Beyond Contrastive Learning: Synthetic Data Enables List-wise Training with Multiple Levels of Relevance

Reza Esfandiarpoor[*1] **George Zerveas**[*2] **Ruochen Zhang**[1] **Macton Mgonzo**[1]
Carsten Eickhoff[3] **Stephen H. Bach**[1]
[1]Brown University [2]Microsoft [3]University of Tübingen
{reza_esfandiarpoor,ruochen_zhang,macton_mgonzo,stephen_bach}@brown.edu
gzerveas@microsoft.com c.eickhoff@acm.org

Abstract

Although synthetic data has changed various aspects of information retrieval (IR) pipelines, the main training paradigm remains: contrastive learning with binary relevance labels, where one positive document is compared against several negatives using the InfoNCE loss. This objective treats all documents that are not explicitly annotated as relevant on an equally negative footing, regardless of their actual degree of relevance, thus missing subtle nuances useful for ranking. To overcome this limitation, in this work, we forgo real documents and annotations and use large language models to directly generate synthetic documents that answer the MS MARCO queries according to *several different levels of relevance*. We also propose using Wasserstein distance as a more effective loss function for training transformer-based retrievers with graduated relevance labels. Our experiments on MS MARCO and BEIR benchmark show that our proposed approach outperforms conventional training with InfoNCE by a large margin. Without using any real documents, our method significantly improves self-supervised retrievers and is more robust to distribution shift compared to contrastive learning using real data. Our method also successfully integrates existing real data into the synthetic ranking context, further boosting the performance. Overall, we show that generating multi-level ranking contexts is a better approach to synthetic data generation for IR than just generating the standard positive and negative documents. Code: https://github.com/BatsResearch/sycl

1 Introduction

The ability of information retrieval (IR) methods to rank a collection of documents based on their relevance to a given query is critical for many applications like web search and, more recently, retrieval augmented generation (RAG) (Lewis et al., 2020; Shi et al., 2023). However, since existing large-scale IR datasets only provide binary relevance labels (Bajaj et al., 2018), most recent work predominantly trains retrievers to simply separate relevant from irrelevant documents (Ma et al., 2025). This implicitly assumes that the similarity metric learned through this simple training objective is precise enough during inference to rank multiple relevant documents differing in very nuanced ways. Instead, here we use large language models (LLMs) to generate multiple synthetic documents with *graduated relevance levels* for each query, which enables us to explicitly guide retrievers to *rank a collection of documents* during training.

Most large-scale IR datasets only provide binary relevance labels that divide documents into relevant

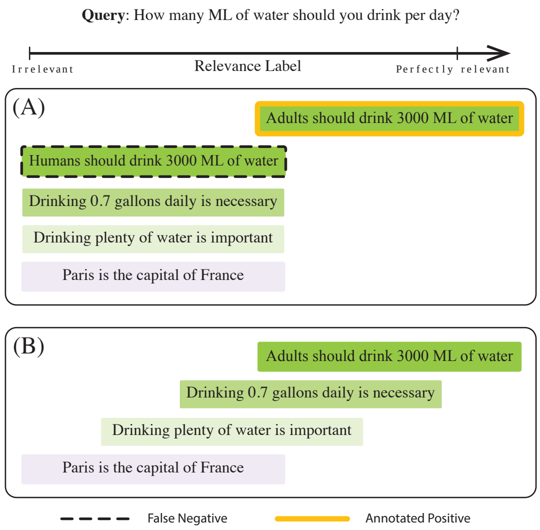


Figure 1: A) Standard contrastive training with real data treats all passages except the explicitly annotated positive passage the same, on a binary basis, regardless of their actual similarity to the given query. It is also vulnerable to false negatives. B) SyCL generates a synthetic multi-level ranking context and trains the model to rank passages based on their degree of relevance to the given query.

[*]Equal contributions.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 22860–22882
November 4-9, 2025 ©2025 Association for Computational Linguistics

("positive") and irrelevant ("negative") (Fig. 1A). Moreover, they contain very few, often only one, positive(s) per query (Bajaj et al., 2018). Similarly, even the most recent synthetic datasets adopt a binary definition of relevance (Weller et al., 2024). These limitations are reflected in the predominant training paradigm: contrastive learning with the InfoNCE loss (van den Oord et al., 2019). However, this objective differs from ranking in that all documents other than a single annotated positive are treated as negatives of equal non-relevance, regardless of their actual semantic similarity to the query. Additionally, it only considers a single relevant document at each training step. By contrast, an effective retriever is expected to *rank a collection of documents* with potentially multiple positives according to nuanced semantic differences. Also, since existing datasets like MS MARCO are sparsely annotated, many unannotated positives are falsely used as negatives, which further degrades the training signal (Qu et al., 2021).

On the other hand, the benefits of a ranking context (i.e., annotated documents for each query) with multiple relevance levels are well established in the *learning-to-rank* (L2R) literature (Cao et al., 2007; Ai et al., 2019, 2018). Most L2R works date before the advent of transformers and rely on small datasets with engineered features (Qin et al., 2010b; Chapelle and Chang, 2011; Dato et al., 2016), which are not useful for training contemporary retrievers. To train transformer-based retrievers with fine-grained annotations, some have used cross-encoders to pseudo-label the top retrieved documents for each query (Wang et al., 2022). However, because of the sparse annotations, the candidate documents are often mostly unannotated positives. In general, it is challenging to select a small set of candidate documents that covers a wide range of relevance levels, i.e., from irrelevant to perfectly relevant (see Appendix A.2 for a detailed discussion). Besides, pseudo-labeling is not applicable where existing data is scarce, such as niche domains like climate research or new tasks like retrieval with reasoning or instructions (Su et al., 2024; Weller et al., 2024). As an alternative, LLMs provide a unique opportunity to generate rich ranking contexts without these limitations.

In this paper, we propose SyCL (**Sy**nthetic ranking **C**ontext for **L**ist-wise training), a novel approach that enables training large transformer-based retrievers with graduated relevance labels. First, we create a large-scale IR dataset (~2M passages) that provides several passages with different relevance levels for each query (Fig. 1B). To avoid data annotation problems (e.g., sparsity and noise) while maintaining diversity and scale, we forgo real documents and use LLMs to generate synthetic documents with four different relevance levels for training queries of the MS MARCO dataset. During training, our dataset allows us to penalize the model's scoring choices differently depending on the relative degree of disagreement between predicted and ground-truth relevance. Second, we propose to use the Wasserstein distance as a list-wise loss function that can effectively leverage graduated relevance labels to optimize large transformer-based retrievers.

Through extensive experiments, we show the importance and effectiveness of multi-level ranking contexts. Without using any real documents, SyCL significantly improves the performance of self-supervised retrievers in both in-domain evaluation on MS MARCO and zero-shot evaluation on BEIR (Thakur et al., 2021). Most importantly, we show that generating multi-level ranking contexts instead of just positives and negatives is a better approach to synthetic data generation for IR. Specifically, training on graduated relevance labels improves the nDCG@10 score compared to training on the same synthetic data with binary labels by 5.5 and 6.4 points on average for MS MARCO and BEIR respectively. Using synthetic data alone, SyCL outperforms training on binary *real data* on out-of-domain evaluation on BEIR by an average of 2.3 nDCG@10 points. Moreover, we successfully integrate real data into the synthetic ranking context, which achieves better performance than both synthetic and real data alone. Through additional analytical experiments, we show the individual significance of the Wasserstein loss and graduated relevance labels. Finally, we analyze our data generation pipeline and find that even small 32B LLMs can generate high-quality training data. We summarize our main contributions as follows:

- We introduce SyCL, a novel method for training dense retrievers, which (a) uses publicly available LLMs to generate a large corpus of synthetic documents with graduated relevance labels and (b) uses Wasserstein distance as a list-wise loss function for training with multiple relevance levels.

- We show that using the same synthetic data, training with multiple levels of relevance outperforms standard contrastive training with binary relevance labels and InfoNCE loss.

- We show that without using real documents, SyCL significantly boosts the performance of self-supervised retrievers and is more robust to distribution shift, outperforming contrastive learning with real binary data in zero-shot evaluation on BEIR. SyCL can also combine real and synthetic data to further boost performance.

Our work uncovers the potential of LLMs for generating datasets that offer a more fine-grained definition of relevance compared to existing training data. Our findings encourage future work to explore novel data generation methods that better represent the retrieval task.

2 Related Work

Dense Retrieval Training Retrieval training pipelines have improved significantly by addressing various limitations of IR datasets. To better delineate the positive and negative regions, many have proposed using a large number of random in-batch negatives (Karpukhin et al., 2020) and similarly using existing retrievers to mine hard-to-detect negatives for each query (Qu et al., 2021; Xiong et al., 2020; Zhan et al., 2021). Some have also used existing retrievers to filter out the unannotated positives during hard negative mining (Moreira et al., 2024). However, the fundamental limitation remains: binary relevance labels provide a crude approximation of the ranking task.

Ranking Context The benefits of a multi-level ranking context are well established in the learning-to-rank (L2R) literature before the advent of transformer-based retrievers (Cao et al., 2007; Ai et al., 2019, 2018). Most L2R works use 4 to 6 levels of relevance during training (Qin et al., 2010b; Chapelle and Chang, 2011; Dato et al., 2016) and hundreds of annotated documents per query, compared to current large-scale datasets, which only provide a binary definition of relevance, and mostly a single positive document. As a result, the impact of multiple relevance levels for training large transformer models is largely unexplored, except for a few limited attempts. For example, Zerveas et al. (2022, 2023) use a large number of mined documents per query and label propagation based on a custom metric to show that even modern retrievers benefit from a rich ranking context. However, their progress is fundamentally constrained by the limitations of available datasets.

