithm

Input: Passage embeddings $\boldsymbol{P^*}$, minimum moving distance ϵ, number of clusters K, temperature τ, degree of freedom ν, and mask attention value ι.
Output: Cluster assignments $\tilde{\boldsymbol{\alpha}}$ and centroids $\tilde{\boldsymbol{\mu}}$

Function GDKM($\boldsymbol{P^*}, \epsilon, K, \tau, \nu, \iota$):
  $\boldsymbol{\mu} \leftarrow K$-means++$(\boldsymbol{P^*}, K)$ // Initialization; $|\boldsymbol{\mu}| == K$
  while *True* **do**
    $\boldsymbol{\delta} \leftarrow \{\delta_{ij} = \cos(\boldsymbol{p_i}, \boldsymbol{\mu_j})\},\ 1 \leq i \leq |\boldsymbol{P^*}|, 1 \leq j \leq |\boldsymbol{\mu}|$
    $\boldsymbol{\alpha} \leftarrow \{\alpha_{ij} = \frac{\exp(\delta_{ij}/\tau)}{\sum_j \exp(\delta_{ij}/\tau)}\},\ 1 \leq i \leq |\boldsymbol{P^*}|, 1 \leq j \leq |\boldsymbol{\mu}|$
    for $i = 1, 2, \cdots, |\boldsymbol{P^*}|$ **do**
      $t \leftarrow$ sort-desc$(\alpha[i])[\nu]$ // ν−largest in $\alpha[i]$
      for $j = 1, 2, \cdots, |\boldsymbol{\mu}|$ **do**
        if $\alpha_{ij} \geq t$ **then**
          $\tilde{\alpha_{ij}} \leftarrow \alpha_{ij}$
        else
          $\tilde{\alpha_{ij}} \leftarrow \iota$
        end
      end
    end
    $\tilde{\boldsymbol{\mu}} \leftarrow \{\tilde{\mu_j} = \frac{\sum_i \tilde{\alpha_{ij}} \boldsymbol{p_i}}{\sum_i \tilde{\alpha_{ij}}}\},\ 1 \leq i \leq |\boldsymbol{P^*}|, 1 \leq j \leq |\boldsymbol{\mu}|$
    if $\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \leq \epsilon$ **then**
      $\tilde{\boldsymbol{\alpha}} \leftarrow \{\tilde{\alpha_{ij}}\},\ 1 \leq i \leq |\boldsymbol{P^*}|, 1 \leq j \leq |\boldsymbol{\mu}|$
      return $\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\mu}}$ // converge and exit
    else
      $\boldsymbol{\mu} \leftarrow \tilde{\boldsymbol{\mu}}$ // go to the next iteration
    end
  end

Aspect Extractor using GDKM-based Clustering. An alternative intuition for query aspect extraction is that passages covering the same query aspect have more similar embeddings compared to those covering different aspects [84]. Based on this intuition, we perform clustering on candidate passage embeddings $\boldsymbol{P^*}$ to derive representations of aspects. In the clustering-based aspect extractor component, we directly incorporate passage interactions. This is in contrast to the MHA-based approach, where passage interactions are indirectly captured through their similarity to the query embedding. Clustering-based aspect extraction involves two steps.

Step 1: Clustering passages with Generalized Differentiable K-means. The clustering component for aspect extraction should be differentiable so that the encoder ($\mathcal{E}$) parameters can be optimized through gradients from the loss function. To address the limitations of DKM [20] for query-specific passage clustering (Section 2.3.2), we propose GDKM, a generalization of DKM, by limiting the number of clusters that each instance (passage) can be assigned to. A passage can belong to multiple, but *not all*, clusters in a probabilistic way. For this purpose, we introduce a hyper-parameter *degree of freedom*, denoted by ν, that represents the maximum number of clusters an instance can be assigned to. GDKM is generalized because it reduces to the original K-means algorithm [63] by setting ν=1, and on the other hand, to DKM by setting ν to K. Setting ν

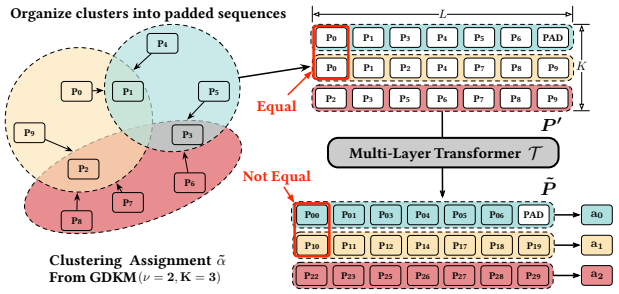

Figure 2: Aspect extraction using GDKM clustering (K=3, ν=2). Each cluster contains passages that are inside its border and passages that point to its border. E.g., both p_0 and p_4 are part of the yellow cluster, with the former having a higher probability, whereas p_3 does not belong to this cluster.

to an integer between 1 and K, GDKM allows modeling passages covering multiple aspects of the query without having the convergence issue of DKM (Section 2.3.2). The pseudocode of GDKM is presented in Algorithm 1. Highlighted lines indicate the extension to DKM. For each passage embedding, the clustering layer first estimates the probability of its membership in each latent cluster, resulting in an attention matrix denoted by $\boldsymbol{\alpha}$ in line 5. Instead of using this attention matrix to compute new centroids as in DKM, we only keep the highest ν attention weights per passage (line 10) and mask the rest with a small constant value ι (line 12). New cluster centroids are then calculated based on the masked attention matrix $\tilde{\boldsymbol{\alpha}}$. After convergence, GDKM outputs cluster assignments $\tilde{\boldsymbol{\alpha}}$ and cluster centroids $\tilde{\boldsymbol{\mu}}$ from the final iteration.

