Enhancing Dense Retrievers' Robustness with Group-level Reweighting

Anonymous ACL submission

Abstract

The anchor-document data derived from web graphs offers a wealth of paired information for training dense retrieval models in an unsupervised manner. However, unsupervised data contains diverse patterns across the web graph and often exhibits significant imbalance, leading to suboptimal performance in underrepresented or difficult groups. In this paper, we introduce `WebDRO`, an efficient approach for clustering the web graph data and optimizing group weights to enhance the robustness of dense retrieval models. Initially, we build an **embedding model** for clustering anchor-document pairs. We the embedding model through contrastive learning for link prediction, which guides the embedding model to capture the document features behind the web graph links. Subsequently, we employ the group distributional robust optimization to recalibrate the weights across different clusters of anchor-document pairs when training **retrieval models**. During training, the algorithm will assign higher weights to clusters with higher loss and focus more on worst-case scenarios. This approach ensures that the model has strong generalization ability on all data patterns. Our experiments on MS MARCO and BEIR demonstrate that our method can effectively improve retrieval performance in unsupervised training and finetuning settings. Further analysis confirms the stability and validity of group weights learned by `WebDRO`. All codes will be released via GitHub.

1 Introduction

Dense retrieval models, which encode the semantics of text into dense vectors, have become prevalent and achieved state-of-the-art performance on many public benchmarks thanks to the surge of pre-trained language models like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020). Dense retrieval models can adeptly capture semantic features in supervised scenarios, outperforming traditional sparse methods which solely retrieves documents containing keywords already present in the query given enough data, However, annotated data is inaccessible in many scenarios either due to the extensive effort needed for annotation or privacy concerns. This highlights the necessity for unsupervised dense retrievers. Several paradigms of unsupervised dense retrieval have been proposed, among which leveraging **web anchors** is a promising choice. Primarily, anchor-document pairs contain weak supervision signal that indicates overlapping information (Xie et al., 2023a; Zhang et al., 2020) and thus can approximate real search queries better than plain-text-based methods like ICT (Izacard et al., 2021). In addition, anchor resources are abundant across the Internet, bearing greater potential for the scalability of retrievers compared to methods that utilize the title and abstract of the passage.

Although the web graph contains rich data patterns, the imbalanced data distribution is an inherent flaw of unsupervised retrieval data derived from the web graph. Training robust dense retrievers with these data is challenging, as different patterns exhibit significantly different frequencies. Due to insufficient exposure to underrepresented data patterns, the model's performance on downstream tasks involving these patterns will be hindered. To mitigate this, previous research has primarily focused on instance-level data selection to identify data points whose patterns haven't been sufficiently learned by the model. To reach a more tailored selection, these approaches typically leverage feedback signals during the training process to assign a weight to each instance, such as Active Learning (Settles, 2011; Coleman et al., 2020) and Meta Adaptive Ranking (Sun et al., 2021). Naturally, the model need to be trained for multiple rounds with these methods, which is highly time-consuming and computationally intensive.

We address the lack of both efficient and effective retrieval data reweighting methods by intro-

ducing a **group-level clustering and reweighting strategy**, WebDRO. By introducing groups, WebDRO enables us to reweight training data in a single pass, significantly saving computational resources. Additionally, data clustered into groups encapsulates much more information than a single data point and represents the common characteristics of the same pattern, WebDRO can enhance the quality and stability of reweighting.

Specifically, we employ a two-step approach to train a robust dense retriever using WebDRO. The first step (which we denote as Step A) is to generate clusters from unsupervised web data. To begin with, we utilize structural information from the web graph to train an **embedding model** with contrastive learning (Step A-1). By distinguishing related web pages from unrelated ones, the embedding model can work as a link predictor that captures the semantic features of web documents. Leveraging the embedding model, we generate vector representations for documents in the training data and produce high-quality clusters through K-Means (Step A-2).

In the second step (which we denote as Step B), we utilize Group Distributionally Robust Optimization (**GroupDRO**) (Oren et al., 2019; Sagawa et al., 2019) to reweight the clusters while contrastively training the **retrieval model**. Originally designed to tackle distributional shifts in language model training, GroupDRO dynamically adjusts the weights of different groups based on the training loss. It increases the weight of the groups with large training loss, therefore highlighting the underrepresented groups that the model has difficulty learning from and enhancing the model's robustness in diverse retrieval tasks.

The main contributions of our work are:

- We explored web anchors' usage in training dense retrievers and proposed WebDRO, a clustering and group-level reweighting algorithm for unsupervised dense retriever training.
- Empirical analysis on MS MARCO (Nguyen et al., 2016) and BEIR (Thakur et al., 2021) shows that WebDRO can notably enhance dense retrievers' robustness, improving their performances on both unsupervised and finetuned settings.
- We further confirmed WebDRO's ability to efficiently identify underrepresented and challenging groups, thereby helping the model improve its capabilities more effectively.