Data Annotation Pseudo-labeling with cross-encoders is one approach for obtaining fine-grained relevance judgments (Hashemi et al., 2023; Wang et al., 2021; Zeng et al., 2022; Faggioli et al., 2023; Lee et al., 2024a). However, because of the large corpus sizes, only a small set of retrieved documents is annotated for each query. Since existing datasets contain many unannotated positives (Qu et al., 2021), selecting a small set of candidate documents that covers a wide range of relevance levels is challenging, which reduces the annotation diversity for each query (Appendix A.2). Moreover, pseudo-labeling requires abundant data, which is not available for niche domains or novel applications like tool retrieval (Qu et al., 2024), or retrieval with reasoning and instructions (Weller et al., 2025; Shao et al., 2025; Su et al., 2024).

Recent works have used LLMs for judging relevance in various setups (Khramtsova et al., 2024; Thomas et al., 2024; Faggioli et al., 2023; Balog et al., 2025; Jin et al., 2024; Chen et al., 2024). However, the costs limit the scale of relevance judgments with LLMs. Often, LLMs are used to only rerank the retrieved documents for a small number of test queries (Zhuang et al., 2024; Qin et al., 2023; Sun et al., 2023; Ma et al., 2023). LLMs are also used to create small tests for evaluation or to judge the quality of other models (Rahmani et al., 2025). Furthermore, a few works have used a small set of LLM annotations to fine-tune downstream rerankers (Pradeep et al., 2023a,b). In addition to the extra costs, all the aforementioned problems for selecting candidate documents also apply to data annotation with LLMs.

Finally, it is possible to create fine-grained relevance data from search engine logs (Rekabsaz et al., 2021). However, this also faces major challenges for rare queries or novel variants of the retrieval task where existing data is scarce (see Appendix A.1).

Synthetic Data Generation IR pipelines have integrated synthetic data in different ways. A popular approach is to create synthetic queries for existing passages (Dai et al., 2022; Bonifacio et al., 2022; Jeronymo et al., 2023; Alaofi et al., 2023; Lee et al., 2024b). Another approach is to enhance the quality of existing queries (Wang et al., 2023b; Shen et al., 2023; Jagerman et al., 2023; Rajapakse and de Rijke, 2023; Anand et al., 2023; Li et al., 2024; Dhole

Instruction:

(omitted) Given a text query, your mission is to write four different passages, each with a different level of relevance to the given query.
- **Perfectly relevant**: a passage that is dedicated to the query and contains the exact answer.
- **Highly relevant**: a passage with some answer for the query, but the answer may be unclear, or hidden amongst other information.
- **Related**: a passage that seems related to the query but does not answer it.
- **Irrelevant**: a passage that has nothing to do with the query.

Passage generation instructions
- All passages should be about *{{sentences}}* sentences long.
- All passages require *{{difficulty}}* level education to understand.
- *The very first sentence of the passage must NOT completely answer the query.* *

(omitted)

Example Input:

Query: {{*example query*}}

Example Output:

[Perfectly relevant passage]
{{*example perfectly relevant passage*}}
(omitted)

Figure 2: To create a multi-level ranking context for dense retrieval training, we prompt the LLM to sequentially generate four passages with graduated relevance levels for each query. To generate diverse passages, we randomly sample the value of `{{sentences}}` and `{{difficulty}}` for each prompt. To avoid easy-to-identify passages, we include the instruction with "*" in the prompt for a random subset of queries. See Appendix E for details.

and Agichtein, 2024; Zhang et al., 2024). More recently, synthetic data has played an important role in training dense retrievers with reasoning (Shao et al., 2025) and instruction-following (Weller et al., 2024; Asai et al., 2022; Wang et al., 2024) capabilities. Despite this diversity, all existing works generate synthetic data only with binary relevance levels, which inherits many problems of existing IR datasets discussed in Section 1. By contrast, we use the flexibility of LLMs to overcome the limitations of real data and generate multi-level ranking contexts, which are more suitable for training dense retrievers.

3 Synthetic Ranking Context for List-wise Training (SyCL)

To better approximate the inference objective, we propose to train dense retrievers on passages with multiple levels of relevance, thus creating a rich *multi-level ranking context* for each query. Since most large-scale IR datasets only provide passages with binary ground truth labels, we use LLMs to generate passages with graduated relevance levels for each query (Section 3.1). Additionally, we propose to use Wasserstein distance as a loss function that uses graded relevance labels and multiple positives per query more effectively than alternative list-wise losses (Section 3.2).

3.1 Multi-level Ranking Context

We leverage open-source LLMs to generate multi-level ranking contexts for the MS MARCO training queries at scale. We use the official TREC Deep Learning[1] relevance guidelines to prompt LLMs to write passages that answer each query at four different levels of relevance: perfectly relevant, highly relevant, related, and irrelevant (Fig. 2). See Appendix E for the exact prompt.

For ranking, the *relative* relevance of passages is the most important. Even with clear instructions, when asked to generate passages of a specified relevance level without references, the LLM is not aware of how each will compare to other independently generated documents of the same, higher, or lower specified relevance to the same query. Thus, we prompt the LLM to generate all four passages for each query *sequentially* in the same inference session. This allows the LLM to gradually decrease the relevance of each generated passage relative to already generated passages in its context in order to achieve the correct ranking order. To help the LLM better understand the task, we provide one in-context example consisting of a query and four passages, i.e., one passage for each relevance level.

Corpus Diversity Additionally, the in-context example reduces the distribution shift between the synthetic and real passages in terms of attributes like style. Without examples, our synthetic documents tend to be distinctly clearer and more direct than real passages. Hence, to increase the diversity of synthetic passages, for each prompting instance, we select a different in-context example

[1]https://trec.nist.gov/data/deep2019.html

from TREC DL 2023, which is meant for the new version of the MS MARCO dataset (v2) and not used for training or evaluation by recent works. Specifically, we randomly sample one of the 82 queries and one of its corresponding passages for each level as the in-context example. This requires a very small number of ground truth labels (328 labeled passages in total), and compared to the scale of annotation in MS MARCO (more than 500k annotated queries), the incurred cost is negligible.

Moreover, similar to Wang et al. (2024), for each prompt, we use templates to specify a randomly sampled passage length and difficulty level. We also noticed that the LLM tends to provide the exact answer to the query in the very first sentence of the perfectly relevant passages, making them easily identifiable. To prevent this, we explicitly instruct the LLM to avoid this in a random subset of prompts. See Appendix E for more details.

We use simple text processing to extract the four passages from the LLM response and assign them sequential labels, `{3,2,1,0}`, based on the specified relevance level for which they were generated.

3.2 Training with Multiple Levels of Relevance

To effectively leverage the generated multi-level ranking contexts, we propose using the 2-Wasserstein distance as loss function. Although it has been used in retrieval pipelines as a distance in different roles, e.g., regularization (Yu et al., 2020), to the best of our knowledge we are the first to propose it as a relevance loss function for training dense retrievers. We use a differentiable analytical expression of Wasserstein distance that can be efficiently computed when comparing two Gaussian distributions (Mathiasen and Hvilshøj, 2020). Although neither our ground truth nor estimated score distributions are Gaussian, this approximation outperforms the most popular list-wise loss functions. For two multivariate Gaussian distributed inputs $X \sim \mathcal{N}(\mu_x, C_x)$ and $Y \sim \mathcal{N}(\mu_y, C_y)$, where μ and C are the mean and covariance of each distribution, we calculate the 2-Wasserstein distance as follows:

$$\mathrm{D}(X,Y) = \|\mu_x - \mu_y\|^2 - \mathrm{tr}(C_x + C_y - 2(C_x C_y)^{\frac{1}{2}}) .$$

During training, we present labels and predicted scores as matrices H and $\hat{H}$ of shape `(batch size, ranking context size)` and minimize $\mathrm{D}(H, \hat{H})$.

Compared to KL divergence, which has been used as a multi-level list-wise loss function (Zerveas et al., 2023; Wang et al., 2022), the Wasserstein distance has the following main advantages. First, distributing probability mass over candidate documents is penalized according to their ground-truth score distance from the ground-truth target document: assigning some probability mass to a document with g.t. label 0 instead of the correct document with g.t. 3 is penalized more strongly than assigning it to a document with g.t. label 2. By contrast, the KL divergence is insensitive to this relative distance in the estimated score distribution. As long as the g.t. relevant document (or any other document) is not assigned its due g.t. probability mass in the estimated score distribution, it will be penalized the same regardless of where this probability mass goes. Second, it is computed by comparing the ground truth and estimated score distributions across documents of the entire batch, not only across those in the context of a single query. We hypothesize that this acts as a regularization, e.g., granting resilience to the range of score values or outliers.

4 Experiments

Our experiments demonstrate the effectiveness of synthetic multi-level ranking contexts and the Wasserstein loss for training dense retrievers. First, without using any real documents or annotations, SyCL fine-tuning improves the performance of self-supervised dense retrievers. Second, we show that the Wasserstein loss with multiple levels of relevance outperforms InfoNCE using the same queries and passages. Third, we find that SyCL training only on synthetic documents performs similarly to contrastive training with real data of the same size on TREC DL, while on average, it outperforms it in terms of out-of-domain generalization on BEIR. Overall, we achieve the best ranking effectiveness when incorporating existing real data into our synthetic multi-level ranking context. Through additional analytical experiments, we show the individual impact of the Wasserstein loss and graduated relevance labels. Finally, we inspect different components of our data generation pipeline and find that even smaller, 32B-scale LLMs can generate high-quality data comparable to larger 70B-parameter models.

