Improving Document Retrieval Coherence for Semantically Equivalent Queries

Stefano Campese
Amazon AGI
University of Trento
campeses@amazon.com

Alessandro Moschitti
Amazon AGI
amosch@amazon.com

Ivano Lauriola
Amazon AGI
lauivano@amazon.com

Abstract

Dense Retrieval (DR) models have proven to be effective for Document Retrieval and Information Grounding tasks. Usually, these models are trained and optimized for improving the relevance of top-ranked documents for a given query. Previous work has shown that popular DR models are sensitive to the query and document lexicon: small variations of it may lead to a significant difference in the set of retrieved documents. In this paper, we propose a variation of the Multi-Negative Ranking loss for training DR that improves the coherence of models in retrieving the same documents with respect to semantically similar queries. The loss penalizes discrepancies between the top-k ranked documents retrieved for diverse but semantically equivalent queries. We conducted extensive experiments on various datasets, MS-MARCO, Natural Questions, BEIR, and TREC DL 19/20. The results show that (i) models optimizes by our loss are subject to lower sensitivity, and, (ii) interestingly, higher accuracy.

1 Introduction

In the recent years, pre-trained Language Models (PLMs) have shown striking performance on a plethora of NLP tasks including, but not limited to, Question Answering, Information Retrieval, Machine Translation, chat-bots, and many more (Min et al., 2023; Wang et al., 2023). One popular and well-studied application of PLMs is Dense Retrieval (DR) (Karpukhin et al., 2020), consisting of dual encoders that create dense vector representations (embeddings) of both queries and documents. Embeddings similarities are then used to retrieve relevant documents from an index.

Recently, DR proven to be an effective solution for both, simple document retrieval applications and Retrieval Augmented Generation (RAG) (Zhao et al., 2024), where a Larger Language Model (LLM) is tasked to produce answers based on retrieved documents. DR models are typically fine-tuned from PLMs to align the embeddings between queries and relevant texts or documents. Previous work has shown improvements through various approaches, e.g.: (i) specialized loss functions (Henderson et al., 2017), (ii) mechanisms to mine meaningful training examples and hard-negatives (Lin et al., 2023), and (iii) labeled data at scale (Nguyen et al., 2016). One potential drawback of DR is their *sensitivity* to the query and document lexicon. Intuitively, this is defined as the *difference* in the output response with respect to changing of the query wording (Chen et al., 2024; Liu et al., 2023a). We note two aspects: First, low query sensitivity is empirically proven to be proportional to high accuracy (Lu et al., 2024; Lauriola et al., 2025). Not being able to answer some variations of the same query corresponds to poor generalization. For instance, a model trained on natural questions may have problems in answering web-like versions of the same queries, or questions with negations (Guo et al., 2025). Second, behavioral studies showed that users start multiple searches with rewritten queries when the initial search output does not contain satisfactory results (Bernard et al., 2007; Jansen et al., 2005), and up to 50% traffic in early retrieval engines may be just reformulations. Recent work suggests that the problem is not solved in modern retrieval engines (Wang et al., 2021b). These aspects may lead to an increase of search cost, as multiple searches require the re-execution of the retrieval pipeline.

Previous work explored various approaches to make the model less sensitive, and thus more *coherent*, including synthetic data generation (Guo et al., 2025; Chaudhary et al., 2024; Meng et al., 2022) and query reformulation (Ma et al., 2023). The former shows that generating lexical variations of annotated queries can improve the generalization of the model. The latter tries to reshape the query to be more aligned to the DR input while preserving the intent. Although query reformulation showed

some benefits, it requires the introduction of a *re-writer* (Ma et al., 2023), typically implemented as an LLM, with a consequent drop in efficiency and increase in cost.

In this work, we focus on analyzing and improving the coherence of DR models, intuitively defined as the ability of a model in retrieving the same set of documents (or the same ranked list) from a given collection (or index) for different lexical variations of the same equivalent input query. Differently from most of previous work, based on query reformulation or simple data augmentation, we inject the coherence into the loss function directly. Specifically, we extend the Multiple Negative Ranking (MNR) loss (Henderson et al., 2017) to (i) penalize dissimilarities of embeddings from lexical variations of the same query and to (ii) optimize for query-document similarity alignment.