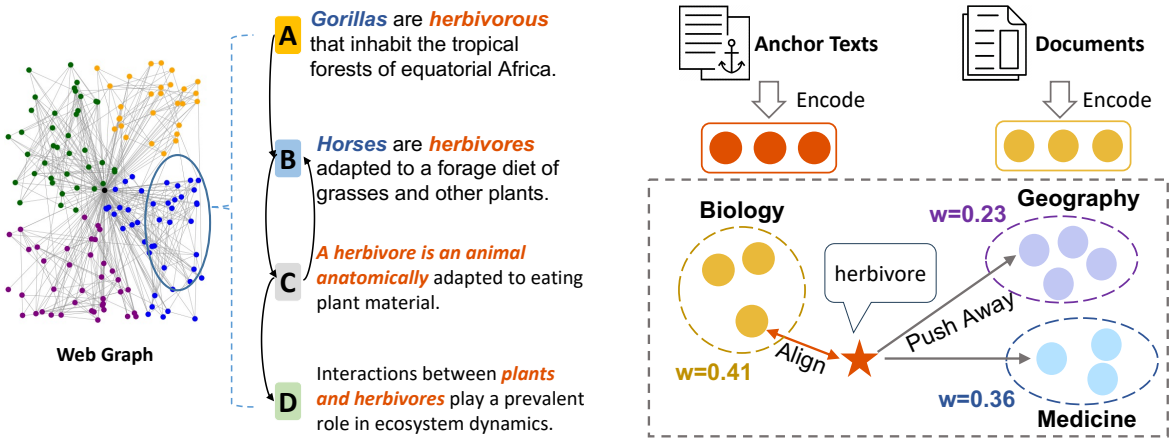
2 Related Work

This section describes relevant work about dense retrieval models and distributionally robust optimization.

Dense Retrieval. Dense Retrieval is an approach where both the query and document are separately encoded through a neural network, and the relevance score is calculated by the similarity of the encoded vectors. The prevalent training approach, introduced by Karpukhin et al. (2020), trains the network with a contrastive loss. Although dense retrieval has outperformed sparse methods (Robertson et al., 2009) in many tasks, its benefits are generally confined to well-annotated data.

Unsupervised dense retrieval efforts focus on using unannotated data to generate query-document pairs that semantically resemble contrastive pairs in supervised scenarios. For instance, ICT (Izacard et al., 2021) samples a text span in a passage as the query and the rest part as the document, and Codoc(Gao and Callan, 2022) independently sample two text spans as the query and document. More recently, researchers have exploited unsupervised data at a greater scale and with more diverse constructing methods, namely E5 (Wang et al., 2022), BGE M3 (Chen et al., 2024), and COCO-DR (Yu et al., 2022). In our work, we focus on one specific method of constructing contrastive pairs. Following Anchor-DR (Xie et al., 2023b), we use anchors and corresponding webpages on the Internet to train dense retrievers. Web anchors have been proved to be a high-quality data source for unsupervised retrievers, and have great potential for scaling up, considering the ubiquity and large number of anchors available on the Internet.

Distributionally Robust Optimization. To address the issue of distribution shift (Agarwal et al., 2021; Wiles et al., 2022), researchers have introduced Distributionally Robust Optimization (DRO) (Ben-Tal et al., 2013; Rahimian and Mehrotra, 2019; Lin et al., 2022). The goal of DRO is to optimize the worst-case performance of a model across various data distributions, ensuring its robustness against uncertain distributional changes. Specifically, GroupDRO considers robustness over groups and calculates the worst-case loss over groups by optimizing group weights and model weights concurrently. GroupDRO can be used to train more robust models, as well as to learn the importance



(a) Web-based clustering.

(b) Contrastive training with WebDRO.

Figure 1: Illustration of key steps in WebDRO. Figure 1(a) shows the clustering process in Step A, where we use web links to train an embedding model and use the model to cluster documents. Figure 1(b) shows the training process of the retrieval model in Step B. We aim to align the anchor with the corresponding document while pushing apart unrelated ones. Each group is assigned a weight that is dynamically updated based on the training loss.

of different data groups (Xie et al., 2023a). Prior applications of GroupDRO used explicitly pre-partitioned groups, and the number of groups is relatively small (< 50). In this work, we extend the applicable scope of GroupDRO to a large number of groups clustered by implicit information.

3 Methodology

In this section, we first describe the preliminary of dense retrieval (Sec. 3.1). Then we introduce our WebDRO method, which is shown in Figure 1. In Step A, we use the method in Sec. 3.2 to train an **embedding model** using the web links. Then, we cluster the anchor-document pairs based on vector representations produced by the embedding model. In Step B, we train a robust dense **retrieval model** with Group Distributionally Robust optimization (GroupDRO) as introduced in Sec. 3.3.

3.1 Preliminary of Dense Retrieval

Given a user's query q, the dense retrieval task aims to search relevant document p to satisfy the user's information need.