4.1 Setup

Training We use Llama 3.3 70B (Dubey et al., 2024) to generate one passage for each level of

nDCG@10	**DL19**	**DL20**	MM Dev	FEVER	HotpotQA	FiQA	NQ	Quora	Touche
Base Contriever (BC)	45.5	44.8	20.6	66.8	48.2	24.6	25.4	83.5	18.6
BC + InfoNCE $_{\text{Synth.}}$	55.3	51.5	26.3	68.0	46.4	26.8	33.2	75.8	15.0
BC + WS $_{\text{Synth.}}$	59.6^{ab}	59.8^{ab}	30.2^{ab}	81.8^{ab}	57.2^{ab}	27.3^{a}	41.9^{ab}	83.3^{b}	20.3^{b}
BC + InfoNCE $_{\text{Real}}$	63.0	61.2	34.2	69.6	59.8	29.1	42.8	81.7	14.6
BC + WS $_{\text{Synth. + Real}}$	63.2	61.6	32.9	80.6	59.8	30.0	42.5	83.7	16.8

nDCG@10	CQADup Android	Scidocs	Climate FEVER	DBPedia	TREC COVID	Scifact	NFCorpus	ArguAna	**BEIR Avg**
Base Contriever (BC)	37.5	15.1	15.2	29.4	27.7	63.9	32.4	31.4	37.1
BC + InfoNCE $_{\text{Synth.}}$	35.0	15.1	21.4	32.0	26.6	62.5	31.5	26.4	36.8
BC + WS $_{\text{Synth.}}$	39.0^{b}	16.4^{ab}	27.0^{ab}	36.7^{ab}	52.7^{ab}	62.0	31.8	28.2^{ab}	43.2
BC + InfoNCE $_{\text{Real}}$	38.2	16.2	18.3	37.6	34.0	65.1	31.5	33.6	40.9
BC + WS $_{\text{Synth. + Real}}$	40.5	16.0	25.5	38.9	51.2	66.6	33.0	33.0	44.2

Table 1: Ranking effectiveness (nDCG@10). Base Contriever (BC): self-supervised Contriever model. 'BC +' denotes the fine-tuning setting in terms of **loss function**: InfoNCE / Wasserstein (WS), and **type of data**: real data from the MS MARCO training set with annotated positives and mined hard negatives (Real) / fully synthetic multi-level documents (Synth.) / combination. DL19, DL20, and MM Dev are the TREC DL 2019, TREC DL 2020, and Dev evaluation sets of MS MARCO. Evaluation on the rest of sets is zero-shot. Symbols ^{a} and ^{b} denote a statistically significant difference (paired t-test) with $p < 0.05$ when compared to BC and BC + InfoNCE $_{\text{Real}}$, respectively. Purple: SyCL, our method.

relevance (i.e., ranking context size of four) for training queries of the MS MARCO dataset (total of ~2M passages). During training, we use all passages corresponding to other queries in the batch as level zero passages in the multi-level ranking context of a given query. We use the unsupervised Contriever (Izacard et al., 2021) model as our base model. See Appendix G for experiments with other models. More implementation details are provided in Appendix F.

Evaluation For in-domain evaluations, we use the TREC DL 2019, TREC DL 2020, and Dev set of the MS MARCO dataset. To evaluate how well our model performs in the real world, we use the 14 publicly available datasets in the BEIR benchmark (Thakur et al., 2021) for out-of-domain evaluation. To simplify our BEIR evaluations for duplicate question retrieval, we only use the Android subforum of the CQADupStack dataset.

4.2 Results

Table 1 shows our main results on the effectiveness of using a synthetic multi-level ranking context with the Wasserstein loss to train dense retrievers.

SyCL significantly improves the performance of the unsupervised Contriever model for both in-domain evaluation on the MS MARCO dataset and out-of-domain evaluation on the BEIR benchmark. In terms of nDCG@10, our method improves the base model performance by 6.2 across BEIR, 14.1 on TREC DL19, and 14.9 on TREC DL20.

Notably, for in-domain evaluation, the performance boost for the DL19 and DL20 sets is more significant than that of the Dev set (9.7). This is expected: MS MARCO Dev is extremely sparsely annotated (mostly, one positive per query) and missing most real positive documents. Compared to contrastive training with a single positive, a training method like ours teaches the model to distribute relevance scores among more documents in the ranking context (see Fig. 4 in the Appendix). Consequently, it has a much higher probability of assigning a high score to documents other than the annotated positive, and the chance for the latter to be displaced to lower ranks increases. Therefore, the question is whether the documents displacing the ground-truth positive are indeed relevant. Qualitative inspection of ranked documents (Table 12 in the Appendix) and evaluation on more densely annotated sets (Table 1) indicate that the answer is affirmative and may explain the difference in performance improvements. DL19 and DL20 additionally provide multi-level relevance labels, which helps to better evaluate the fine-grained ranking capabilities of retrievers.

	DL19	DL20	MS Dev	BEIR
Synth. Binary + WS	48.9	48.7	22.1	40.5
Synth. Multi-Level + WS	59.6	59.8	30.2	43.2
Real Binary + WS	50.6	47.4	23.6	36.6
Real Binary + InfoNCE	63.0	61.2	34.2	40.9

Table 2: Top: nDCG@10 of models trained with Wasserstein loss on the same synthetic data with binary ($\{1, 0\}$) and graduated ($\{3, 2, 1, 0\}$) relevance labels. Bottom: nDCG@10 of models trained with Wasserstein and InfoNCE loss on real data with binary labels.

Loss	DL19	DL20	MS Dev	BEIR
Approx. nDCG	54.7	52.5	27.8	39.1
RankNet	54.3	48.9	24.6	35.3
ListNet	56.9	55.4	27.5	42.2
KL-div	56.1	54.9	27.4	42.1
Wasserstein	59.6	59.8	30.2	43.2

Table 3: Performance (nDCG@10) of models trained on multi-level synthetic data with different list-wise losses.

Multi-level ranking context with Wasserstein loss uses the same data more effectively than InfoNCE. For an apples-to-apples comparison with the standard contrastive training, we train the model with InfoNCE loss using the same synthetic passages (InfoNCE $_{\text{Synth.}}$ in Table 1). For this, we use the passages from levels 3 and 2 as positives and passages from levels 1 and 0 as negatives. Although both setups use the same queries and passages, multi-level ranking context with Wasserstein loss uses the data more effectively and clearly outperforms contrastive training.

To evaluate contrastive training with real data, we use the human-annotated positives and two hard negatives mined by BM25 to match the number of negatives in synthetic data (InfoNCE $_{\text{Real}}$ in Table 1). Although training with real, labeled documents leads to slightly better performance for in-domain evaluation on MS MARCO, training exclusively on synthetic documents performs comparably. On the other hand, SyCL better generalizes to out-of-domain datasets in the BEIR benchmark and outperforms real data by 2.3 nDCG@10 on average. This indicates better robustness to distribution shift and unseen data, which has been argued to be the most important attribute of IR methods for real-world applications (Thakur et al., 2021).

Augmenting real data with multi-level synthetic passages further improves performance. To benefit from both real and synthetic data, we assign relevance levels 3 and 1 to positive and negative real passages respectively, and incorporate them into the synthetic multi-level ranking context. Combining synthetic and real data improves SyCL's ranking effectiveness on DL19 from 59.6 to 63.2, and on DL20 from 59.8 to 61.4. Compared to training with real data and the InfoNCE loss, training with SyCL on the combined data improves nDCG@10 scores from 40.9 to 44.2 on the BEIR benchmark. Adding real data seems to slightly degrade performance on MS MARCO Dev, which we attribute to its extremely sparse annotation (see our earlier discussion in this section).

5 Additional Analysis

Fine-grained relevance levels are necessary for achieving good performance. To separate the impact of using multiple relevance levels from the Wasserstein loss, we repeat our main experiment with the Wasserstein loss but instead use binary labels. We assign relevance levels 1 and 0 to more relevant (levels 3 and 2) and less relevant (levels 1 and 0) synthetic passages, respectively (Table 2 top). Even with the same data and loss function, fine-grained relevance levels are necessary for achieving good performance: using binary relevance levels instead decreases the boost in performance by 4.0 nDCG@10 on average across all sets.
Although our main comparison is between binary and multi-level synthetic data, we also experiment with fine-tuning on real binary data using Wasserstein loss (Table 2 bottom). For real data with binary labels, InfoNCE performs better than Wasserstein loss. Therefore, without a multi-level ranking context, the Wasserstein loss by itself does not explain the performance gains of our approach, which reinforces our main claim: the combination of synthetic multi-level data and the Wasserstein loss is particularly effective for fine-tuning dense retrievers.

Wasserstein loss is more effective than other list-wise loss functions. We compare our proposed Wasserstein loss against other list-wise loss functions that can take advantage of multiple levels of relevance (Table 3). We evaluate the Approximate NDCG (a smooth, differentiable approximation of the nDCG metric) (Qin et al., 2010a), RankNet (Burges et al., 2005), and ListNet (Cao et al., 2007) loss functions, which have been used extensively in learning-to-rank approaches before

	DL19	DL20	MS Dev	BEIR
Direct Synth. Binary	47.8	40.4	20.5	36.6
Approximated Synth. Binary	58.4	52.0	26.0	37.6

Table 4: Performance using InfoNCE loss with binary passages directly generated by the LLM and approximated binary passages (i.e., multi-level passages with with binary labels)

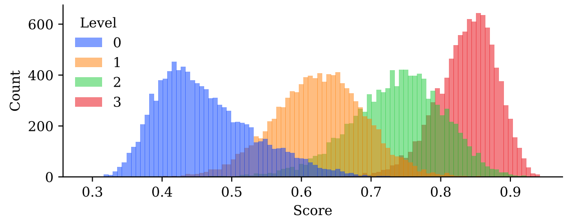


Figure 3: Distribution of the similarity scores between queries and the corresponding synthetic passages for each relevance level.