To validate the effectiveness of the loss function, we conducted extensive experiments on MS-MARCO, Natural Questions, BEIR, and TREC-DL with multiple PLMs, namely MPNet (Song et al., 2020), ModernBERT (Warner et al., 2024), and MiniLM (Wang et al., 2020a). Our results show that our loss consistently improves the coherence of DR models (and thus reducing the general idea of sensitivity to the input query) measured through Rank Biased Overlap (RBO) (Webber et al., 2010) between documents retrieved from multiple equivalent queries, with an average increase of +15% absolute on MS-MARCO, from 0.43 to 0.58, and +29% on Natural Questions, from 0.38 to 0.67. Beyond coherence, our approach shows an improvement in NDCG of +0.60% MS-MARCO, +1.8% on NQ, +0.5% on 11 BEIR, and +1.4%/0.3% on TREC-DL benchmarks averaged.

2 Related work

Coherence in LLMs Popular LLMs have shown to be very sensitive to the input (Voronov et al., 2024; Mizrahi et al., 2024; Arora et al., 2022; Chatterjee et al., 2024), and the selection of the prompt format plays a crucial role. Lu et al. (2024) demonstrated that coherence can be seen as the opposite of sensitivity, and can be considered as an unsupervised proxy for model performance. In addition, Raina et al. (2024) performed a deep analysis on adversarial robustness of LLMs, showing how to deceive an LLM judge to manipulate the output and predict inflated scores. Except for the input lexicon, the position of words and concepts, e.g.: order of options in multi-choice Q&A (Zheng et al., 2023) or order of in-context examples (Liu et al., 2022; Zhao et al., 2021), also affects the judgment.

Beyond analyzing the phenomenon, Chatterjee et al. (2024) introduced a metric, named POSIX, to measure the prompt sensitivity. Moreover, Rabinovich et al. (2023) introduced PopQA-TP, a curated dataset that extends PopQA (Mallen et al., 2022) with 118,000 paraphrased questions, to benchmark LLMs' sensitivity. Similarly, Lauriola et al. (2025) shower how up to 70B LLMs are unable to provide coherent answers from equivalent queries, and highlighted how coherence optimization is linked to overall accuracy.

Sensitivity in Dense Retrieval Narrowing down the focus on Dense Retrieval (DR) models, previous work showed similar insights. Chen et al. (2024) proposed an unsupervised technique to make the model scores robust towards irrelevant paragraphs in a document. Liu et al. (2023a) studied the sensitivity of models in generative retrieval settings through simple query variations (misspelling, token order modification, rule-based paraphrasing). However, the authors focused on observing the phenomenon and quantify the impact of these simple perturbations. Campos et al. (2023) focused on making the query encoder robust to noise (lemma, stemming, character swap or delete, and some forms of paraphrasing) while keeping the document encoder frozen. Other authors highlighted sensitivity issues from an adversarial viewpoint (Liu et al., 2023b; Wu et al., 2022).

Synthetic query data augmentation has been widely explored as mitigation (Chaudhary et al., 2024; Liang et al., 2020; Meng et al., 2022), showing that generated queries can improve generalization of DR models on some public benchmarks. Based on the same intuition, Guo et al. (2025) used query augmentation targeting improvements on queries with negations. Similarly, Sunkara (2024) mixed query data augmentation with multi-task learning. First, they generated variations of queries through back-translation. Then they apply a multi-task loss that forces embeddings of the same queries to be similar while optimizes for query-document relevancy. However, results did not show improvement over classical DR training.

Query re-writing As possible mitigation of the coherence issue, query rewriting has become a popular solution, aligning input distribution to DR *favourite* query shape (He et al., 2016). For in-

stance, Shi et al. (2024) showed benefits of using multiple re-writing of the query and a subsequent combination of documents retrieved. On the same line, Ma et al. (2023) introduced a trainable rewrite-and-retrieve approach in RAG setting to align the input query to the retriever. However, query re-writing requires the introduction of a query generator component in the retrieval pipeline, typically through LLMs, which may cause higher latency and cost in industrial applications. Other type of re-writing associated with query expansion (Cao et al., 2021; Baek et al., 2025) or conversational Q&A (Christmann et al., 2022; Ye et al., 2023; Qian and Dou, 2022; Yu et al., 2020) are outside the scope of this work. We do not compare against query re-writing approaches as our focus is to train a standalone DR model to improve sensitivity, without external components.