In dense retrieval, we train **one** single model to encode queries and documents separately as dense vectors and then conduct relevance modeling in an embedding space. We use cosine similarity to measure the relevance between queries and documents. Through contrastive training (Oord et al., 2018; Chen et al., 2020; He et al., 2020), the encoder is optimized to align the queries with related documents and pull away the irrelevant documents from queries. In other words, the model will encode relevant query-document pairs as two similar vectors, and vice versa. Specifically, the following loss function is minimized during training:

$$l_{\text{dr}}(q) = -\log \frac{e^{sim(E(q),E(p^+))}}{e^{sim(E(q),E(p^+))}+\sum_{i=1}^{k} e^{sim(E(q),E(p_i^-))}}, \quad (1)$$

where q stands for the query, p^+ stands for the related document for the query and $p_1^-, ..., p_k^-$ are negative (unrelated) documents.

3.2 Step A: Clustering with the Embedding Model

This subsection introduces our anchor-document cluster method based on a web graph. It trains an embedding model to learn the document representations using links in the web graph, and then clusters the documents according to their semantic features.

Document Relations in Web Graph. The web graph models the document relationship with hyperlinks, which allows users to navigate from one web page to another or within the same web page. It is evident that the documents linked with a hyperlink will likely share similar or related information. As shown in Figure 1(a), these four documents are related and are linked with the anchor hyperlink, *e.g.* the link between document A and document B indicate that "horses and gorillas have some similarities", showing an analogy relationship; the links between document B and document C indicate the definitions of "horses" and "herbivorous", showing the hypernymy relationship.

Embedding Model Training. We train an **embedding model** using the document relations in the web graph. We encode the documents using a pre-trained language model encoder and contrastively train the encoder. A web page can contain multiple links, so we enumerate each link to construct contrastive pairs as training examples. For each link, we regard the source web page as the query, the target web page as the positive document, and sample non-linked web pages as negative documents. The objective function is identical to that in Sec. 3.1. As a result, the **embedding model** is trained as a link predictor that can distinguish linked web pages from non-linked ones.

Document Clustering. Since the embedding model is capable of capturing webpages' features, we use it to encode all documents in our training data as dense vectors. Then we use Mini-Batch K-Means (Sculley, 2010), a variant of K-Means (Krishna and Murty, 1999) to cluster embedding vectors of documents into a fixed number of groups. Some documents have more than one anchor that links to it, these anchor-document pairs are naturally clustered into the same group. To make sure each group has statistical significance, we set a threshold of the number of documents in a group as $MinSize = 128$. All the groups with sizes smaller than $MinSize$ will be aggregated together and considered as a larger group. As this large group contains heterogeneous elements with different topics, we should not assign a single weight to represent this group. Therefore, the large group will not be considered in the reweighting discussed in Sec. 3.3.

3.3 `Step B`: Robust Dense Retriever Training

For each anchor in our corpus, we regard the text of the anchor as the query, and the content of the linked webpage as the positive document. The anchor-document pairs contain different types of documents, and the distribution of different types could vary greatly: some occur frequently, while others are rare; some are easy for the model to learn, while others are challenging. Treating all data equally may affect the performance of dense retrievers when faced with underrepresented data types. To alleviate this problem, we follow Sagawa et al. (2019) to reweight different anchor-document groups and train more robust dense retrievers.

In GroupDRO, each group is assigned a weight that indicates its importance, and the weights are dynamically updated during the training process. Specifically, we denote n as the number of groups after clustering. Then, for a training instance p in step t that belongs to group k, the training loss of `WebDRO` is calculated as:

$$L_{\text{WebDRO}}(p) = l_{\text{dr}}(p) \cdot w_k^t \cdot n \cdot C_k, \qquad (2)$$

where l_{dr} is the contrastive loss calculated using Eq. 1, w_k^t is the weight of the k-th group in step t, and C_k is size factor of the k-th group. We will next describe C_k and w_k respectively.

Size Factor (C_k). In a standard GroupDRO scenario, data are sampled from each group according to the uniform distribution. However, the size of different anchor-document groups in our method may vary significantly. To accommodate this circumstance, `WebDRO` introduces a size factor C_k:

$$C_k = \frac{\sum_{j=1}^{n} N_j}{n} \cdot \frac{1}{N_k}, \qquad (3)$$

where N_k is the size of the k-th anchor-document group. Smaller groups will have a larger C_k value, which serves as a balance for different group sizes. This is a natural extension of the original DRO, as $C_k = 1$ for all ks when all groups have the same size (all N_ks are the same).

Group Weighting (w_k). The initial weight of all groups is the same: $w_k^0 = \frac{1}{n}$. In each step, the groups' weights are updated **before** the calculation of L_{WebDRO} (Eq. 2). First, we increase group weights based on training loss and get intermediate weights w'_k. For each group k,

$$w'_k = w_k^{t-1} \cdot e^{lr \cdot C_k \cdot L_{\text{avg}k}}. \qquad (4)$$

In this formula, lr is the learning rate of weight updating. $L_{\text{avg}k}$ portraits the contribution of group k in step t's training loss, which can be calculated as:

$$L_{\text{avg}k} = \frac{\sum_{p \text{ in group } k} l_{\text{dr}}(p)}{m}, \qquad (5)$$

where m is the number of training data in step t. This is a natural extension of the case when all training data in a step belongs to a single group. $L_{\text{avg}k}$ for that group will thus be the average loss, and other group weights will remain unmodified.