Step 2: Generating aspect embeddings from clusters. A passage belonging to at most ν clusters means that the passage can be represented in at most ν ways. DUB thus represents each passage with ν embeddings, each corresponding to one of the query aspects it might cover. For this, DUB includes a multi-layer Transformer, denoted by $\mathcal{T}$ (different from text encoder $\mathcal{E}$), to learn aspect-specific representations of passages $\tilde{\boldsymbol{P}}$ from initial embeddings $\boldsymbol{P^*}$ and their cluster assignments $\tilde{\boldsymbol{\alpha}}$, i.e., $\tilde{\boldsymbol{P}}, \boldsymbol{\alpha'} = \mathcal{T}(\boldsymbol{P^*}, \tilde{\boldsymbol{\alpha}})$. This process is illustrated in Figure 2. We use clustering assignment $\tilde{\boldsymbol{\alpha}}$ to organize $\boldsymbol{P^*}$ into K sequences of passage embeddings $\boldsymbol{P'}$, in which a passage embedding from $\boldsymbol{P^*}$ is copied ν times. We pad the shorter sequence(s) with zero vectors for batching, and denote the padded sequence length as L. We reformat $\tilde{\boldsymbol{\alpha}}$ into $\boldsymbol{\alpha'}$ by removing entries with masked value ι (as they are not represented in $\boldsymbol{P'}$) and adding entries for padded embeddings with ι, resulting in $\boldsymbol{\alpha'} \in \mathbb{R}^{K \times L}$. Then, the multi-layer Transformer $\mathcal{T}$ updates passage embeddings with in-sequence (in-cluster) self-attention. Each passage embedding is thus updated based on other passages in the same cluster, covering the same query aspect. The obtained aspect-specific passage embeddings $\tilde{\boldsymbol{P}}$ are less ambiguous compared to $\boldsymbol{P^*}$ and can lead to more accurate aspect embeddings. Finally, aspect embeddings $\boldsymbol{A}$ are obtained by averaging aspect-specific embeddings $\tilde{\boldsymbol{P}} \in \mathbb{R}^{K \times L \times d}$ weighted by their degree of membership $\boldsymbol{\alpha'} \in \mathbb{R}^{K \times L}$, i.e., $\boldsymbol{A} = \text{Softmax}(\boldsymbol{\alpha'})\tilde{\boldsymbol{P}}^\intercal$, where $\tilde{\boldsymbol{P}}^\intercal$ represents transposing the first two dimensions of $\tilde{\boldsymbol{P}}$. In Section 5.7, we demonstrate the efficacy of $\mathcal{T}$ in contrast to the direct use of clustering centroids $\tilde{\boldsymbol{\mu}}$ from GDKM as aspect representations.

3.4 Diversified Ranker

Explicitly modeling the possible query aspects covered by candidate documents allows us to estimate their relevance to those aspects and provide a diversified ranking of retrieved results, similar to explicit models [28, 30, 44, 48, 58, 69, 70, 78, 79, 82]. For this purpose, we first estimate document-aspect coverage by simply taking the cosine similarity of query-biased document embeddings $\boldsymbol{D}$ and aspect embeddings $\boldsymbol{A}$. In addition, we also use the cosine similarity between document embeddings $\boldsymbol{D}$ and the original query embedding $\boldsymbol{q}$ to represent documents' overall relevance to the query. Thus, a document is represented with $K+1$ features (K for aspects and 1 for query). In the *score-and-sort* re-ranking paradigm, we adopt a multi-layer feed-forward neural network $\mathcal{P}$ with batch normalization, following the approach of previous studies [97, 98], as the diversified ranker. The formal definition of the diversified ranker $\mathcal{P}$ can then be represented as follows:

$$S = \mathcal{P}(\text{Concat}(\cos(\boldsymbol{D}, \boldsymbol{A}); \cos(\boldsymbol{D}, \boldsymbol{q}))), \tag{6}$$

4 TRAINING

DUB has three learnable components $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$. It can be trained end-to-end with search result diversification (SRD) dataset such that the text encoder $\mathcal{E}$ and the aspect extractor $\mathcal{A}$ can be optimized towards the final goal of diversified re-ranking. However, a data scarcity challenge arises due to the limited number of queries available for training in the largest publicly accessible SRD dataset, TREC Web Tracks [21–24], which contain less than 200 queries.

To address this challenge, we propose the use of **optional pretraining** and **end-to-end SRD training**. Specifically, the parameter-heavy components $\{\mathcal{E}, \mathcal{A}\}$ of the model $\mathcal{F}$ can be first trained on a related task with a larger amount of weak-supervision data. Subsequently, the entire model $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$ is trained end-to-end for SRD. We emphasize that the pretraining step is *optional*. As shown in later experiments, training DUB on a larger SRD dataset (automatically generated, more than 8K queries) is viable without pretraining.

4.1 Optional Pretraining Using Aspect Matching

Drawing inspiration from previous studies that utilized the large amount of entity-heading-section structured data from Wikipedia to replicate the query-aspect-passage structure in information retrieval tasks [33, 75, 100], we use an automatic annotation approach to generate weak training data. The details of this approach are explained in Section 5.1. Each training sample comprises a query q, K reference aspects denoted as A_r, and passages that are relevant to both the query and a specific reference aspect. The embeddings of the reference aspects are represented by $\boldsymbol{A_r} = \mathcal{E}(A_r)$, where $\boldsymbol{A_r} \in \mathbb{R}^{K \times d}$. In order to align the predicted aspects ($\boldsymbol{A}$) with the reference aspects ($\boldsymbol{A_r}$) in the embedding space, it is necessary to adopt an objective that quantifies the difference between them. However, due to the lack of clear alignments between the two sets of aspect embeddings, minimizing their pairwise differences becomes a challenging task. We introduce two solutions.

Optimal-Transport Based Objective. We cast the alignment of predicted and reference aspect embeddings as an instance of the optimal transport (OT) problem and solve it with an existing OT solver. This is inspired by works on aligning token embeddings

from different languages [5, 45, 65]. In this OT formulation, a cost matrix $\boldsymbol{M}$ needs to be defined, where m_{ij} reflects the distance between the i-th predicted aspect embedding $\boldsymbol{a_i}$ and the j-th reference aspect embedding $\boldsymbol{a_j^r}$. We choose as this cost the cosine distance: $m_{ij} = 1 - \cos(\boldsymbol{a_i}, \boldsymbol{a_j^r})$. The total *transportation cost* from $\boldsymbol{A}$ to $\boldsymbol{A_r}$ is defined as $\boldsymbol{\gamma} \cdot \boldsymbol{M}$, where, $\boldsymbol{\gamma} \in \mathbb{R}^{K \times K}$ is called the *transportation matrix*. The $\boldsymbol{\gamma}_*$ that minimizes the transportation cost is called the optimal transport matrix, which intuitively represents the optimal alignment between $\boldsymbol{A}$ and $\boldsymbol{A_r}$. To overcome the intractability of the linear programming solutions for finding $\boldsymbol{\gamma}_*$, we use the IPOT algorithm [96] to compute this OT matrix. Finally, we define the objective for aspect matching as the optimal transportation cost:

$$\mathcal{L}_{\text{OT}}(\boldsymbol{A}, \boldsymbol{A_r}) = \boldsymbol{\gamma}_* \cdot \boldsymbol{M}, \tag{7}$$