3 Coherence of ranked documents

In this section, we introduce our loss that targets sensitivity improvement by penalizing rank inconsistencies with different variations of the query.

3.1 Preliminaries - query equivalence

Let $\mathcal{Q}$ be a distribution of open-domain info-seeking queries and let $\mathcal{C} \subseteq \mathcal{Q}$ be a subset of queries equivalent each other, that is, $\forall(q_i, q_j) \in \mathcal{C}^2 : q_i \equiv q_j$, where $\equiv$ indicates that two questions are semantically equivalent. In this work, we refer $\mathcal{C}$ as equivalent set or cluster of queries. We consider the equivalence definition introduced by Campese et al. (2023). Two questions (q_i, q_j) are semantically equivalent iff they have the same information-seeking intent and their answers can be interchanged. In other words, $\forall_a : l(q_i, a) \leftrightarrow l(q_j, a)$, where l is a labeling function based on an arbitrary interpretation of correctness. $l(q, a) = 1$ if the answer a is correct for q, 0 otherwise. Although this definition applies to both, single- and multi-answer queries, this study focuses non-subjective queries with well-defined and verifiable answers. When dealing with Q&A systems or generative LMs, the coherence of the models can be easily defined as the semantic similarity of answers responding to queries belonging to the same cluster (Rabinovich et al., 2023).

In this work, we focus on the coherence of DR models, where their sensitivity is given by the ranked list of relevant documents retrieved. Let δ be a DR scoring model that, given a query $q \in \mathcal{Q}$ and a document d from a given collection $\mathcal{D}$, produces a similarity score, that it $\delta : \mathcal{Q} \times \mathcal{D} \rightarrow [0, 1]$. For simplicity, we define the top-k list of documents retrieved by δ from the query q as:

$$\psi_{\delta,\mathcal{D}}(q, k) = [d_{q_1}, d_{q_2}, \ldots, d_{q_k}] \quad s.t.\ \delta(q, d_{q_i}) \geq \delta(q, d_{q_{i+1}})\ \forall d_{q_i} \in \mathcal{D} \tag{1}$$

Based on this definition of top-k retrieved list of documents, the coherence of a ranking model can easily be defined as the average rank-similarity between multiple queries in a cluster, e.g.: $\sigma(\psi_{\delta,\mathcal{D}}(q_i, k), \psi_{\delta,\mathcal{D}}(q_j, k))$, where σ is a given rank-similarity function and $(q_i, q_j) \in \mathcal{C}^2$ are two queries from the same cluster. In this work, we used Rank-Biased Overlap (RBO) (Webber et al., 2010) and Spearman correlation, two established metrics to measure similarities of two ranked lists of items. The higher the rank-similarity between two equivalent queries, the small the sensitivity of the model to the input. Any disparity in the ranks highlights a sensitivity issue.

3.2 Coherence Ranking Loss

Here we introduce Coherence Ranking (CR) loss, a support multi-task loss that, paired with classical Multiple-Negative Ranking (MNR) loss, explicitly targets coherence improvements.

CR loss comprises three main factors: Query Embedding Alignment (QEA), Similarity Margin Consistency (SMC), and query-document relevance implemented through MNR. QEA component simply tries to aligning the embeddings of lexically different queries by penalizing their differences measured through Mean Squared Error (MSE). The second component, SMC, enforces equivalent queries to have the same similarities when compared to the same positive and negative documents. Differently from QEA, which focuses on embeddings alignment, SMC targets alignment in similarity scores. The resulting formulation is:

$$\begin{aligned}\mathcal{L}_{CR}(q, d^+, \mathcal{D}^-, \mathcal{C}) = \\ \lambda_1 \frac{1}{|\mathcal{C}|} \sum_{q_i \in \mathcal{C}} \|\mathbf{q} - \mathbf{q}_i\|_2^2 \\ + \lambda_2 \sum_{q_i \in \mathcal{C}} \sum_{d \in \mathcal{D}^-} \big(m(q, d^+, d) - m(q_i, d^+, d)\big)^2 \\ + \ MNR(q, d^+, \mathcal{D}^-),\end{aligned} \tag{2}$$

where $q \in \mathcal{C}$ is a query from a given cluster, $d^+ \in \mathcal{D}$ is a document relevant (or positive) to

q, $\mathcal{D}^- \subset \mathcal{D}$ is a set of irrelevant (or negative) documents and $m(q, d^+, d)$ expresses the difference between the relevance of the two documents (one positive and one negative) with respect to the query, that is: $m(q, d^+, d) = s(\mathbf{q}, \mathbf{d}^+) - s(\mathbf{q}, \mathbf{d})$, where s is a vector similarity function, here implemented as cosine. We use bold symbols to indicate the embeddings associated with queries and documents.

4 Experiments

We ran various experiments to evaluate CR loss on MS-MARCO (Nguyen et al., 2016) and Natural Questions (NQ) (Kwiatkowski et al., 2019). Results with BEIR and TREC-DL are reported in Appendix D.

MS-MARCO. This is a popular benchmark for IR. It consists of an index of 8.8M passages documents, 495K training queries, 523K positive query-document pairs, and a number of 5 hard negatives per query extracted as described by Wang et al. (2021a). Given that labels of the official test queries are not released, we divided the development set in development and test, with 3490 queries each. For training, we used up to 5 hard-negatives per queries, made available in the official repository.

Natural Questions (NQ). It originally contained 132,803 unique queries, each associated with a Wikipedia page used to extract an answer. We were able to successfully extract hard negatives for 120K queries following the technique described by Wang et al. (2021a), generating 10 different hard negatives per query. We randomly selected 3,000 queries to create our development set. For the test set we use the original split consisting of 3,452 queries and 2,681,468 passage documents.

For each dataset, MS-MARCO and NQ, we used Phi-3 generate up to 10 different lexical variations of original queries. The model is prompted to generate different queries with the same intent and information-seeking need, while varying the writing style. See Appendix A for the full prompt and some examples of generated queries. The queries are paired with positive documents associated with the input, augmenting the training data. A summary of the two datasets is available in Table 1.

In most of our experiments, we considered the following baselines and configurations that are derived from MPNet (Song et al., 2020):

	MS-MARCO	NQ
Queries TRAIN	495260	119554
Queries DEV	3490	3000
Queries TEST	3490	3452
Hard negatives	5	10
Gen. Queries	10	10

Table 1: Datasets statistics.

Public checkpoint - As simplest baseline we consider the public checkpoint continuously pre-trained on various supervised and self-supervised Sentence Text Similarity (STS) tasks including, but not limited to, paraphrasing, question answering, information retrieval, and natural language inference[1].

Fine-Tuning - The public checkpoint is fine-tuned on target training data, that is MS-MARCO or NQ. Training data consists of triplets $\langle q, d^+, \mathcal{D}^- \rangle$, where $q \in \mathcal{Q}$ is a query, d^+ is a relevant document, and $\mathcal{D}^-$ is the set of hard negatives associated with q. Following the established training approach of DRs, MNR loss is employed.

Query Augmentation - The training data is expanded with the equivalent but lexically different queries generated through Phi. For each training triplet $\langle q, d^+, \mathcal{D}^- \rangle$ we consider 10 extra examples $\{\langle q_i, d^+, \mathcal{D}^- \rangle\}_{i=1}^{10}$, where q_i is an equivalent query generated from q.

$\mathcal{L}_{QQ}$ - Generated queries are used to enforce query similarity reasoning, replacing data augmentation. The training mixes query/document and query/generated batches in round-robin fashion (multi-task learning). On each iteration, we apply (i) an optimization step with simple MNR as described in the FT approach; (ii) a second optimization step where we optimize for query similarity, training examples consist of $\langle q_i, q_j \rangle$, $q_i \equiv q_j$. This baselines shows the impact of training the model to learn similarities over different queries, improving the rank indirectly.