After each step, the sum of group weights should be 1, so we normalize the intermediate weights to get w_k^t. For each group k,

$$w_k^t = \frac{w'_k}{\sum_{j=1}^{n} w'_j}. \qquad (6)$$

Algorithm 1 Weight Updating in `WebDRO`

Input: Model M, step number t, number of groups n, size factor $C[1, \ldots, n]$, data batch $p[1, \ldots, m]$, original weights $w^{t-1}[1, \ldots, n]$
Output: Updated weights $w^t[1, \ldots, n]$

Initialize $L_{\text{avg}}[1, \ldots, n] \leftarrow [0, \ldots, 0]$
for $i = 1$ to m **do**
    Find the group number k of $p[i]$
    Calculate the contrastive loss $l_{dr}(M, p[i])$
    $L_{\text{avg}}[k] \leftarrow L_{\text{avg}}[k] + l_{dr}(M, p[i])$
end for
for $k = 1$ to n **do**
    $L_{\text{avg}}[k] \leftarrow L_{\text{avg}}[k]$ / m
    $w'[k] \leftarrow w^{t-1}[k] \cdot \exp(lr \cdot C[k] \cdot L_{\text{avg}}[k])$
end for
$w_{\text{sum}} \leftarrow \sum_{k=1}^{n} w'[k]$
for $k = 1$ to n **do**
    $w^t[k] \leftarrow w'[k]$ / w_{sum}
end for
RETURN $w^t[1, \ldots, n]$

During training, `WebDRO` aims to increase the weights of groups with higher contrastive loss, which leads the model to draw more attention to groups that are challenging for it. This enhances the model's robustness on underrepresented groups in the source, thereby implicitly improving the model's generalization ability on unseen target queries. Algorithm 1 shows the process of weight updating, which is the same as the above description.

4 Experimental Methodology

This section introduces the training dataset, evaluation, baseline models, and details on implementation.

Training and Evaluation Datasets. We use MS MARCO (Nguyen et al., 2016) and BEIR (Thakur et al., 2021), two widely used English retrieval benchmarks, to evaluate the retrieval effectiveness in both **unsupervised** and **finetuned** settings. We collect anchor document pairs from of ClueWeb22 (Overwijk et al., 2022), a dataset containing extensive web page information, to train dense retrievers in an unsupervised setting. Since the dataset is large, we only use the "B" category, which contains the most popular 2% webpages in it. We use the training set of MS MARCO Passage as data for fine-tuning. Following previous work (Thakur et al., 2021; Xie et al., 2023b), nDCG@10 is used as our evaluation metric.

Baseline Models. We compare `WebDRO` with a commonly used sparse retriever, `BM25` (Robertson et al., 2009), and five unsupervised dense retrieval models. `Anchor-DR` (Xie et al., 2023b) serves as the baseline for direct comparison with `WebDRO`. `Anchor-DR` regards anchor texts as queries and uses anchor-document pairs without clustering. For the other four baselines, we describe them in Appendix A.2.

To evaluate `WebDRO` in a fine-tuned setting, we compare three different base models: `T5-base`, `Anchor-DR`, and `WebDRO`. These base models are finetuned in exactly the same manner.

Implementation Details. Our implementation is based on OpenMatch (Liu et al., 2021; Yu et al., 2023). When training the **embedding model** for clustering, we think the URL of web pages contains valuable information about the relations between web pages. Therefore, we incorporate the URL with the title and content as the text representation of web pages. We cluster training data into 500 groups, yet the actual data may contain slightly less than 500 groups because the groups smaller than 128 are merged (as mentioned in Sec. 3.2). We analyzed the effect of using different group numbers in Appendix A.4.

For unsupervised **retrieval models**, we start from the T5-base (Raffel et al., 2020) (with 220M parameters) checkpoint and contrastively train the dense retrieval model on ClueWeb data with in-batch negatives and `BM25` retrieved negatives for 1 epoch. Then we reinitialize the group weight and train the model for another epoch using in-batch negatives and more challenging negatives generated with ANCE (Xiong et al., 2021). To enhance the stability of `WebDRO`, we use gradient accumulation and update the group weights every 500 training step. Gradients of the model parameters are not cached and are updated every step.

For the finetuned setting, we tune base models with the training set of MS MARCO Passage with in-batch negatives and `BM25` retrieved negatives for 1 epoch. The finetuning process doesn't involve clustering or reweighting.

5 Evaluation Results

In this section, we first show the performance of different retrieval models and analyze different clus-

[1] Average score of 14 public datasets in BEIR.