	DL19	DL20	MS Dev	BEIR
Llama 3.3 70B	59.6	59.8	30.2	43.2
Qwen 2.5 72B	60.0	57.9	30.3	42.4
Qwen 2.5 32B	61.1	56.9	29.3	42.9

Table 5: Performance of models trained with synthetic data generated by different LLMs.

the advent of dense retrieval. We also evaluate the KL divergence, which is often used for model distillation but has also been used for training with a multi-level ranking context (Zerveas et al., 2022, 2023). Except for RankNet, all other loss functions take advantage of multiple levels of relevance and outperform the binary InfoNCE loss. However, the Wasserstein loss is the most effective and provides significant gains over the next best loss function (ListNet).

Generating multi-level synthetic data is better even for binary training. In our experiments thus far, we approximate binary synthetic data by using the same multi-level synthetic passages but converting the labels from multi-level to binary. This helps us study the impact of label granularity without confounding factors like variation in passage content. We now evaluate directly generating binary data using Qwen 2.5 32B and report the results (nDCG@10) in Table 4 (exact prompt in Appendix E). The bespoke binary synthetic passages perform even worse than the simulated binary passages used in our main experiments. This further strengthens our claim about the merits of generating multi-level synthetic data. We hypothesize that when prompted to generate passages with multiple levels of relevance, the LLM more precisely controls the content of each passage in order to meet the relevance requirements, resulting in more nuanced and challenging passages.

Even with a strict interpretation of relevance, models trained with SyCL outperform BM25 without using any real passages. We show that even under a strict interpretation of relevance labels, our method outperforms BM25 without using any real passages or their annotations (Table 8 in the Appendix). Following TREC guidelines, we exclude passages with relevance label 1 for the strict evaluation setup. To be a viable approach for practical applications, dense retrievers should at least perform better than BM25, which does not require any training and still achieves strong performance. However, most dense retrieval methods fail to outperform BM25 without additional fine-tuning on labeled training data. Recently, Wang et al. (2022) managed to outperform BM25 without using any labeled data. However, they resorted to a complex multi-stage training pipeline to achieve this, while we use synthetic data to better capture the ranking objective during training with a simple pipeline.

5.1 Analysis of Synthetic Data Generation Process

LLMs successfully follow the definition of relevance levels. To check if synthetic passages adhere to their corresponding relevance level, we use a high-quality embedding model, `e5-mistral-instruct` (Wang et al., 2023a), to measure the similarity between 10,000 randomly selected queries and their corresponding synthetic documents. Figure 3 reports the distribution of similarity scores for passages in each relevance level. We find that the generator LLM understands the relevance levels and appropriately decreases the relevance between the query and generated document based on its pre-specified target level and the documents sequentially generated before it.

We provide a sample of the generated passages in Table 6, which shows that the LLM first generates a positive passage that fully answers the query and then, with some nuanced changes, creates a less relevant positive passage that provides a partial answer to the query, and similarly keeps reducing the relevant content for the other two less relevant passages in the context.

Query	what do cells do?
Level 3	Cells are the basic structural and functional units of living organisms. They are responsible for carrying out various functions necessary for life, including growth, reproduction, metabolism, and response to stimuli. Cells take in nutrients, convert them into energy, and eliminate waste products. They also contain genetic material, known as DNA, which carries information necessary for the cell's development and function. Additionally, cells communicate with each other to coordinate their activities and maintain overall health.
Level 2	The human body is made up of trillions of cells, each with its own unique characteristics and functions. Some cells, such as muscle cells, are specialized for movement and contraction, while others, like nerve cells, are responsible for transmitting signals and controlling various bodily functions. Cells also have the ability to adapt to their environment, change shape, and even migrate to different parts of the body. For example, immune cells can recognize and respond to foreign substances, helping to protect the body from infection and disease.
Level 1	Tissues are groups of similar cells that work together to perform specific tasks. Organs are structures made up of two or more types of tissues that function together to maintain the overall health of an organism. The study of the structure, behavior, and function of tissues and organs is essential for understanding how the body works and how diseases develop. By examining the relationships between different tissues and organs, researchers can gain insights into the complex processes that occur within living organisms.
Level 0	The city of Paris is known for its stunning architecture, rich history, and vibrant cultural scene. Visitors from around the world flock to the City of Light to marvel at iconic landmarks like the Eiffel Tower, Notre-Dame Cathedral, and the Louvre Museum. The city's charming streets and alleys are lined with quaint cafes, boutiques, and restaurants, offering a unique and unforgettable experience for travelers. Whether you're interested in art, fashion, or food, Paris has something to offer everyone.

Table 6: Synthetic multi-level ranking context generated for one query. The LLM successfully follows the defined relevance levels and generates passages with correct relative similarity to the given query.

	DL19	DL20	MS Dev	BEIR
Full	59.6	59.8	30.2	43.2
No in-context example	59.5	60.0	29.9	42.8
No random variation	59.9	58.3	29.8	42.8

Table 7: Impact of prompt design on retrieval performance. Full: our main prompt. No IC example: prompt without in-context examples. No random variation: prompt without randomly sampled instructions (e.g. length requirement).

Small LLMs also generate high-quality data. To understand the impact of the LLM on the quality of the synthetic data, we also generate data with two other LLMs, Qwen 2.5 72B and Qwen 2.5 32B (Team, 2024), and use it to train the retriever similar to our main experiments (Table 5). For in-domain evaluation, data generated with larger LLMs leads to better performance on DL20 and Dev splits of MS MARCO. However, for out-of-domain evaluation on BEIR datasets, data generated with the smaller Qwen 2.5 32B leads to performance similar to data generated with Llama 3.3 70B. Although recent works use 70B scale public or larger proprietary models (Wang et al., 2023a; Weller et al., 2024), our results show that data generated with larger models does not always lead to better performance. See Appendix B for experiments using combined data from all LLMs.

We investigate the impact of the in-context example and the randomly selected instructions (e.g., length) on the quality of the synthetic data. We create two alternative prompts, one without the in-context examples and the other without the randomly selected instructions, and use the resulting data for training (Table 7). Although both techniques contribute to the quality of synthetic data, in-context examples are more important, especially for out-of-domain generalization to BEIR datasets.

6 Conclusion

In this work, we introduce SyCL, a novel method that first uses LLMs to generate rich multi-level ranking contexts and then the Wasserstein distance to train retrievers with multiple levels of relevance. We show that LLMs can successfully generate synthetic data with graduated relevance levels, significantly improving the effectiveness of unsupervised retrievers. When using the same synthetic queries and passages, SyCL utilizes the available data more effectively and performs better than training with binary relevance labels. SyCL can also combine real and synthetic datasets to further improve performance. Moreover, we show that Wasserstein distance is more effective at fine-tuning transformer-based retrievers with graduated relevance labels and performs better than the usual list-wise loss functions. Our results show that generating multiple passages with graduated relevance levels is a better approach to synthetic data generation for IR than generating the standard positive and negative passages. These results encourage future work to explore synthetic data generation methods that are better suited for information retrieval tasks, going beyond the binary definition of relevance.

Limitations

LLM Capabilities Similar to other works on synthetic data generation, our work is limited by the capabilities of LLMs. For instance, data generation for specialized domains could pose a challenge for existing LLMs, especially at smaller scales. Considering the progress in generating synthetic instruction tuning data for specialized domains (Nayak et al., 2024), we encourage future work to explore opportunities to expand applications of synthetic ranking data to specialized domains as well.

Dependency on Existing Queries Our work requires the availability of a collection of user queries in the target domain. For many domains, a large collection of user queries is already provided by existing datasets or can be collected from online forums like Reddit or from users' conversation history with LLM assistants. However, for very rare applications where none of these resources is available, we encourage future work to explore the combination of our work with synthetic query generation techniques (Wang et al., 2024). However, generating a large collection of queries from scratch also comes with its own challenges. While there are many frequently occurring queries, 70% of (distinct) queries occur only once (Brenes and Gayo-Avello, 2009). Therefore, the LLM would be challenged to imagine representative user queries in most situations.

More Fine-grained Relevance Levels Moreover, we assume that LLMs understand the difference between relevance levels and can generate suitable data accordingly. We show experimentally that this is, in fact, the case, and LLMs successfully generate documents with four different relevance levels. However, we speculate that if we increase the number of relevance levels, after a certain point, the differences would be too nuanced for existing LLMs to recognize and follow. We encourage future work to study the limitations of existing LLMs in terms of understanding nuanced semantic differences through instructions and also explore more advanced approaches for controlling the semantic similarity of the generated documents.

Ethical Considerations

Since we use the MS MARCO training queries to guide the data generation process, our synthetic data might inherit the social biases and ethical concerns related to the MS MARCO dataset. Moreover, similar to other works on synthetic data generation, our data also inherits the social biases and ethical concerns related to the LLM used for generating the synthetic documents. Although we did not observe any harmful content during the course of this project, a principled analysis of social biases, factual correctness, and other ethical concerns is needed before use in sensitive real-world applications.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. RISE-2425380. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Disclosure: Stephen Bach is an advisor to Snorkel AI, a company that provides software and services for data-centric artificial intelligence.

References

Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning a Deep Listwise Context Model for Ranking Refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 135–144, New York, NY, USA. Association for Computing Machinery.

Qingyao Ai, Xuanhui Wang, Sebastian Bruch, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2019. Learning Groupwise Multivariate Scoring Functions Using Deep Neural Networks. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 85–92, Santa Clara CA USA. ACM.

Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. 2023. Can generative llms create query variants for test collections? an exploratory study. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1869–1873.

Abhijit Anand, V. Venktesh, Vinay Setty, and Avishek Anand. 2023. Context aware query rewriting for text rankers using llm. *ArXiv*, abs/2308.16753.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268 [cs]*. ArXiv: 1611.09268.