Teacher-Forcing Based Objective. The clustering-based aspect extractor can be trained with an alternate objective. Note that this aspect extractor comprises an GDKM clustering layer (without trainable parameters) and a multi-layer Transformer $\mathcal{T}$ (with trainable parameters). Only GDKM causes nondeterministic matching between predicted and reference aspect embeddings. Therefore, during the pretraining step, we can skip GDKM and directly give true "clustering" assignment $\hat{\boldsymbol{\alpha}}$ as input of $\mathcal{T}$. This is similar to teacher-forcing training [92] used for training sequence generation models [73]. This approach provides training stability by eliminating potential misalignment of embedding sets from the OT solver. The loss function of this training method is defined based on the cosine distance of matching embeddings as:

$$\mathcal{L}_{\text{TF}}(\boldsymbol{A}, \boldsymbol{A_r}) = \sum_{i=1}^{K} \left(1 - \cos(\boldsymbol{a}, \boldsymbol{a_i^r})\right), \tag{8}$$

We observe slightly better performance of DUB-GDKM using this loss for pretraining compared with the OT-based loss (Eq. 7).

4.2 End-to-end SRD Training

As described in Section 3.4, a multi-layer feed-forward neural network $\mathcal{P}$ predicts ranking scores S for documents in R based on $(K+1)$-dimensional features of aspect coverage and query similarity. We use the α-DCG loss [97] to optimize the entire DUB model $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$ using training data with aspect-level relevance judgements. Due to space limitations, the formal definition of the α-DCG loss is not repeated here.

5 EXPERIMENTS

5.1 Datasets

We introduce two evaluation datasets and one pretraining dataset.

TREC-Web. The TREC Web track datasets from 2009 to 2012 [21–24] are used for evaluating search result diversification [37, 48, 69, 70, 84, 85, 93, 97]. The combined dataset, referred to as **TREC-Web**, consists of 198 topics after excluding two topics without subtopic judgments, and documents from the ClueWeb'09-Category B collection [13]. Five-fold cross-validation is conducted using the same data folds as previous works [48, 70].

MIMICS-Div. The **MIMICS-Div** dataset is constructed based on the "ClickExplore" version of the MIMICS datasets [103], which originate from real search queries obtained from the Bing query logs. Initially developed for search clarification [102, 103], MIMICS has been adapted for intent representation learning [40] and search result explanation [100]. We repurpose it to evaluate search result diversification, particularly to simulate a scenario with (relatively) abundant real queries and investigate the effects of optional pretraining. Specifically, each candidate answer for a query-clarification pair is considered as a query aspect. The MIMICS datasets do not provide full document contents or relevance labels. Following prior studies [40, 100], we consider the concatenation of a document's heading and snippet as its content, and a document is deemed relevant to a query aspect if it contains all the aspect terms. MIMICS-Div contains 8,166 queries, with an average of 3.17 aspects per query. It is important to note that the relevance assessments in MIMICS-Div are inferred rather than manually verified.

Aspect pretraining data. To pretrain the text encoder $\mathcal{E}$ and aspect extractor $\mathcal{A}$ (Section 4.1), we construct a weakly supervised dataset from Wikipedia, referred to as **Wiki**. This is mostly in line with previous uses of Wikipedia for related tasks [33, 75, 99–101]. For each Wikipedia article, its title is used as the query, its section headings are treated as query aspects, and its sections (excluding the introductory paragraph) are divided into multiple passages that are relevant to the corresponding aspect. We select Wiki articles with 8 or more sections from the pre-processed data released by Yu et al. [100] and obtain 203,751 training samples. During pre-training DUB on Wiki, we randomly sample K aspects per query (which differs depending on evaluation dataset, Section 5.5). The reference aspect embeddings $\boldsymbol{A_r}$ are obtained by encoding the concatenation of the title and the section heading with the same text encoder $\mathcal{E}$ used for the queries and passages.

5.2 Competing Models

We categorize competing models into three groups.

(1) Explicit models by extracting aspects from search results. We develop two baselines based on topic modeling [17]. They align with the setting of DUB (i.e., explicit diversification based on finding aspects from candidate documents) but employ unsupervised aspect extraction and diversification components. Given documents, topic models extract latent topics (query aspects in our task) and represent each topic as a probability distribution over vocabulary (a unigram language model). The probability of a document covering a query aspect can then be approximated by $\Pr(d_i|a_j) = \prod \Pr(v|a_j, v \in d_i)$. This probability is computed for each document and each aspect. We then use xQuAD [81], an unsupervised explicit diversification algorithm, to re-rank candidate documents. Specifically in xQuAD, we use uniform aspect importance distributions, and tune parameter λ (for balancing relevance and diversity) with cross-validation. We compute aspect models using statistical model LDA [9] and neural topic model BERTopic [39], resulting in LDA-xQuAD and BERTopic-xQuAD.

(2) Neural implicit models. We focus on recent implicit SRD models, namely Graph4DIV [84], DALETOR [97], and KEDIV [85]. Like DUB, these models do not rely on provided query aspects. Graph4DIV and KEDIV are also knowledge-enhanced, similar to how DUB can be pretrained on Wikipedia. As we cannot reproduce KEDIV, we directly cite their reported numbers.

(3) Explicit models by using aspects from external sources. xQuAD [78], PM2 [29] and HxQuAD/HPM2 [44] are unsupervised

explicit SRD approaches relying on query aspects from external sources. DSSA [48], DESA [69], and GDESA [70] also use Google query suggestions, but they are supervised neural models that can be optimized with aspect-level judgements. In addition, we consider large language models (LLMs) as a source of query aspects. Specifically, for TREC-Web queries, we use OpenAI GPT-3.5 [12] to generate query aspects using instruction "list 10 potential aspects about the search query: {query}". Each generated aspect contains a short aspect name followed by a descriptive sentence. We only keep the aspect name to be consistent with GQS. We then use xQuAD to re-rank candidate documents using GPT aspects.