Model	BM25	Unsupervised Dense Retrievers						Finetuned Dense Retrievers		
		SPAR	AugTriever	coCondenser	E5-Large	Anchor-DR	**WebDRO**	T5-Base	Anchor-DR	**WebDRO**
# Training Pairs	-	22.6M	52.4M	8.8M	270M	13.8M	13.8M	-	-	-
MS MARCO	22.8	19.2	20.6	16.2	26.2	26.15	27.37	36.47	39.13	**40.58**
trec-covid	65.6	53.1	53.5	40.4	61.8	72.16	69.26	72.43	73.98	**78.04**
nfcorpus	**32.5**	26.4	30.3	28.9	33.7	30.70	30.74	29.22	29.75	31.24
fiqa	23.6	18.5	22.3	25.1	43.2	23.79	25.12	26.72	28.11	**28.45**
arguana	31.5	42.0	39.1	44.4	44.4	28.39	31.27	47.59	**49.18**	48.02
touche2020	**36.7**	26.1	21.6	11.7	19.8	21.85	24.05	23.16	24.06	27.56
quora	78.9	70.4	82.7	82.1	**86.1**	85.57	83.36	85.28	85.48	85.85
scidocs	15.8	13.4	14.7	13.6	**21.8**	13.42	14.93	14.95	15.41	15.30
scifact	66.5	62.6	64.4	56.1	**72.3**	58.84	61.88	57.54	61.06	62.16
nq	32.9	26.2	27.2	17.8	41.7	28.83	34.48	42.48	45.24	**47.23**
hotpotqa	**60.3**	57.2	47.9	34.0	52.2	53.81	52.27	50.25	56.90	57.39
dbpedia-entity	31.3	28.1	29.0	21.5	37.1	34.61	35.34	34.01	36.39	**38.09**
fever	**75.3**	56.9	59.7	61.5	68.6	72.15	73.42	68.42	71.69	70.93
climate-fever	**21.4**	16.4	17.7	16.9	15.7	18.87	21.05	18.45	18.95	18.89
cqadupstack	29.9	27.9	27.1	30.9	**38.9**	28.82	32.97	33.64	35.79	35.25
trec-news	39.8	-	-	25.4	-	41.18	40.24	39.68	**43.05**	41.80
signal1m	**33.0**	-	-	21.4	-	22.99	24.36	26.48	27.56	27.82
bioasq	**46.5**	-	-	22.7	-	33.88	38.45	30.07	33.73	35.86
robust04	40.8	-	-	29.8	-	41.01	37.60	**44.19**	43.23	43.51
Avg. on BEIR14[1]	43.0	36.2	37.0	33.4	44.2	40.84	42.15	43.15	45.14	**46.03**
Avg. All	41.3	-	-	31.6	-	38.80	39.91	41.11	43.09	**43.89**

Table 1: Retrieval results of different models on MS MARCO and BEIR. All models are evaluated by nDCG@10. Model names in the "Finetuned Dense Retrievers" category only indicate the base model before finetuning. The best result on each task is marked in bold, and the best result among unsupervised dense retrievers is underlined.

tering methods in our approach. Then we compare WebDRO's performance in different tasks and analyze the validity of the group weights learned by WebDRO.

5.1 Overall Performance

Table 1 shows the unsupervised retrieval results of WebDRO on MS MARCO and BEIR.

In the unsupervised setting, WebDRO outperforms Anchor-DR, its counterpart without reweighting by achieving a 1.2% improvement on MS MARCO and a 1.1% improvement on the average of BEIR. WebDRO enhances retrieval performance on most datasets compared with Anchor-DR, indicating its effectiveness in identifying and highlighting rare or challenging data patterns in anchor-document pairs. WebDRO also significantly outperforms the first three unsupervised baselines.

Although WebDRO doesn't achieve state-of-the-art performance compared to larger models like E5-Large, WebDRO's average score only falls behind by 2.1%. Given that the amount of data used to train WebDRO is 5.1% of that used to train E5-Large, we believe anchor-document pairs are a high-quality resource for training unsupervised retrievers. We only use the "B" category of ClueWeb22 which accounts for 2% of the full dataset, so the anchor method can be easily scaled up and stronger dense retrievers can be trained in the same manner as WebDRO.

Model	MS MARCO	Avg. on BEIR
Anchor-DR	26.15	39.50
DRO w/URL	26.97	39.79
DRO w/Anchor-DR	26.64	39.65
WebDRO	27.37	40.60

Table 2: Performance of variants of WebDRO with different clustering strategies in an unsupervised setting.

As shown in the right part of Table 1, WebDRO also significantly outperforms Anchor-DR and T5-base on most datasets when they are all fine-tuned with the same MS MARCO data. Additionally, finetuned WebDRO outperforms BM25 on 10 subtasks (out of 19) and has an average performance gain of 2.6%. The performance gain of fine-tuned WebDRO further indicates the advantages of our clustering and reweighting method can be brought to the finetuned scenarios.