Krisztian Balog, Donald Metzler, and Zhen Qin. 2025. Rankers, judges, and assistants: Towards understanding the interplay of llms in information retrieval evaluation. *arXiv preprint arXiv:2503.19092*.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

David J. Brenes and Daniel Gayo-Avello. 2009. Stratified analysis of AOL query log. *Information Sciences*, 179(12):1844–1858.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, volume 14 of *Proceedings of Machine Learning Research*, pages 1–24, Haifa, Israel. PMLR.

Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. 2024. Attention in large language models yields efficient zero-shot re-rankers. *arXiv preprint arXiv:2410.02642*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Fast Ranking with Additive Ensembles of Oblivious and Non-Oblivious Regression Trees. *ACM Transactions on Information Systems*, 35(2):15:1–15:31.

Kaustubh D. Dhole and Eugene Agichtein. 2024. Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. *ArXiv*, abs/2404.03746.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50.

Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. In *EMNLP*.

Luyu Gao and Jamie Callan. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1.

Helia Hashemi, Yong Zhuang, Sachith Sri Ram Kothur, Srivas Prasad, Edgar Meij, and W Bruce Croft. 2023. Dense retrieval adaptation using target domain description. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 95–104.

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

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.

Can Jin, Hongwu Peng, Shiyu Zhao, Zhenting Wang, Wujiang Xu, Ligong Han, Jiahui Zhao, Kai Zhong, Sanguthevar Rajasekaran, and Dimitris N Metaxas. 2024. Apeer: Automatic prompt engineering enhances large language model reranking. *arXiv preprint arXiv:2406.14449*.

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 the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Leveraging llms for unsupervised dense retriever ranking. In

Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1307–1317.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024a. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024b. Gecko: Versatile text embeddings distilled from large language models. *Preprint*, arXiv:2403.20327.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2024. Can query expansion improve generalization of strong cross-encoder rankers? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2321–2326.

Xueguang Ma, Xi Victoria Lin, Barlas Oguz, Jimmy Lin, Wen-tau Yih, and Xilun Chen. 2025. Drama: diverse augmentation from large language models to smaller dense retrievers. *arXiv preprint arXiv:2502.18460*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Alexander Mathiasen and Frederik Hvilshøj. 2020. Backpropagating through fr\'echet inception distance. *arXiv preprint arXiv:2009.14075*.

Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*.

Nihal V Nayak, Yiyang Nan, Avi Trost, and Stephen H Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv:2402.18334*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*.

Tao Qin, Tie-Yan Liu, and Hang Li. 2010a. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 13:375–397.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010b. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Towards completeness-oriented tool retrieval for large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1930–1940.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Hossein A Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles LA Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2025. Judging the judges: A collection of llm-generated relevance judgements. *arXiv preprint arXiv:2502.13908*.

Thilina C Rajapakse and Maarten de Rijke. 2023. Improving the generalizability of the dense passage retriever using generated datasets. In *European Conference on Information Retrieval*, pages 94–109. Springer.

Navid Rekabsaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. 2021. TripClick: The Log Files of a Large Health Web Search Engine. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2507–2513. Association for Computing Machinery, New York, NY, USA.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan

Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, et al. 2025. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1930–1940.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577*.

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*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. 2024. Promptriever: Instruction-trained retrievers can be prompted like language models. *Preprint*, arXiv:2409.11136.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-time compute for reranking in information retrieval. *arXiv preprint arXiv:2502.18418*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1253–1256, New York, NY, USA. Association for Computing Machinery.

Weijie Yu, Chen Xu, Jun Xu, Liang Pang, Xiaopeng Gao, Xiaozhao Wang, and Ji-Rong Wen. 2020. Wasserstein distance regularized sequence representation for text matching in asymmetrical domains. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2985–2994, Online. Association for Computational Linguistics.

Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1979–1983.

George Zerveas, Navid Rekabsaz, Daniel Cohen, and Carsten Eickhoff. 2022. CODER: An efficient framework for improving retrieval through COntextual Document Embedding Reranking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10644, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

George Zerveas, Navid Rekabsaz, and Carsten Eickhoff. 2023. Enhancing the ranking context of dense retrieval through reciprocal nearest neighbors. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10779–10803, Singapore. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1503–1512.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the best practices of query expansion with large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.

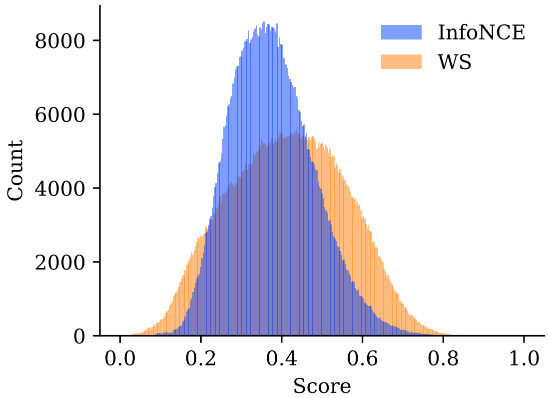


Figure 4: Distribution of the top 100 similarity scores across all Dev queries of MS MARCO dataset by models trained with Wasserstein and InfoNCE losses. The model trained with multiple relevance levels learns a more fine-grained notion of relevance.

nDCG@10	DL19	DL20
BM25 (Yang et al., 2017)	41.7	41.2
Base Contriever (BC)	37.6	36.9
BC + InfoNCE $_{\text{Synth.}}$	48.0	45.0
BC + WS $_{\text{Synth.}}$	52.6	53.4
BC + InfoNCE $_{\text{Real}}$	57.7	55.4
BC + WS $_{\text{Synth. + Real}}$	56.6	54.6

Table 8: Evaluation excluding passages with label 1 (Related), as per the official TREC guidelines

A Discussion

A.1 Search Engine Logs

Although it is feasible to create IR datasets with graduated relevance labels using search engine click logs (Rekabsaz et al., 2021), it comes with significant technical and practical challenges. Technically, extensive search engine logs are only available for popular domains, leaving out niche applications (e.g., climate research). Even for popular domains, given the nature of click-through models, graduated relevance labels are only possible for frequent queries, and rare queries are left with sparse binary annotations (Rekabsaz et al., 2021). Practically, search engine logs are valuable business assets and are only selectively released by large companies, which limits the coverage and quality of the resulting datasets. By contrast, our method uses open-weight LLMs to generate large datasets with graduated relevance labels, which are applicable to many diverse domains and queries while being publicly accessible.

A.2 Pseudo-labeling

As discussed in Section 2, pseudo-labeling requires access to large amounts of existing data, which is not available for rare domains, new applications, and new variants of the retrieval task. Moreover, pseudo-labeling depends on existing retrievers and cross-encoder, which limits the quality of the resulting data. Besides, there is no existing retriever or cross-encoder with acceptable performance for recent tasks like tool retrieval (Qu et al., 2024) or retrieval with reasoning (Shao et al., 2025). Even beyond these issues, selecting a small collection of candidate documents for pseudo-labeling that covers a wide range of relevance levels for each query is difficult. Simple methods like BM25 often fail to retrieve multiple relevant documents with nuanced differences. On the other end of the spectrum, because existing datasets are sparsely annotated, good dense retrievers often select the unannotated positives as labeling candidates and do not capture slightly less relevant but still informative documents.

We randomly select 10,000 queries and measure the similarity of synthetic documents and pseudo-labeling candidates using `e5-mistral-instruct` (Fig. 5a). We observe that synthetic documents cover a wide range of relevance levels, from irrelevant to perfectly relevant. However, documents selected by BM25 are within a much narrower range of relevance levels. Most of them are relevant to the given query but not perfectly relevant. Candidate documents selected by `e5-mistral-instruct` are within an even narrower range of relevance levels and are mostly unannotated positives. Although with `e5-mistral-instruct`, we can ignore the top-ranked documents and choose less relevant documents, it does not significantly improve the diversity of annotations. In Fig. 5b, we use `e5-mistral-instruct` and choose the top 4 documents (equal to the number of synthetic documents) as well as the 90^{th} to 95^{th} documents. It definitely widens the similarity range of candidate documents, but it is still much more limited than synthetic documents.

In Table 11, we show the synthetic documents generated for a sample query as well as the candidate documents selected for pseudo-labeling by BM25 and `e5-mistral-instruct`. For the synthetic documents, the LLM first answers the query directly in the most relevant document and then makes nuanced changes to decrease the relevancy of the answer for each subsequent document. Finally, it generates a totally irrelevant passage for the last relevance level. On the other hand, `e5-mistral-instruct` selects the unannotated positives for pseudo-labeling, which reduces the diversity of annotations (i.e., all candidates will be labeled as "perfectly relevant"). Finally, the documents selected by BM25 do not answer the query at all and are not useful for learning the nuanced differences between multiple relevant documents.

A.3 Decoder-only Retrievers

Although retrievers based on large LLMs, such as E5-Mistral-Instruct (Wang et al., 2023a) and RepLlama (Ma et al., 2024), have achieved significant performance improvements in academic setups, smaller BERT-sized models are still of extreme importance. Inference costs are a major concern for information retrieval. And, even the authors of E5-Mistral-Instruct emphasize the importance of smaller models. Quote from Wang et al. (2023a): "*In comparison to the mainstream BERT-style encoders, the employment of LLMs, such as Mistral7B, for text embeddings results in a significantly increased inference cost.*" and "*With regards to storage cost, our model is comparatively more expensive, with embeddings of 4096 dimensions.*" Many practical applications involve millions, if not billions, of documents. Smaller BERT-sized retrievers are preferred in such cases. Even in academic setups, many recent RAG methods use BERT-based retrievers in their pipeline (Gao et al., 2023), which further emphasizes the importance of smaller models for dense retrieval.