5.3 Experimental Settings on TREC-Web

Comparing SRD approaches on TREC-Web presents challenges. We provide clarity on the experimental settings.

Candidate documents can be categorized into two types on TREC-Web. *Original* involves reranking documents specifically labeled for a given query [95, 97, 98], while *Lemur* refers to reranking the top-50 documents retrieved by the Lemur Indri system [44, 48, 69, 70, 84]. We follow most approaches in the literature and use the *Lemur* candidate set as it represents a plausible web search setting.

Text embeddings used in DALETOR, Graph4DIV, KEDIV, DSSA, DESA, and GDESA are static Doc2Vec embeddings. We also conduct experiments by substituting these embeddings with Transformer-based embeddings used in DUB, and report corresponding results (except for KEDIV, since we cannot reproduce their model). Specifically, for document embeddings, we use the query-biased document embeddings $\boldsymbol{D}$ derived from $\mathcal{E}$. We find the optimal hyper-parameter n for each model individually using cross-validation. For DSSA, DESA, and GDESA, which also require subtopic embeddings, we use $\mathcal{E}$ to encode the first-level query suggestions released by Hu et al. [44]. We also explored the use of pseudo document embeddings as subtopic embeddings, as suggested by Jiang et al. [48], but found it to yield inferior performance.

Relevance features play a crucial role in Graph4DIV, KEDIV, DSSA, DESA, and GDESA. These include 18 query-level static handcrafted document features, including BM25, TF-IDF, PageRank scores, and number of links [48]. The latter three models also incorporate aspect-level features, resulting in $18\times(K+1)$ features in total (assuming K aspects per query). To study the effect of these relevance features, we introduce a variant of DUB, denoted as DUB-*-RF, that takes query-level features (in addition to features introduced in Section 3.4) in the diversified ranker $\mathcal{P}$. Note that we do **not** use aspect-level relevance features in DUB-RF, as we do not acquire aspects from anywhere but the candidate documents.

5.4 Evaluation Metrics

We adopt the official TREC evaluation methodology for the diversity task. On TREC-Web, we report the following evaluation metrics with a cut-off set to 20, as done in previous studies [48, 69, 70, 84]: α-nDCG [25], ERR-IA [18], NRBP [6], Pre-IA [2], and S-rec [104]. We set the parameter α to 0.5, the default setting in the official TREC evaluation program. On MIMICS-Div, we report α-nDCG@{5,10} and ERR-IA@{5,10} since the candidate set contains at most 10 documents. For conducting statistical significance tests, we employ the t-test with Bonferroni correction at the 95% confidence level.

Table 1: Search result diversification on TREC-Web.

#	Metric	α-nDCG	ERR-IA	NRBP	Pre-IA	S-rec
	Term-level Representations					
1	(1) LDA-xQuAD	0.335	0.224	0.183	0.127	0.608
2	(3) GPT-xQuAD	0.400	0.307	0.271	0.159	0.616
3	(3) GQS-xQuAD	0.413	0.317	0.284	0.161	0.622
4	(3) GQS-PM2	0.411	0.306	0.267	0.169	0.643
5	(3) GQS-HxQuAD	0.421	0.326	0.294	0.158	0.629
6	(3) GQS-HPM2	0.420	0.317	0.279	0.172	0.645
	Doc2Vec as Text Encoder					
7	(2) DALETOR	0.399	0.308	0.270	0.149	0.608
8	(2) Graph4DIV	0.468	0.370	0.338	0.186	0.666
9	(2) KEDIV (from [85])	0.485	0.390	0.362	-	0.671
10	(3) DSSA	0.452	0.350	0.318	0.184	0.645
11	(3) DESA	0.464	0.363	0.332	0.184	0.653
12	(3) GDESA	0.469	0.369	0.337	0.185	0.662
	SBERT as Text Encoder					
13	(1) BERTopic-xQuAD	0.330	0.232	0.199	0.140	0.555
14	(2) DALETOR	0.411	0.317	0.278	0.151	0.614
15	(2) Graph4DIV	0.475	0.375	0.343	0.187	0.669
16	(3) DSSA	0.461	0.357	0.324	0.185	0.649
17	(3) DESA	0.473	0.370	0.338	0.185	0.657
18	(3) GDESA	0.478	0.376	0.344	0.186	0.666
19	DUB-MHA	$0.490^{\dagger}$	$0.386^{\dagger}$	$0.358^{\dagger}$	0.188	0.672
20	DUB-MHA-RF	$0.497^{\dagger}$	$0.391^{\dagger}$	$0.363^{\dagger}$	0.188	0.674
21	DUB-GDKM	$0.502^{\dagger}$	$0.393^{\dagger}$	$0.368^{\dagger}$	0.189	$0.677^{\dagger}$
22	DUB-GDKM-RF	$\mathbf{0.508}^{\dagger}$	$\mathbf{0.399}^{\dagger}$	$\mathbf{0.374}^{\dagger}$	**0.190**	$\mathbf{0.680}^{\dagger}$
	Ablations					
23	DUB-MHA-RF (nP)	0.473	0.372	0.336	0.185	0.663
24	DUB-GDKM-RF (nP, ν=2)	0.461	0.360	0.320	0.184	0.658
25	DUB-GDKM-RF (ν=1)	$0.493^{\dagger}$	$0.387^{\dagger}$	$0.358^{\dagger}$	0.188	0.673
26	DUB-GDKM-RF (ν=3)	$0.506^{\dagger}$	$0.397^{\dagger}$	$0.371^{\dagger}$	**0.190**	$0.679^{\dagger}$
27	DUB-GDKM-RF (ν=8)	0.437	0.343	0.293	0.181	0.647

Statistical improvements that are significant over all baseline models (expect KEDIV, as metrics per query are unknown) are indicated with a † symbol in the result tables.