5.2 Analyses on Clustering Strategy

To compare the effectiveness of different document clustering strategies, we also implement two variants of WebDRO by using different clustering strategies. DRO w/URL directly clusters the web pages that share the same domain name into the same group. DRO w/Anchor-DR uses Anchor-DR, a **dense retriever** to encode the documents for clustering instead of using a task-specifically trained **embedding model**. The results in the unsupervised setting are shown in Table 2.

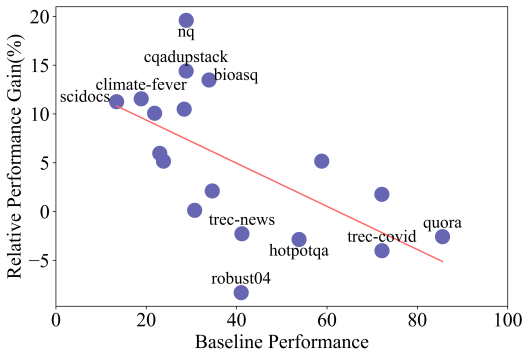


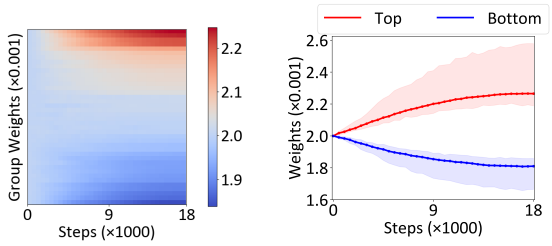
Figure 2: The relationship between relative performance gain of WebDRO and performance of Anchor-DR. Each point represents a dataset in BEIR. Datasets with high or negative gains are annotated.

From the table, we can see WebDRO outperforms DRO w/Anchor-DR by achieving about a 1% improvement. We hypothesize that this disparity arises from the fact that the clustering task concerns the relations between **documents**, whereas Anchor-DR is trained with **anchor-document pairs**, emphasizing the modeling of relations between queries and documents. A link predictor style embedding model (the one in WebDRO) focuses more on extracting document features to model the relation between web pages, making it more tailored for clustering documents.

URLs are commonly employed to encapsulate the content of a web page, and naturally, DRO w/URL exhibits an improvement over vanilla Anchor-DR. However, in comparison to DRO w/URL, WebDRO shows better retrieval performance, illustrating that titles and contents also play a crucial role in enhancing the URL-based representations of web pages. Another reason for DRO w/URL's performance decline compared to WebDRO is its coarse clustering for the documents belonging to large domains. For instance, all web pages within Wikipedia will be clustered in one single group (this group contains roughly 1M anchors, constituting 7.2% of whole training data) regardless of their topics.

5.3 Retrieval Effectiveness in Different Testing Scenarios

This section focuses on evaluating the role of the group weighting mechanism in training dense retrieval models. In Figure 2, we present the performance points of all 18 datasets from BEIR and show the relationship between Anchor-DR's performance and the relative performance gain achieved by WebDRO over Anchor-DR.



(a) The weight changes of 50 random groups.

(b) The weight changes of top and bottom groups.

Figure 3: The visualization of group weights during training. Group weights are recorded every 500 steps.

The evaluation results show that the performance gap between Anchor-DR and WebDRO tends to be larger when the baseline nDCG@10 is low. It illustrates that WebDRO performs particularly well on difficult retrieval tasks, like *scidocs* (with a relative gain of 11.26%), *hotpotQA* (19.60%) and *cqadupstack* (14.39%). Conversely, on datasets where the baseline already exhibits strong performance, such as *trec-covid* (-4.30%) and *quora* (-2.58%), the model's enhancements are marginal or even negative. This aligns with the optimization goal of WebDRO: groups with higher training losses are assigned larger weights to improve the overall performance across all datasets. This phenomenon suggests that WebDRO's unique optimization strategy effectively identifies and prioritizes challenging tasks during pretraining, offering a strategic advantage for performance gains, particularly in scenarios where traditional models struggle.

5.4 Analyze on the Group Weights Learned by WebDRO

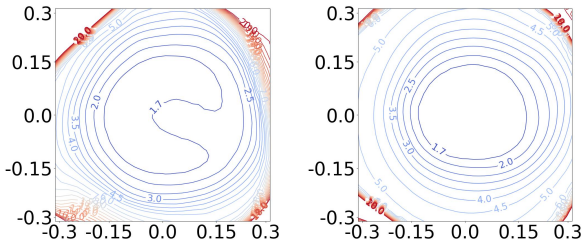
In this section, two experiments are conducted to show the stability and effectiveness of the group weights learned by WebDRO.

Trend and Distribution of Group Weights. As shown in Figure 3, we explore the distributions of group weights that learned by WebDRO.

We first randomly sample 50 groups to track the changes in group weights during training. As seen in Figure 3(a), the color within each group gradually shifts towards darker blue or darker red, indicating an approximate monotonic change in group weights. Then we rank the groups according to the weights that are assigned by our final WebDRO model, select the top-ranked and bottom-ranked 30 groups respectively, and observe the trend of

Training Data	MS MARCO	Avg. on BEIR
All Groups	27.37	40.60
Top Groups	24.12	38.53
Bottom Groups	20.70	34.70
Random	23.42	36.99

Table 3: Performance of unsupervised retriever trained with different groups. Models are evaluated with nDCG@10.