Finally, although we do not use decoder-only retrievers in our work due to practical constraints, we expect that such retrievers would benefit even to a greater extent from our training methodology, as they would be more sensitive to the nuanced training signal offered by our multi-level ranking contexts.

	DL19	DL20	MS Dev	BEIR
Llama 3.3 70B	59.6	59.8	30.2	43.2
All LLMs	60.3	58.3	30.0	43.1

Table 9: Performance of the model trained with synthetic data generated only by Llama 3.3 70B compared to the model trained on the combination of synthetic data generated by Llama 3.3 70B, Qwen 2.5 72B, and Qwen 2.5 32B.

B Impact of Data Size

We run additional experiments to investigate how increasing the number of synthetic passages for each query impacts the performance. We combine the data generated by all three LLMs (i.e., Llama 3.3 70B, Qwen 2.5 72B, and Qwen 2.5 32B) and use it to train the base retriever, similar to our main experiments (Table 9). We find that increasing the size of the data does not have a noticeable impact on performance, which suggests that the quality of the data is more important than its quantity. However, these results should be interpreted with caution since there could be other confounding factors, such as the calibration of ground-truth label values between the three different LLMs. For instance, even if the level 3 document generated by one LLM is less relevant than the level 3 document generated by another LLM for the same query, we use label 3 for both of them in this experiment. Making reliable conclusions about the impact of data size requires extensive experiments that control for this and other confounding factors. We leave such analysis to future work.

C Qualitative Examples

Sample Retrieved Passages Table 12 shows the retrieved passages for a sample query by a model trained on binary ranking contexts with InfoNCE and another model trained on multi-level ranking

nDCG@10	DL19	DL20	MS Dev	BEIR
Condenser	1.1	3.3	0.6	6.3
+ InfoNCE $_{\text{Synth.}}$	58.1	57.0	28.3	37.1
+ WS $_{\text{Synth.}}$	63.3	55.9	29.7	39.3
CoCondenser-Marco	31.1	33.7	14.0	31.0
+ InfoNCE $_{\text{Synth.}}$	59.6	59.0	29.7	39.4
+ WS $_{\text{Synth.}}$	59.7	59.6	30.5	41.3

Table 10: Self-supervised Condenser (Gao and Callan, 2021a) and CoCondenser trained on MS MARCO. The models are further fine-tuned on our synthetic data using InfoNCE with binarized labels or Wasserstein distance with the original 4-level labels.

contexts with Wasserstein distance. Although both models identify the most relevant passage correctly, the model trained on multi-level ranking contexts has a better understanding of relevance and retrieves better passages in other ranks.

D Additional Evaluation

For our main experiments in Section 4, we also measure MRR@100 and Recall@100. As shown in Tables 13 and 14, we observe similar improvements for SyCL compared to other methods.

E Prompting Details

Table 16 shows the exact prompt that we used to generate multi-level ranking contexts for training queries of the MS-MARCO dataset. To create in-context examples, we use the annotations in the TREC DL 2023 split. For each prompt, we randomly sample one query and four passages (one for each relevance level in TREC DL 2023 annotations) and use them as the in-context example. To increase the diversity of the generated passages, for each prompt, we randomly sample the value of `{{num_sentences}}` from `{none,2,5,10,15}` with probabilities `{0.5,0.1,0.2,0.1,0.1}`. Similarly, we randomly sample the value of `{{difficulty_level}}` from `{none, high school, college, PhD}` with probabilities `{0.4,0.2,0.2,0.2}`. For both variables, if the sampled value is `none`, we do not include the corresponding instruction in the prompt.

We also noticed that the LLM has a tendency to provide the exact answer to the query in the very first sentence of the perfectly relevant passage. To avoid such spurious patterns, in 30% of the prompts, we include an additional instruction and explicitly ask the LLM to avoid answering the query in the very first sentence of the perfectly relevant passage.

Direct Binary Data Generation In Section 5, we adapt the short-long matching prompt in Table 8 of Wang et al. (2024) to generate one positive and two negatives for existing queries, which matches the ranking context size in our experiments. Specifically, we use the prompt in Table 15 to directly generate these binary passages.

F Implementation Details

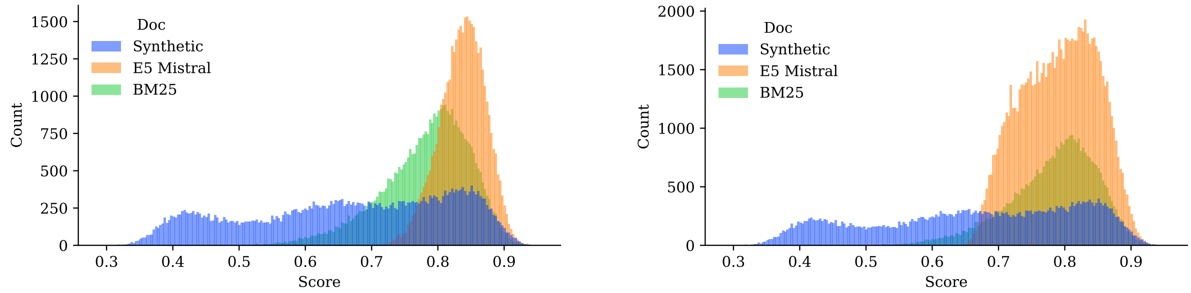
We train our models for only one epoch using the Trainer module in the Huggingface transformers library[2]. For both training and evaluation, we use the maximum length of 256 for both queries and passages. We use a total batch size of 64 across four GPUs (batch size of 16 per device). We set the learning rate to 1e-5, gradient accumulation steps to 4, and warm-up ratio to 0.05. We use the default parameters in version 4.48.0 of the transformers library for all other configurations, e.g., optimizer, learning rate scheduler, etc. Each one of our experiments takes about 2 hours using one machine with four L40s GPUs.

Note that since the original Contriever paper (Izacard et al., 2021) uses a sequence length of 512, our evaluation results are slightly different from what is reported in Izacard et al. (2021).

Data Generation Costs We have generated our data locally over many sessions using different GPU devices, which unfortunately, makes calculating exact cost figures challenging. Here, we approximate the costs based on the number of tokens used for generating data for the 502,000 training queries in MS MARCO. Since our data is very similar to MS MARCO, we use an average length of 128 and 32 tokens for each passage and query, respectively. This is an overestimation, and the actual average length of each passage and query in MS MARCO is 80 and 10 tokens, respectively. For a reasonable approximation, we use the prices of GPT-4o Mini batch API at the time of writing (input: $0.075/1M, output: $0.30/1M), which leads to ~$100 for the cost of API calls.

Note that we use public models that can be deployed on local hardware, which reduces costs. More importantly, we show that we can generate data of comparable quality with smaller 32B LLMs. Inference with a 32B model is drastically cheaper,

[2]https://github.com/huggingface/transformers



(a) (b)

Figure 5: Comparison of the distribution of similarity scores for synthetic documents and candidate documents selected for pseudo-labeling by BM25 and E5-Mistral-Instruct. a) For E5-Mistral-Instruct, we only select the the top-4 mined documents. b) For E5-Mistral-Instruct, we select both the top-4 and the the 90^{th} to 95^{th} mined documents for each query.

which makes synthetic data generation even more appealing.

G Other Models

We repeat our main experiments using Condenser (Gao and Callan, 2021a) and CoCondenser-Marco (Gao and Callan, 2021b) as the base retrievers. Condenser is a BERT model with a slight architectural modification during pre-training that makes the learned representations more suitable for retrieval. CoCondenser-Marco is a Condenser model fine-tuned on the MS MARCO corpus in an *unsupervised manner* (i.e., without using any labels). Since these two models do not perform as well as the Contriever model, we train them for three epochs instead of one and also increase the learning rate to 1e-4. As shown in Table 10, synthetic data significantly improves the base unsupervised model in both cases. Moreover, except for Condenser on the DL20 split of MS MARCO, training using multiple relevance labels leads to better performance compared to contrastive training with binary labels using the InfoNCE loss. Notably, the base Condenser model is only trained with a language modeling objective without any retrieval-specific fine-tuning, which could potentially impact its ability to learn the nuanced differences between multiple levels of relevance. Furthermore, we noticed that Wasserstein loss leads to smaller gradient norms than InfoNCE loss (i.e., smaller updates and thus slower convergence). As a result, we speculate that for lower-quality models or models without contrastive pre-training, the difference between InfoNCE and Wasserstein losses will increase with more training steps.

H Loss Functions

We calculate the similarity between query q and document d as the inner product between their embeddings. Specifically,

$$\mathrm{sim}(d, q) = f_\theta(d) \cdot f_\theta(q)\,,$$

where f is the embedding function parameterized by θ.

InfoNCE We calculate the InfoNCE loss as follows:

$$-\log \frac{\exp(\mathrm{sim}(d^+, q))}{\sum_{d \in D_q} \exp(\mathrm{sim}(d, q))}\,,$$

where d^+ is the positive document, and D_q is the ranking context for query q (i.e., the collection of positive and negative documents for q). Note that for InfoNCE loss, D_q can contain one and only one positive document, and the rest must be negative.

KL Divergence Given the similarity scores between a query and documents in its ranking context, we calculate the KL loss as follows:

$$\mathrm{D}_{\mathrm{KL}}(\sigma(Y) \| \sigma(\hat{Y}))\,,$$

where σ is the softmax function, and $Y \in \mathbb{R}^{|D_q|}$ and $\hat{Y} \in \mathbb{R}^{|D_q|}$ are the ground truth relevance labels and predicted relevance labels (i.e., similarity scores) for documents in the ranking context of query q, respectively.