5.5 Model Specifications of DUB

To initialize the text encoder $\mathcal{E}$, we use the "all-mpnet-base-v2" Sentence-BERT (SBERT) [76]. Each ClueWeb document is segmented into overlapping passages of 96 tokens, with a stride of 32 tokens. We use $\theta = 0.6$ to select candidate passage embeddings and set $n = 20$ for building query-biased document embeddings. For DUB-GDKM, we employ a randomly initialized two-layer Transformer as $\mathcal{T}$. The specific model settings for (pre)training depend on the evaluation dataset. On TREC-Web, the number of aspects per query K (also the number of heads h for DUB-MHA) is set to 8, which corresponds to the maximum number of subtopics per query in TREC-Web. On MIMICS-Div, K is set to 3. The default parameter values for GDKM (Algorithm 1) are: $\epsilon = 10^{-3}$, $\iota = 10^{-6}$, $\nu = 2$, and $\tau = 1$. The diversified ranker $\mathcal{P}$ is implemented as a three-layer MLP with hidden sizes of 256, 64 and 8. The input size of $\mathcal{P}$ is $K+1$ for DUB and $K+19$ for DUB-*-RF. DUB-MHA is pretrained with optimal transport objective (Eq. 7) and DUB-GDKM is pretrained with teacher forcing objective (Eq. 8).

Training hyper-parameters. For optional pretraining on Wiki, we conduct training for 1 epoch with a batch size of 8 and a learning rate of 10^{-4} using AdamW [60] as the optimizer. During end-to-end SRD training, DUB is trained for 100 epochs using a batch size of 4 and a learning rate of 10^{-4}.

5.6 Diversification Performance

The performance of the compared approaches on TREC-Web is presented in Table 1, where the results are categorized into three groups

Table 2: Search result diversification on MIMICS-Div.

Metric	α-nDCG@5	α-nDCG@10	ERR-IA@5	ERR-IA@10
Graph4DIV	0.614	0.701	0.420	0.459
DALETOR	0.678	0.759	0.484	0.512
DUB-MHA	0.699†	0.790†	0.512†	0.543†
DUB-GDKM	**0.705**†	**0.797**†	**0.514**†	**0.549**†
Ablations				
DUB-MHA (nP)	0.687†	0.780†	0.503†	0.532†
DUB-GDKM (nP)	0.682†	0.773†	0.498†	0.524†

based on the representation of queries and documents. Baselines are also categorized according to their characteristics (Section 5.2). Remarkably, DUB-GDKM-RF outperforms all three groups of baselines, providing strong evidence of its effectiveness. The improvements are statistically significant (marked by †) across all metrics except Pre-IA. We also discuss the following observations.

Effect of contextualized representations. To facilitate a direct comparison, we conduct experiments by replacing the 100-dimensional Doc2Vec embeddings with 768-dimensional SBERT embeddings in both supervised implicit and explicit models (Section 5.3). Notably, we observed improvements in the diversification effectiveness across all models we investigated (rows #7-8,10-12 vs. #14-18 in Table 1). This highlights the effectiveness of contextualized representations offered by SBERT. We also note a significant performance advantage of DUB over the strong baselines employing the same SBERT encoder (#19-22 vs. #13-18). This observation suggests that the superior performance of DUB cannot be solely attributed to the improved effectiveness of text encoder.

Effect of relevance features. To fairly compare with Graph4DIV, KEDIV, DSSA, DESA and GDESA which use query-level (and aspect-level, if applicable) relevance features for modeling relevance, we implement variants of DUB (marked with -RF) that integrate those query-level features into document features in the diversified ranker $\mathcal{P}$ (Section 5.3). We first observe that DUB without resorting to relevance features (#19,21) already achieve significant improvement over baselines using relevance features (#15-18). We also observe that using these features in DUB-RF leads to further improvement (#19 vs. #20, #21 vs. #22). This finding demonstrates that these relevance features continue to be effective in providing crucial relevance signals that cannot be captured solely through query-document matching in the embedding space.

Importance of supervised aspect extraction. Results in Table 1 show that all explicit baselines using aspects from external sources (marked with category (3)) outperform those explicit baselines extracting aspects from top-retrieved documents using topic modeling (marked with category (1)). Despite this trend in existing works, our proposed DUB that extracts aspects from top-retrieved documents surpasses the performance of supervised explicit models (DSSA, DESA, and GDESA) using Google query suggestions as aspects (#19-22 vs. #16-18). This finding demonstrates that with an appropriate aspect extractor component, obtaining query aspects from top-retrieved documents is more effective for search result diversification compared to relying on external systems to provide aspects solely based on the query.

Utility of pretraining. As reported in Table 1, ablated variants of DUB without pretraining on Wiki (marked by "nP") exhibit inferior performance on TREC-Web (#20,22 vs. #23,24). The decrease in effectiveness is more pronounced in the GDKM-based model compared to the MHA-based model. This observation can be attributed to the GDKM-based model having a larger number of parameters, thus requiring more training data. For a comprehensive analysis of the role of pretraining, we study the performance of our models and applicable baselines on the MIMICS-Div dataset, that in contrast to TREC-Web, contains numerous training queries.

Diversification on MIMICS-Div. Considering the superior performance of SBERT representations on TREC-Web, we exclusively present approaches using the SBERT encoder on MIMICS-Div in Table 2. Due to the high costs for acquiring query suggestions for over 8,000 queries, we exclude explicit models from this experiment. We observe that DUB significantly outperforms strong baselines, even without pretraining on Wiki. This finding suggests that pretraining on Wiki is not essential for DUB if a sufficient number of search result diversification training queries are available. Furthermore, the additional improvement observed with pretraining on Wiki highlights the effectiveness of this pretraining approach.

5.7 Clustering-based Aspect Extractor

We discuss two key design choices of the clustering-based aspect extractor, using experiments conducted on TREC-Web.

Parameter ν of GDKM (Section 3.3) is the maximum number of clusters that a passage can be assigned to. Table 1 reports the performance of DUB-GDKM when ν is set to $\{1, 2, 3, 8\}$. We observe that setting ν to 2 or 3 yields the best performance for DUB-GDKM. The obtained results demonstrate the effectiveness of the proposed GDKM clustering for aspect extraction compared to the original K-means [63] (ν=1) and differentiable K-means [20] (ν=8).