(a) Anchor-DR. (b) WebDRO.

Figure 4: Loss landscapes of Anchor-DR and WebDRO. We constrain the perturbation in each direction to $(-0.3, 0.3)$. Each closed curve in the figure indicates that the corresponding positions have the same loss.

weight changes during training. As shown in Figure 3(b), the weights of top-weighted groups and bottom-weighted groups consistently increases and decreases, further demonstrating the stability of the group weights learned by WebDRO.

Performance of Retrieval Models Trained with Different Weighted Groups. We conduct experiments to pretrain dense retrieval models using groups assigned with different weights derived from WebDRO. Specifically, we select the top and bottom 30 groups based on their weight ranks, and then sample 500k anchor-document pairs from the top and bottom groups respectively. For comparison, we also train a model using a randomly sampled set of 500k pairs. Subsequently, we utilize these subsets of training data to train dense retrieval models for 1 epoch.

As shown in Table 3, the unsupervised retriever trained with top-weighted groups significantly outperforms other models trained with an equal number of 500k anchor-document pairs. This illustrates that the groups assigned higher weights are more informative for the model, indicating that they are underrepresented in the training data and post greater challenges. The retrieval performance of the model that is trained with randomly sampled training data significantly decreases when we replace the training data with instances in bottom-weighted groups. This highlights the capacity of WebDRO to effectively identify data already well-interpreted by the model.

5.5 Analyses of the Loss Landscapes

Loss landscape is a common tool for assessing a model's stability and generalization capabilities (Li et al., 2018; Hao et al., 2019). Specifically, we apply small perturbations to the model's parameters from two random directions and visualize the losses after perturbations in a plane. The "flatness" of the loss landscape is defined as the size of the region around the origin where the training loss remains low (Hochreiter et al., 1997). A flatter loss landscape usually indicates better robustness to noises and a higher chance of converging to the optimal when randomly initialized.

As shown in Figure 4, we can see that with increasing perturbation, the training loss in WebDRO exhibits a gradual increase in all directions, whereas in Anchor-DR, the rise in training loss is steep in specific directions. Besides, it can be observed that WebDRO encompasses a larger area with low training loss (the area inside the red lines) than Anchor-DR. In summary, WebDRO presents a flatter and more uniform landscape than Anchor-DR, indicating enhanced stability in training and heightened generalization capabilities to different tasks.

6 Conclusion

This paper introduces WebDRO, an unsupervised dense retrieval model that utilizes web structures to cluster documents and adjusts group weights during contrastive training. The training process is divided into two steps. In Step-A, we use structures of the web graph to train a link prediction style embedding model for clustering. In Step-B, we utilize GroupDRO to reweight the clusters during training based on training loss. Experiments on MS MARCO and BEIR show that WebDRO achieves significant improvements over baseline models, especially on difficult tasks. By comparing different clustering strategies, we find that training the embedding model on the web graph with URL information yields the most satisfactory results. Further analysis shows that group weights learned by WebDRO are stable and valid, which means underrepresented and challenging groups are assigned higher weights consistently throughout training. This confirms WebDRO's effectiveness in improving dense retrievers' robustness.

Limitations

An important part of WebDRO is training the embedding model and generating clusters. However, we only use the cluster information in reweighting, which is a waste of resources. Research has shown that introducing more challenging negatives would enhance the model's performance (Xiong et al., 2021). As being in the same group indicates similarity, we hope to build stronger negatives with clusters to achieve better performance.

In this work, we cluster different contrastive pairs solely by their semantic features, which means contrastive pairs with similar content will be clustered in the same group. This approach is based on a hypothesis: semantically resembled contrastive pairs have similar contributions to retrieval model training. However, other factors may also influence the data point's contribution, like format, style, or even features we can't explicitly describe.

Data imbalance in unsupervised training data is the key motivation of our method. Although prior work has confirmed this phenomenon and its consequences, we didn't closely examine data distribution in our scenario due to our data's size.

References

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. 2021. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, 22:98:1–98:76.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. 2013. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? In *Proceedings of EMNLP Findings*, pages 250–262, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. In *Proceedings of ICLR*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proc. of EMNLP*, pages 981–993, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proc. of ACL*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of BERT. In *Proceedings of EMNLP*, pages 4143–4152, Hong Kong, China. Association for Computational Linguistics.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE.

Sepp Hochreiter, Sepp Hochreiter, Sepp Hochreiter, and Jürgen Schmidhuber. 1997. Flat minima. *Neural computation*, 9(1):1–42.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *ArXiv preprint*, abs/2112.09118.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781, Online. Association for Computational Linguistics.

K Krishna and M Narasimha Murty. 1999. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Proceedings of NeurIPS*, pages 6391–6401.