Wasserstein Distance We use the special case of Wasserstein distance between two multivariate Gaussian distributed inputs $X \sim \mathcal{N}(\mu_x, C_x)$ and $Y \sim \mathcal{N}(\mu_y, C_y)$, where μ and C are the mean and covariance of each distribution, respectively. For

Gaussian distributions, the 2-Wasserstein distance reduces to

$$\mathrm{D}(X, Y) = \|\mu_x - \mu_y\|^2 - \mathrm{tr}(C_x + C_y - 2(C_x C_y)^{\frac{1}{2}}) \,.$$

In our implementation, we calculate the Wasserstein score for the entire batch. Specifically, for each batch, we create matrices $H \in \mathbb{R}^{b \times |D_q|}$ and $\hat{H} \in \mathbb{R}^{b \times |D_q|}$ of shape `(batch size, ranking context size)` and minimize $\mathrm{D}(H, \hat{H})$ during training. Each row of H corresponds to ground truth relevance labels for one query in the batch. Similarly, one row of $\hat{H}$ corresponds to the predicted similarity scores between one query in the batch and documents in its ranking context. We use the fast implementation[3] proposed by Mathiasen and Hvilshøj (2020).

[3] https://gist.github.com/Flunzmas/6e359b118b0730ab403753dcc2a447df

Query: *"border personality disorder symptoms"*

Synthetic Documents	E5 Mistral Candidates	BM25 Candidates
Borderline personality disorder (BPD) is a serious mental illness characterized by pervasive instability in moods, interpersonal relationships, self-image, and behavior. Symptoms of BPD include frantic efforts to avoid real or imagined abandonment, intense interpersonal relationships marked by alternating extremes of idealization and devaluation, and unstable self-image or sense of self. Individuals with BPD may also exhibit impulsive behaviors, such as excessive spending, reckless driving, or risky sex, and have recurring suicidal thoughts or self-mutilating behaviors.	The symptoms of borderline personality disorder include: a recurring pattern of instability in relationships, efforts to avoid abandonment, identity disturbance, impulsivity, emotional instability, and chronic feelings of emptiness, among other symptoms.	Description of Affective personality disorder Affective personality disorder: Related Topics These medical condition or symptom topics may be relevant to medical information for Affective personality disorder: Related Topics. These medical condition or symptom topics may be relevant to medical information for Affective personality. Personality disorder (2 causes) Affective. Affective symptoms. Affective disorder. Personality.
Certain personality disorders, including borderline personality disorder, can have a significant impact on an individual's emotional and psychological well-being. People with these conditions may experience intense emotional dysregulation, leading to mood swings, irritability, and impulsive behaviors. They may also struggle with maintaining stable relationships, due to fear of abandonment or difficulty with emotional intimacy. While the exact causes of these disorders are not fully understood, treatment options such as dialectical behavior therapy and medication can help alleviate symptoms and improve overall functioning.	5 min read. The symptoms of borderline personality disorder include: a recurring pattern of instability in relationships, efforts to avoid abandonment, identity disturbance, impulsivity, emotional instability, and chronic feelings of emptiness, among other symptoms.	Narcissistic Personality Disorder symptoms include a complete and total lack of empathy, along with a highly-exaggerated sense of self-importance.... narcissistic,personality,disorder,treatment,personality disorder treatment,narcissistic disorder symptoms,signs of narcissistic personality disorder,narcissistic personality disorder npd.
Emotional regulation is a critical aspect of mental health, and difficulties in this area can contribute to a range of psychological problems. Research has shown that individuals with mental health conditions, such as depression and anxiety, often struggle with managing their emotions in a healthy and adaptive way. This can lead to a range of negative consequences, including strained relationships, decreased productivity, and increased risk of self-destructive behaviors. By improving emotional regulation skills, individuals can better cope with stress and adversity, leading to improved overall well-being.	Borderline personality disorder (BPD) is a personality disorder that typically includes the following symptoms: 1 Inappropriate or extreme emotional reactions. 2 Highly impulsive behaviors. 3 A history of unstable relationships.	Personality disorder - Symptoms. Signs and symptoms of personality disorders. The different types of personality disorder that might need treatment can be broadly grouped into one of three clusters, called A, B or C. Cluster A personality disorders.
The city of Paris is known for its stunning architecture, rich history, and vibrant cultural scene. Visitors can explore famous landmarks like the Eiffel Tower, Notre-Dame Cathedral, and the Louvre Museum, which houses an impressive collection of art and artifacts from around the world. The city is also famous for its fashion industry, with top designers like Chanel and Dior showcasing their latest creations during Paris Fashion Week. Whether you're interested in history, art, or food, Paris has something to offer everyone.	By Mayo Clinic Staff. Borderline personality disorder affects how you feel about yourself, how you relate to others and how you behave. Signs and symptoms may include: An intense fear of abandonment, even going to extreme measures to avoid real or imagined separation or rejection.	Symptoms. Types of personality disorders are grouped into three clusters, based on similar characteristics and symptoms. Many people with one personality disorder also have signs and symptoms of at least one additional personality disorder.

Table 11: Synthetic documents generated for one query compared to documents selected for pseudo labeling by BM25 and E5-Mistral-Instruct for the same query. The differences between synthetic documents are nuanced, and documents gradually change from perfectly relevant to irrelevant. All the selected candidates by E5-Mistral-Instruct are actually unannotated positives, and the candidates selected by BM25 are not actually relevant to the query or informative.

Query: *"how many ml a day should you drink"*

rank	InfoNCE, binary relevance labels	Wasserstein loss, 4-level relevance labels
1	How many ml of water should you drink in a day? The Institute of Medicine recommends that men drink 3000 ml of water each day and women drink 2100 ml. This equals approximately 13 cups of liquid for men and 9 cups from women. An individual often requires more water to stay hydrated in hot weather or due to strenuous exercise. < continued >.	How many ml of water should you drink in a day? The Institute of Medicine recommends that men drink 3000 ml of water each day and women drink 2100 ml. This equals approximately 13 cups of liquid for men and 9 cups from women. An individual often requires more water to stay hydrated in hot weather or due to strenuous exercise.
2	Since 2,000 mL of fluid are needed daily for normal body functions, first determine how many mL each patient has consumed so far today. Identify which patients need to be encouraged to consume more fluids to meet the 2,000 mL intake standard.	How many ml of water should you drink in a day? The Institute of Medicine recommends that men drink 3000 ml of water each day and women drink 2100 ml. This equals approximately 13 cups of liquid for men and 9 cups from women. An individual often requires more water to stay hydrated in hot weather or due to strenuous exercise. < continued >
3	How many ml of water should you drink in a day? The Institute of Medicine recommends that men drink 3000 ml of water each day and women drink 2100 ml. This equals approximately 13 cups of liquid for men and 9 cups from women. An individual often requires more water to stay hydrated in hot weather or due to strenuous exercise.	The recommendation stated that a person should have 1 ml (about 1/5 of a teaspoon) of water for each calorie he or she consumes. The average diet at the time was approximately 1900 calories, meaning you needed about 64 ounces of water per day. Now the Institute of Medicine sets general guidelines for total water intake.It recommends that women consume a total of 91 ounces (that's about 2.7 liters) per day from all food and beverages combined. For men, it's about 125 ounces a day (or 3.7 liters).Depending on your diet, about 25% of the water you consume comes from your food. Most of us healthy folks get enough water in the foods and liquids we consume. That includes any liquid we drink even caffeinated beverages like soda, coffee and tea.t recommends that women consume a total of 91 ounces (that's about 2.7 liters) per day from all food and beverages combined. For men, it's about 125 ounces a day (or 3.7 liters).
4	How many quarts of water should you drink each day? The recommended minimum amount of water you should have each day is 8 cups, which is equal to 2 quarts.	The recommendation stated that a person should have 1 ml (about 1/5 of a teaspoon) of water for each calorie he or she consumes. The average diet at the time was approximately 1900 calories, meaning you needed about 64 ounces of water per day. Now the Institute of Medicine sets general guidelines for total water intake.It recommends that women consume a total of 91 ounces (that's about 2.7 liters) per day from all food and beverages combined.For men, it's about 125 ounces a day (or 3.7 liters).Depending on your diet, about 25% of the water you consume comes from your food. Most of us healthy folks get enough water in the foods and liquids we consume. That includes any liquid we drink even caffeinated beverages like soda, coffee and tea.or men, it's about 125 ounces a day (or 3.7 liters). Depending on your diet, about 25% of the water you consume comes from your food. Most of us healthy folks get enough water in the foods and liquids we consume. That includes any liquid we drink even caffeinated beverages like soda, coffee and tea.
5	If you are overweight or workout vigorously, this number will increase. And then, if you want to lose weight, you can add 500 ml water to your regular water intake to burn around 23 calories per day that will help you lose upto 5 pounds of weight per year.	The Institute of Medicine recommends an average of 3.7 liters (125 ounces) per day for healthy adult men and 2.7 liters (91 ounces) per day for healthy adult women, allowing adjustments for activity and health levels, climate and elevation, and the amount of water consumed from food and other drinks.
6	Presuming you're awake for approximately 16 hours per day, you'll have to drink between 4.65 and 6.25 fluid ounces per hour. That may seem like a lot, but it isn't much more than four to eight sips per hour (depending on how much you take in).	How much you would need to drink daily isn't clear, I would suggest just a bit more than 1 liter a day instead of the often quoted 2 - 3 liters a day. . Metabolic processes will generate about 300 ml of water a day, your food contains about 800 ml of water daily. The rest of your intake is what you drink.
7	It means your normal urine output per hour should be anywhere between 33.3 and 83.3 ml. If it's not within this range, there's something wrong. However, you need to ensure that you're drinking no less than 2 liters of fluid per day. These numbers may change a bit considering your unique circumstances.	The Institute of Medicine advises that men consume roughly 3.0 liters (about 13 cups) of total beverages a day and women consume 2.2 liters (about 9 cups) of total beverages a day.
8	However, each drinking session of three pints is at least six units, which is more than the safe limit advised for any one day. Another example: a 750 ml bottle of 12% wine contains nine units. If you drink two bottles of 12% wine over a week, that is 18 units.	How much should you drink: It's said we need 8-10 glasses of water a day (8 oz. glasses). That's at least 2 quarts of water. This is just to provide the water we need to wash away the acidity from our bodily functions and remove our own wastes.
9	A 10 ml bottle contains 1000 units There are 100 units in a mL. 1 cc equals 100 units, so to figure how long a 10mL bottle, (1000 units) will last, you divide the number of units you use per day into 1000, and there you have it.Actually it depends on the concentration of the bag of solution you have. 10 ml bottle contains 1000 units There are 100 units in a mL. 1 cc equals 100 units, so to figure how long a 10mL bottle, (1000 units) will last, you divide the number of units you use per day into 1000, and there you have it.	How much water does one person need to drink per day? you should drink at least 7 to 10 average sized glasses of water each day. One average sized glass is about eight ounces. There are 16 ounces in a pint, 2 pints in a quart, and 4 quarts in a gallon, so, mathematically, there are about 128 ounces in a gallon.
10	When I got to work, I filled up my 16-ounce water bottle and drank it through a straw. For some reason, drinking though a straw helped me to drink more because I would take sips without thinking. I had to fill this up six times per day to get the 3 liters. For the first few days, I made a conscious effort to keep up with this, but after day four, I started to write down how many ounces I drank just to keep track.	How many ml of water should you drink in a day? A: The Institute of Medicine recommends that men drink 3000 ml of water each day and women drink 2100 ml. This equals approximately 13 cups of liquid for men ... < continued >