Multi-layer Transformer $\mathcal{T}$ in DUB-GDKM**.** We examine an ablated version of DUB-GDKM that removes the Transformer layers $\mathcal{T}$ responsible for building aspect-specific passage embeddings, referred to as DUB-GDKM−$\mathcal{T}$. It directly uses clustering centroids $\tilde{\boldsymbol{\mu}}$ from the last GDKM iteration as aspect embeddings $\boldsymbol{A}$. To control the level of noise input into the aspect extractor $\mathcal{A}$, we experiment with different values of θ={0.5, 0.6}. Additionally, we conduct an *oracle* experiment where only passages from labeled relevant documents are fed into $\mathcal{A}$. The results of DUB and DUB-GDKM−$\mathcal{T}$ are presented in Table 3. We observe that the two models perform comparably when the aspect extractor receives input solely from passages of documents labeled as relevant. However, when re-ranking the results of a first-stage retriever where a threshold θ is used to select input passages, the performance of both models significantly deteriorates. Furthermore, the performance gap between the two models increases as the noise level increases with a lower θ. This trend is likely because irrelevant passages exhibit less similarity to other relevant passages within a cluster. By contextualizing passage embeddings within clusters using $\mathcal{T}$, the embeddings of irrelevant passages receive lower attention weights, effectively reducing their contribution to the aspect embeddings being generated per cluster. Consequently, this leads to improved estimation of aspect embeddings and enhances diversification effectiveness.

5.8 Discussion: MHA vs. GDKM

We discuss the strengths and weaknesses of the two implementations of the aspect extraction component $\mathcal{A}$ (Section 3.3). DUB-GDKM demonstrates the best diversification performance on both

Table 3: The α-nDCG@20 performance of DUB-GDKM with and without the Transformer $\mathcal{T}$ in the aspect extractor $\mathcal{A}$.

	θ=0.5	θ=0.6	Oracle
DUB-GDKM−$\mathcal{T}$	0.427	0.443	0.560
DUB-GDKM	**0.492**†	**0.500**†	**0.562**

evaluation datasets. Additionally, it offers a higher degree of *attributability*, as it allows for easy tracking of an extracted aspect back to a cluster of passages. On the other hand, DUB-MHA, while less effective than DUB-GDKM, still outperforms all baseline models. It has fewer parameters and does not use a clustering layer. This brings two advantages: (1) DUB-MHA requires less training data and performs better without pretraining, as evidenced by the results in Tables 1 and 2; and (2) DUB-MHA is more efficient.

5.9 Evaluation of Latent Aspects

We have established the effectiveness of the latent aspects (embeddings) extracted by DUB in terms of their utility for diversified reranking. In this section, we directly evaluate the generated query aspect representations in terms of diversity and relevance.

Compared methods. We compare aspects from LDA, BERTopic, Google query suggestions (referred to as GQS) [44], GPT-3.5 [12], and aspect embeddings from DUB-GDKM on TREC-Web. For GQS and GPT, we consider the first 8 aspects per query. It is important to note that aspects from LDA, GQS, and GPT are represented in textual form, whereas aspects from BERTopic and DUB-GDKM are represented using SBERT embeddings. To enable comparisons, we employ the same encoder $\mathcal{E}$ to map GQS and GPT query aspects into aspect embeddings. However, when aggregating token embeddings, we exclude embeddings corresponding to the query terms. This is done to avoid aspect embeddings for the same query becoming overly similar and resulting in low diversity scores. Simultaneously, we "translate" the aspect embeddings of BERTopic and DUB-GDKM into tokens by selecting the top-5 tokens whose embeddings from $\mathcal{E}$ are most similar to the aspect embedding. This approach aligns with common practices for interpreting Transformer embeddings [31, 38]. We adopt the same approach for GQS and GPT aspect embeddings, ensuring that each aspect is "expanded" and represented with 5 tokens (majority of them originally have 1 token per aspect). Consequently, we compare extracted aspects in terms of their textual form and their latent representations.

Measuring Diversity. We measure the diversity of aspects in two ways. First, we compute average dissimilarity of aspect embeddings (averaged across all pairs of aspects of the query, then averaged over all queries). We use cosine *distance* for measuring embedding dissimilarity. The second metric is token diversity, which is defined to be the percentage of unique tokens in the top 5 tokens of all aspect models. This metric is originally proposed to evaluate the diversity of topic models [32]. Table 4 shows the diversity of different aspect models. In terms of both token- and embedding-level diversity, DUB-GDKM outperforms the topic model baselines LDA and BERTopic. However, the most diverse aspects are derived from GPT. It is noteworthy that DUB-GDKM extracts 8 aspects for each query, even though the majority of queries, as indicated by TREC labels, contain fewer than 8 aspects. Consequently, it is reasonable to observe overlaps among the aspects generated by DUB-GDKM. On the other hand, aspects from GQS and GPT tend to cover more unique query intents, although these aspects are not guaranteed to be relevant to any document in the corpus.

Table 4: Evaluating the quality of extracted aspects.

Metric	Diversity		Relevance	
	token	embedding	Δ-MAP	Δ-nDCG
RM3	-	-	+0.020	+0.011
LDA	0.287	-	-0.007	-0.005
BERTopic	0.661	0.272	+0.006	+0.004
GQS	0.887	0.533	+0.008	+0.005
GPT	**0.917**	**0.673**	+0.003	+ 0.001
DUB-GDKM	0.862	0.404	**+0.026**†	**+0.017**†

Measuring relevance. Query expansion with language modeling [68] is used as an extrinsic evaluation of aspect models. The intuition of this extrinsic evaluation is that the higher the quality of the extracted aspects for a query, the higher the improvements in the retrieval performance using the expanded query with those aspects. We use the extracted aspects to estimate an expanded language model for the query as $\Pr_{+}(t|q) = \beta \Pr_{\text{ML}}(t|q) + (1-\beta)\Pr(t|\boldsymbol{A})$, where $\Pr_{\text{ML}}(t|q)$ is the maximum-likelihood language model of the original query, $\Pr(t|\boldsymbol{A})$ is the aspect language model, and β is a hyperparameter. Aspect language models are estimated as: $\Pr(t|\boldsymbol{A}) = \frac{\sum_{i=1}^{8}\Pr(t|a_i)}{\sum_{t\in V}\sum_{i=1}^{8}\Pr(t|a_i)}$. To perform query expansion, we retain the top 40 terms (5 terms each for 8 aspects) from each aspect model, using top 50 documents as the source of aspect extraction (expect GQS and GPT, which do not need documents). Additionally, we include the results of RM3 [1, 55] as a strong query expansion baseline. We set the parameters of RM3 as 40 expansion terms being selected from the top 50 documents. The interpolation parameter β is set to its default value of 0.6. The retrieval index is constructed using all documents from TREC-Web (Section 5.1), rather than the entire ClueWeb'09 Category B. We report in Table 4 the *performance difference* between using the maximum-likelihood estimate of the query language model and using the expanded query language model, in terms of MAP and nDCG. Notably, the aspect models derived from DUB-GDKM significantly outperform all other methods. This indicates that DUB-GDKM is capable of generating query aspects that are most helpful in retrieving relevant documents, offering an explanation for its superior effectiveness in diversification.