Fengming Lin, Xiaolei Fang, and Zheming Gao. 2022. Distributionally robust optimization: A review on theory and applications. *Numerical Algebra, Control and Optimization*, 12(1):159–212.

Zhenghao Liu, Kaitao Zhang, Chenyan Xiong, Zhiyuan Liu, and Maosong Sun. 2021. Openmatch: An open source library for neu-ir research. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2531–2535.

Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. Augtriever: Unsupervised dense retrieval by scalable data augmentation.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv preprint*, abs/1807.03748.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *Proceedings of EMNLP*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Arnold Overwijk, Chenyan Xiong, and Jamie Callan. 2022. Clueweb22: 10 billion web documents with rich information. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3360–3362.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. *ArXiv preprint*, abs/1908.05659.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ArXiv preprint*, abs/1911.08731.

D. Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 1177–1178. ACM.

Burr Settles. 2011. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings.

Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-shot text ranking with meta adapted synthetic weak supervision. In *Proceedings of ACL*, pages 5030–5043, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv preprint*, abs/2104.08663.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Ali Taylan Cemgil. 2022. A fine-grained analysis on distribution shift. In *Proc. of ICLR*. OpenReview.net.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. *ArXiv preprint*, abs/2305.10429.

Yiqing Xie, Xiao Liu, and Chenyan Xiong. 2023b. Unsupervised dense retrieval training with web anchors. *ArXiv preprint*, abs/2305.05834.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proc. of ICLR*. OpenReview.net.

Shi Yu, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2023. Openmatch-v2: An all-in-one multi-modality plm-based information retrieval toolkit. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3160–3164.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning. *arXiv preprint arXiv:2210.15212*.

Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective weak supervision for neural information retrieval. In *Proceedings of WWW*, pages 474–485. ACM / IW3C2.

A Appendix

A.1 License

Our code base, OpenMatch, uses MIT License. For datasets used in this work, MS MARCO can be used freely without a license, BEIR uses Apache-2.0 license and ClueWeb22 uses an individual agreement, which we have already obtained.

A.2 Training Details

This subsection describes the training details of our `WebDRO` model, including data processing, hyperparameters and computational resources.

Data Processing. All training data are selected from the English documents in the B category of ClueWeb22. We follow (Xie et al., 2023b) to process the training data by filtering out uninformative anchors (Homepage, Login, etc.), in-domain anchors, headers, and footers using heuristic rules. A miniBERT-based classifier model is also utilized to evaluate the quality of anchor texts. We only retain 25% anchor texts that are top-ranked.

Sizes of Datasets. The training data for training unsupervised the **retrieval model** contains 13.8M contrastive pairs, and the training data for the **embedding model** contains 5.1M contrastive pairs. For evaluation benchmarks, MS MARCO contains 8.8M documents and $8,960$ queries, and BEIR contains 44.0M documents and $47,282$ queries in total.

Hyperparameters. The learning rate and batch size are set to $3e-5$ and 384 for training the embedding model. During the stage of training retrieval models, we set the learning rate of the contrastive loss to $3e-5$ and batch size to 768. The learning rate for weight updates is set to $3e-4$.

Computational Resources. The training of our model uses 8 NVIDIA RTX A6000 GPUs (48GB of memory each). Each epoch of dense retrieval model training (for all 13.8M contrastive pairs) takes 9 to 12 hours depending on the inter-node communication efficiency.

A.3 More Details of Baseline Models

coCondenser (Gao and Callan, 2022) bases on the Condenser model (Gao and Callan, 2021) and is trained with token-level and span-level language modeling objectives. *SPAR* (Chen et al., 2022) further trains dense retrievers on Wikipedia and aims at imitating a sparse retrieval teacher. *AugTriever* (Meng et al.) focuses on extracting the queries for the documents and using them to train retrieval models. *E5-Large* (Wang et al., 2022) trains retrieval models on CC-Pair, a huge unsupervised dataset collected from multiple sources and curated with a consistency-based filter.

#Group	MS MARCO	Average on BEIR
50	26.93	40.40
200	27.43	40.48
500	27.37	40.70
1,000	27.11	40.49
5,000	26.82	40.17
15,000	27.29	40.00

Table 4: Retrieval performance of `WebDRO` with different group numbers (#Group). Results are in nDCG@10.

A.4 Analyse on the Number of Groups

As shown in Sec. 3.2, we use K-Means to cluster the documents and the number of groups (n) is critical for GroupDRO. We conducted a series of experiments to investigate the impact of varying n values on model performance. In Table 4, we can see the model performance initially is improved but later decreases as the n increases. In our training scenario, $n = 500$ is an appropriate value for `WebDRO` to achieve the best retrieval performance. If $n > 500$, the number of the clustered groups is too large and the weights of smaller groups are usually unstable or even faulty updated. If $n < 500$, the clustering is too coarse, and detailed features of documents cannot be represented through clustering. Each group will contain a large number of documents, which should be assigned different weights.