Table 12: Retrieved MS MARCO (real) passages for a sample query by a Contriever trained on synthetic documents using binary labels with InfoNCE (left) and the same model trained on the same documents using multi-level ranking contexts with the Wasserstein distance as a loss, i.e. SyCL (right). SyCL trains models to distribute higher relevance scores over a larger number of documents. Here, only one of these documents is labeled as relevant in the dataset, although, in fact, many are relevant or even near-duplicates; that makes them false negatives. This is a typical case in MS MARCO, confounding training and evaluations that rely on these labels (as MM dev).

MRR@100	**DL19**	**DL20**	MM Dev	FEVER	HotpotQA	FiQA	NQ	Quora	Touche
Base Contriever (BC)	76.0	78.8	17.4	64.3	64.0	31.0	23.1	82.6	38.6
BC + InfoNCE $_{\text{Synth.}}$	84.0	76.9	22.6	65.6	61.9	34.2	29.8	74.8	31.8
BC + WS $_{\text{Synth.}}$	93.8	90.5	26.0	82.6	76.1	35.0	37.9	82.6	41.8
BC + InfoNCE $_{\text{Real}}$	91.3	87.1	29.5	67.9	78.1	36.2	38.4	80.6	30.1
BC + WS $_{\text{Synth. + Real}}$	92.3	87.7	28.1	81.2	78.7	37.4	38.1	82.9	36.9

MRR@100	CQADup Android	Scidocs	Climate FEVER	DBPedia	TREC COVID	Scifact	NFCorpus	ArguAna	**BEIR Avg**
Base Contriever (BC)	38.3	29.0	21.3	59.9	58.0	60.2	51.6	21.6	46.0
BC + InfoNCE $_{\text{Synth.}}$	36.2	28.5	29.5	63.5	49.2	59.2	51.7	18.5	45.3
BC + WS $_{\text{Synth.}}$	39.1	31.2	37.9	73.8	73.5	58.5	52.5	19.5	53.0
BC + InfoNCE $_{\text{Real}}$	38.7	29.8	26.3	70.8	57.2	62.4	51.3	23.3	49.4
BC + WS $_{\text{Synth. + Real}}$	40.7	29.8	35.5	75.0	74.3	64.3	53.2	22.5	53.6

Table 13: Retrieval effectiveness (MRR@100). Base Contriever (BC): self-supervised Contriever model. 'BC +' denotes the fine-tuning setting in terms of **loss function**: InfoNCE / Wasserstein (WS), and **type of data**: real data from the MS MARCO training set with annotated positives and mined hard negatives (Real) / fully synthetic multi-level documents (Synth.) / combination. DL19, DL20, and MM Dev are the TREC DL 2019, TREC DL 2020, and Dev evaluation sets of MS MARCO. Evaluation on the rest of sets is zero-shot. Purple: SyCL, our method.

Recall@100	**DL19**	**DL20**	MM Dev	FEVER	HotpotQA	FiQA	NQ	Quora	Touche
Base Contriever (BC)	41.8	44.6	67.2	93.3	70.5	58.0	77.2	98.7	41.9
BC + InfoNCE $_{\text{Synth.}}$	44.0	47.9	74.1	93.8	66.7	60.0	83.1	97.5	39.7
BC + WS $_{\text{Synth.}}$	44.7	49.4	77.6	95.3	71.9	60.0	86.7	98.8	46.3
BC + InfoNCE $_{\text{Real}}$	48.3	53.1	84.1	93.3	75.7	63.7	90.0	98.8	41.8
BC + WS $_{\text{Synth. + Real}}$	49.0	54.6	82.8	95.1	74.8	64.3	89.5	98.9	44.0

Recall@100	CQADup Android	Scidocs	Climate FEVER	DBPedia	TREC COVID	Scifact	NFCorpus	ArguAna	**BEIR Avg**
Base Contriever (BC)	74.5	36.0	45.6	45.3	3.7	90.4	29.3	94.7	61.4
BC + InfoNCE $_{\text{Synth.}}$	72.1	35.5	51.5	45.0	3.3	92.2	29.2	89.5	61.4
BC + WS $_{\text{Synth.}}$	76.8	36.1	57.3	46.8	8.8	92.8	30.5	94.0	64.4
BC + InfoNCE $_{\text{Real}}$	72.9	36.6	45.3	49.9	3.8	91.1	29.9	96.2	63.5
BC + WS $_{\text{Synth. + Real}}$	75.5	36.5	56.0	50.8	8.4	93.3	31.1	97.1	65.4

Table 14: Retrieval effectiveness (Recall@100). Base Contriever (BC): self-supervised Contriever model. 'BC +' denotes the fine-tuning setting in terms of **loss function**: InfoNCE / Wasserstein (WS), and **type of data**: real data from the MS MARCO training set with annotated positives and mined hard negatives (Real) / fully synthetic multi-level documents (Synth.) / combination. DL19, DL20, and MM Dev are the TREC DL 2019, TREC DL 2020, and Dev evaluation sets of MS MARCO. Evaluation on the rest of sets is zero-shot. Purple: SyCL, our method.

```
# Task

You have been assigned a user query. Your mission is to write one positive passage and two negative
passages for the given query.
- "Positive Passage" is a relevant passage for the user query.
- "Negative Passage" is a passage that only appears relevant to the query.

Please adhere to the following guidelines:
- All passages must be created independent of the query. Avoid copying the query verbatim. It's
acceptable if some parts of the "Positive Passage" are not topically related to the query.
- All passages should be at least num_sentences sentences long.
- The "Negative Passage" contains some useful information, but it should be less useful or comprehensive
compared to the "Positive Passage".
- Do not provide any explanation in any passages on why it is relevant or not relevant to the query.
- The passages require difficulty_level level education to understand.

Do not explain yourself or output anything else. Be creative!
```

Table 15: Our prompt for directly generating binary passages for each query.

Type	Content
System	# Task You are a data engineer whose goal is to generate synthetic passages that teach a ranking system to sort a collection of passages based on how relevant they are to the user’s search query (similar to a web search engine). Given a text query, your mission is to write four different passages, each with a different level of relevance to the given query. Specifically, you should write one passage for each of the following relevancy levels: - "Perfectly relevant passage": a passage that is dedicated to the query and contains the exact answer. - "Highly relevant passage": a passage that has some answer for the query, but the answer may be a bit unclear, or hidden amongst extraneous information. - "Related passage": a passage that seems related to the query but does not answer it. - "Irrelevant passage": a passage that has nothing to do with the query. ## Passage generation instructions - All passages should be about {{num_sentences}} sentences long. - All passages require {{difficulty_level}} level education to understand. - {{The very first sentence of the passage must NOT completely answer the query.}} - Avoid copying the query verbatim. It’s acceptable if some parts of the "Perfectly relevant passage" are not topically related to the query. - How related each passage is to the given query should closely adhere to the corresponding relevancy level. - Passages can be less relevant to a given query for different reasons. For example, they might be less useful, less accurate, less comprehensive, etc. Explore different ways for writing less relevant passages. Be creative! - Do not provide any explanation in any passage on why it is relevant or not relevant to the query. ## Evaluation criteria To double check if you have successfully accomplished the task, you should imagine how a search engine like Google Search would rank the generated passages if you search for the given query. To accomplish the task successfully, a search engine like Google Search should rank your passages in the same order that you generated them. In other words: - the perfectly relevant passage should fully answers the query. - the highly relevant passage should be less relevant to the query than the perfectly relevant passage. - the related passage should be less relevant to the query than the highly relevant passage. - the irrelevant passage should not provide any useful information about the query. Do not explain yourself or output anything else. Be creative!
User	## Query: {{IC_example_query}}
Assistant	[Perfectly relevant passage] {{IC_example_perfectly_relevant_passage}} [Highly relevant passage] {{IC_example_highly_relevant_passage}} [Related passage] {{IC_example_related_passage}} [Irrelevant passage] {{IC_example_irrelevant_passage}}
User	## Query: {{main_query}}

Table 16: Our full prompt template used to generate synthetic multi-level ranking contexts for each query. See Appendix E for more details.