$\mathcal{L}_{CR}$ - We used our loss as defined in Section 3.2, Eq. 2, that jointly optimizes over MNR and query-similarity.

Full - We used $\mathcal{L}_{CR}$ and query augmentation.

[1]Public checkpoint available at https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1.

Lexical - We also considered two additional lexical baselines, BM25 and SPLADE++ (Lassance et al., 2024). For MS-MARCO we used the BM25 corpus from Pyserini (Lin et al., 2021), which already includes query expansion[2].

Amongst other publicly available models, we selected MPNet as (i) it has not a prohibitive dimension (100M parameters) that could affect the volume of our experiments and (ii) it showed leading performance compared to models of similar size on various IR benchmarks as reported in the Sentence Transformer framework[3]. However, other models are tested in Section 4.3 to assess generalization of our approach.

Training details For each configuration, we used the validation set to find the best configuration of hyper-parameters, including learning rate $\{5/7 \cdot 10^{-6}, 1/2/3 \cdot 10^{-5}\}$, and batch size $\{2^x\}_{x=4}^{10}$. We used AdamW optimizer and warmup rate of 10% of the total training steps. We set a limit of 15 epochs for training with an early stopping and a patience of 5. The loss coefficients λ_1 and λ_2, which balance the different components of our objective function, were evaluated across $\{0, 0.2, 0.5, 0.8, 1\}$ to determine their optimal values. For models training, we utilized 8 NVIDIA H100 GPUs.

Metrics and evaluation We consider two sets of metrics to evaluate the relevance of top-k retrieved documents and the coherence of the models when answering lexical variations of equivalent questions. To measure document relevance, we used standard IR metrics: P@1, NDCG@10, MRR@10, and MAP@100. To evaluate relevance, we used only test queries from the original split (no generated queries). Regarding the coherence of the models, we fist run the models on all generated queries (10 per each input test query). Then, without accounting for labels, we compared the rank produced by the original query and the ones produced by generated queries. To measure the alignment and average rank-similarity between original and generated queries, we used RBO (Webber et al., 2010) and Spearman metrics. For simplicity, we considered the top-5 ranked items. The rank similarity is averaged across all test queries. The higher the rank correlation, the higher the coherence of the model, i.e. its ability of generating the same rank while prompted with different input variations.

[2]Pyserini MS-MARCO corpus: msmarco-v1-passage.d2q-t5-docvectors

[3]Public leaderboard as January 2025 https://www.sbert.net/docs/sentence_transformer/pretrained_models.html.

4.1 Main results

Table 2 reports the performance, in terms of document relevance (P@1, NDCG@10, MRR@10, MAP@100) and coherence (RBO@5, Spearman@5), of all baselines and our proposed approach as described in Section 4. The table shows multiple key insights: First, the adoption of generated queries (through Phi-3 as described in Appendix A) to teach the model working with different input variations, either in form of data augmentation (see **Q. Augmentation** in the table) or question similarity loss ($\mathcal{L}_{QQ}$), shows inconsistent results. When used in MS-MARCO training, generated queries produced a drop in document relevancy metrics (e.g.: -1.46 and -0.20 NDCG@10 with Query Augmentation and $\mathcal{L}_{QQ}$ respectively). However, the same techniques lead to an improvement in NQ (e.g.: +0.84 and +1.26 NDCG@10). We hypothesize that generated queries are a mixed blessing and this behavior is linked to the volume of the training data. On the one hand, generated queries can space out the data from the test distribution, leading to lower results in MS-MARCO. On the other hand, NQ is smaller and thus additional generated data may have higher importance and contribute to metrics improvement. Our proposed approach ($\mathcal{L}_{CR}$) shows better results on both datasets on all relevance metrics. It is worth to notice that the combination of data augmentation and coherence loss (**Full**) does not show benefits in document relevance, suggesting that our mechanism to train on query variations is superior to simple query augmentation. Regarding the coherence, we observed that generated queries lead to a strong and consistent improvement in both, RBO and Spearman correlations, over simple fine-tuning. This is expected as the models are explicitly trained to align equivalent yet different questions to the same embedding space or to enforce similarities between different queries and the same documents. Although query augmentation is a surprisingly strong baseline for ranking coherence, our proposed approach showed better results.