6 CONCLUSIONS AND FUTURE WORK

We described DUB, an end-to-end learnable framework for search result diversification. This framework integrates latent aspect embeddings that facilitate the joint optimization of a text encoder, query aspect extractor, and diversified document ranker. Additionally, DUB offers the capability to leverage knowledge from Wikipedia through optional pretraining, addressing the challenge of data scarcity in search result diversification. In future studies, we are interested in conducting formal evaluations to assess the interpretability of search result diversification models.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #2106282. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* (2004), 189.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining*. 5–14.

[3] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 385–394.

[4] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[5] Sawsan Alqahtani, Garima Lalwani, Yi Zhang, Salvatore Romeo, and Saab Mansour. 2021. Using Optimal Transport as Alignment Objective for fine-tuning Multilingual Contextualized Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3904–3919.

[6] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2005. Query recommendation using query logs in search engines. In *Current Trends in Database Technology-EDBT 2004 Workshops: EDBT 2004 Workshops PhD, DataX, PIM, P2P&DB, and ClustWeb, Heraklion, Crete, Greece, March 14-18, 2004. Revised Selected Papers 9*. Springer, 588–596.

[7] Doug Beeferman and Adam Berger. 2000. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 407–416.

[8] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query suggestions in the absence of query logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 795–804.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

[10] Arbi Bouchoucha, Jing He, and Jian-Yun Nie. 2013. Diversified query expansion using conceptnet. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1861–1864.

[11] Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. 2014. Integrating multiple resources for diversified query expansion. In *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings 36*. Springer, 437–442.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[13] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. The clueweb09 dataset, 2009. *URL http://boston. lti. cs. cmu. edu/Data/clueweb09* (2009).

[14] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[15] Claudio Carpineto, Massimiliano D'Amico, and Andrea Bernardini. 2011. Full discrimination of subtopics in search results with keyphrase-based clustering. *Web Intelligence and Agent Systems: An International Journal* 9, 4 (2011), 337–349.

[16] Claudio Carpineto, Massimiliano D'Amico, and Giovanni Romano. 2012. Evaluating subtopic retrieval methods: Clustering versus diversification of search results. *Information Processing & Management* 48, 2 (2012), 358–373.

[17] Ben Carterette and Praveen Chandar. 2009. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 1287–1296.

[18] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[19] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 175–184.

[20] Minsik Cho, Keivan Alizadeh-Vahid, Saurabh Adya, and Mohammad Rastegari. 2021. DKM: Differentiable k-Means Clustering Layer for Neural Network Compression. In *International Conference on Learning Representations*.

[21] Charles LA Clarke, Nick Craswell, and Ian Soboroff. [n. d.]. Overview of the TREC 2009 Web Track. ([n. d.]).

[22] Charles LA Clarke, Nick Craswell, Ian Soboroff, and Gordon V Cormack. [n. d.]. Overview of the TREC 2010 Web Track. ([n. d.]).

[23] Charles LA Clarke, Nick Craswell, Ian Soboroff, and Ellen M Voorhees. [n. d.]. Overview of the TREC 2011 Web Track. ([n. d.]).

[24] Charles LA Clarke, Nick Craswell, and Ellen M Voorhees. [n. d.]. Overview of the TREC 2012 Web Track. ([n. d.]).

[25] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 659–666.

[26] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. *arXiv preprint arXiv:2006.05324* (2020).

[27] Van Dang and Bruce W Croft. 2010. Query reformulation using anchor text. In *Proceedings of the third ACM international conference on Web search and data mining*. 41–50.

[28] Van Dang and Bruce W Croft. 2013. Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 603–612.

[29] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 65–74.

[30] Van Dang and W. Bruce Croft. 2012. Diversity by Proportionality: An Election-Based Approach to Search Result Diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 65–74. https://doi.org/10.1145/2348283.2348296

[31] Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535* (2022).

[32] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics* 8 (2020), 439–453.

[33] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview.. In *TREC*.

[34] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. 2011. Multi-dimensional search result diversification. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 475–484.

[35] Zhicheng Dou, Sha Hu, Yulong Luo, Ruihua Song, and Ji-Rong Wen. 2011. Finding dimensions for queries. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. 1311–1320.

[36] Zhicheng Dou, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song. 2015. Automatically mining facets for queries from their search results. *IEEE Transactions on knowledge and data engineering* 28, 2 (2015), 385–397.

[37] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2018. From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 125–134.

[38] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 5484–5495.

[39] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[40] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 669–679.

[41] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2022. Stochastic Optimization of Text Set Generation for Learning Multiple Query Intent Representations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4003–4008.

[42] Jiyin He, Vera Hollink, and Arjen de Vries. 2012. Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 851–860.

[43] Jiyin He, Edgar Meij, and Maarten de Rijke. 2011. Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology* 62, 3 (2011), 550–571.

[44] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 63–72.

[45] Zhiqi Huang, Puxuan Yu, and James Allan. 2023. Improving Cross-lingual Information Retrieval on Low-Resource Languages via Optimal Transport Distillation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1048–1056.

[46] Bernard J Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic. 1998. Real life information retrieval: A study of user queries on the web. In *ACM Sigir Forum*, Vol. 32. ACM New York, NY, USA, 5–17.

[47] Zhengbao Jiang, Zhicheng Dou, and Ji-Rong Wen. 2016. Generating query facets using knowledge bases. *IEEE transactions on knowledge and data engineering* 29, 2 (2016), 315–329.

[48] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. 2017. Learning to diversify search results via subtopic attention.

In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*. 545–554.

[49] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. 387–396.

[50] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.

[51] Thomas N Kipf and Max Welling. [n. d.]. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[52] Weize Kong and James Allan. 2013. Extracting query facets from search results. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 93–102.

[53] Weize Kong, Swaraj Khadanga, Cheng Li, Shaleen Kumar Gupta, Mingyang Zhang, Wensong Xu, and Michael Bendersky. 2022. Multi-Aspect Dense Retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3178–3186.

[54] Reiner Kraft and Jason Zien. 2004. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web*. 666–674.

[55] Victor Lavrenko and W. Bruce Croft. 2017. Relevance-Based Language Models. *SIGIR Forum* 51, 2 (aug 2017), 260–267. https://doi.org/10.1145/3130348.3130376

[56] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR, 1188–1196.

[57] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7871–7880.

[58] Jiongnan Liu, Zhicheng Dou, Xiaojie Wang, Shuqi Lu, and Ji-Rong Wen. 2020. DVGAN: A minimax game for search result diversification combining explicit and implicit features. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 479–488.

[59] Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni, and Jian-Yun Nie. 2014. Compact aspect embedding for diversified query expansions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[60] Ilya Loshchilov and Frank Hutter. [n. d.]. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[61] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345. https://doi.org/10.1162/tacl_a_00369

[62] Sean MacAvaney, Craig Macdonald, Roderick Murray-Smith, and Iadh Ounis. 2021. IntenT5: Search Result Diversification using Causal Language Models. *arXiv preprint arXiv:2108.04026* (2021).

[63] J MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*. 281–297.

[64] Dong Nguyen and Djoerd Hiemstra. [n. d.]. Ensemble Clustering for Result Diversification. ([n. d.]).

[65] Thong Thanh Nguyen and Anh Tuan Luu. 2022. Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11103–11111.

[66] Kezban Dilek Onal, Ismail Sengor Altingovde, and Pinar Karagoz. 2015. Utilizing word embeddings for result diversification in tweet search. In *Information Retrieval Technology: 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings 11*. Springer, 366–378.

[67] Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2019. Self-attentive document interaction networks for permutation equivariant ranking. *arXiv preprint arXiv:1910.09676* (2019).

[68] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 275–281. https://doi.org/10.1145/290941.291008

[69] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. 2020. Diversifying search results using self-attention network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1265–1274.

[70] Xubo Qin, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2023. GDESA: Greedy Diversity Encoder with Self-attention for Search Results Diversification. *ACM Transactions on Information Systems* 41, 2 (2023), 1–36.

[71] Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. 2009. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR Forum*, Vol. 43. ACM New York, NY, USA, 46–52.

[72] Filip Radlinski, Martin Szummer, and Nick Craswell. 2010. Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on World wide web*. 1171–1172.

[73] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[74] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (jun 2022), 67 pages.

[75] Razieh Rahimi, Youngwoo Kim, Hamed Zamani, and James Allan. 2021. Explaining Documents' Relevance to Search Queries. *arXiv preprint arXiv:2111.01314* (2021).

[76] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[77] Chris Samarinas, Arkin Dharawat, and Hamed Zamani. 2022. Revisiting Open Domain Query Facet Extraction and Generation. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 43–50.

[78] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*. 881–890.

[79] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2012. On the role of novelty for search result diversification. *Information retrieval* 15 (2012), 478–502.

[80] Rodrygo LT Santos, Craig Macdonald, Iadh Ounis, et al. 2015. Search result diversification. *Foundations and Trends® in Information Retrieval* 9, 1 (2015), 1–90.

[81] Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit search result diversification through sub-queries. In *Advances in Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings 32*. Springer, 87–99.

[82] Sheikh Muhammad Sarwar, Raghavendra Addanki, Ali Montazeralghaem, Soumyabrata Pal, and James Allan. 2020. Search Result Diversification with Guarantee of Topic Proportionality. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 53–60.

[83] Karen Spärck Jones, Stephen E Robertson, and Mark Sanderson. 2007. Ambiguous requests: implications for retrieval tests, systems and theories. In *ACM SIGIR Forum*, Vol. 41. ACM New York, NY, USA, 8–17.

[84] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. Modeling Intent Graph for Search Result Diversification. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 736–746. https://doi.org/10.1145/3404835.3462872

[85] Zhan Su, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2022. Knowledge Enhanced Search Result Diversification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1687–1695.

[86] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[87] TREC. 2000. Text REtrieval Conference (TREC) Data - English Relevance Judgements. https://trec.nist.gov/data/reljudge_eng.html.

[88] Saúl Vargas, Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2013. Selecting effective expansion terms for diversity.. In *OAIR*. 69–76.

[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[90] Qinglei Wang, Yanan Qian, Ruihua Song, Zhicheng Dou, Fan Zhang, Tetsuya Sakai, and Qinghua Zheng. 2013. Mining subtopics from text fragments for a web query. *Information retrieval* 16 (2013), 484–503.

[91] Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 479–488.

[92] Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* 1, 2 (jun 1989), 270–280. https://doi.org/10.1162/neco.1989.1.2.270

[93] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 113–122.

[94] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling document novelty with neural tensor network for search result diversification. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 395–404.

[95] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov decision process for search result diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 535–544.

[96] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2020. A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference (Proceedings of Machine*

Learning Research, Vol. 115), Ryan P. Adams and Vibhav Gogate (Eds.). PMLR.

[97] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. 2021. Diversification-Aware Learning to Rank using Distributed Representation. In *Proceedings of the Web Conference 2021*. 127–136.

[98] Hai-Tao Yu. 2022. Optimize What You Evaluate With: Search Result Diversification Based on Metric Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 9 (Jun. 2022), 10399–10407.

[99] Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. 2019. Corpus-based set expansion with lexical features and distributed representations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1153–1156.

[100] Puxuan Yu, Razieh Rahimi, and James Allan. 2022. Towards Explainable Search Results: A Listwise Explanation Generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 669–680.

[101] Puxuan Yu, Razieh Rahimi, Zhiqi Huang, and James Allan. 2020. Learning to rank entities for set expansion from unstructured data. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 21–28.

[102] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*. 418–428.

[103] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 3189–3196.

[104] ChengXiang Zhai, William W Cohen, and John Lafferty. 2015. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Acm sigir forum*, Vol. 49. ACM New York, NY, USA, 2–9.

[105] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 293–302.