A final highlight goes to lexical baselines (**BM25**, **SPLADE++**). Not surprisingly, BM25 is the least coherent approach. The technique, entirely based on tokens overlap, produces results that are tailored to the input wording. Differently, SPLADE, thanks to its ability of highlighting the

Model	P@1	NDCG@10	MRR@10	MAP@100	RBO@5	Spearman@5
MS-MARCO						
Public ckpt	21.58	39.88	33.79	34.27	$0.42_{\pm 0.25}$	$0.46_{\pm 0.12}$
FT	$22.82_{\pm 0.11}$	$41.51_{\pm 0.08}$	$35.34_{\pm 0.12}$	$35.68_{\pm 0.11}$	$0.46_{\pm 0.26}$	$0.47_{\pm 0.13}$
+ Q. Augm.	$21.85_{\pm 0.12}$	$40.05_{\pm 0.21}$	$33.96_{\pm 0.41}$	$34.31_{\pm 0.21}$	$0.59_{\pm 0.27}$	$0.54_{\pm 0.17}$
+ $\mathcal{L}_{QQ}$	$22.87_{\pm 0.21}$	$41.31_{\pm 0.10}$	$35.10_{\pm 0.08}$	$35.50_{\pm 0.10}$	$0.51_{\pm 0.27}$	$0.49_{\pm 0.15}$
+ $\mathcal{L}_{CR}$	$\mathbf{23.01}_{\pm 0.10}$	$\mathbf{41.98}_{\pm 0.17}$	$\mathbf{35.73}_{\pm 0.16}$	$\mathbf{35.70}_{\pm 0.13}$	$0.60_{\pm 0.26}$	$0.53_{\pm 0.17}$
Full	22.46	41.43	34.71	35.18	$\mathbf{0.63}_{\pm 0.26}$	$\mathbf{0.55}_{\pm 0.18}$
BM25	16.74	33.19	27.13	27.85	$0.22_{\pm 0.24}$	$0.45_{\pm 0.11}$
SPLADE++	21.74	40.08	33.72	34.35	$0.46_{\pm 0.28}$	$0.49_{\pm 0.15}$
Natural Questions						
Public ckpt	30.71	46.53	42.59	40.79	$0.57_{\pm 0.22}$	$0.49_{\pm 0.15}$
FT	$38.16_{\pm 0.17}$	$52.16_{\pm 0.13}$	$49.50_{\pm 0.17}$	$47.50_{\pm 0.18}$	$0.54_{\pm 0.23}$	$0.49_{\pm 0.16}$
+ Q. Augm.	$38.57_{\pm 0.11}$	$53.0_{\pm 0.01}$	$49.89_{\pm 0.08}$	$47.66_{\pm 0.16}$	$0.66_{\pm 0.23}$	$0.54_{\pm 0.19}$
+ $\mathcal{L}_{QQ}$	$38.84_{\pm 0.04}$	$53.42_{\pm 0.07}$	$50.23_{\pm 0.08}$	$48.25_{\pm 0.10}$	$0.59_{\pm 0.23}$	$0.51_{\pm 0.17}$
+ $\mathcal{L}_{CR}$	$\mathbf{39.49}_{\pm 0.11}$	$\mathbf{53.85}_{\pm 0.08}$	$\mathbf{50.65}_{\pm 0.09}$	$\mathbf{48.56}_{\pm 0.04}$	$0.70_{\pm 0.22}$	$0.55_{\pm 0.19}$
Full	39.36	53.73	50.50	48.29	$\mathbf{0.71}_{\pm 0.21}$	$\mathbf{0.57}_{\pm 0.20}$
BM25	16.48	30.55	26.34	25.86	$0.40_{\pm 0.27}$	$0.49_{\pm 0.15}$
SPLADE++	29.66	44.89	41.11	39.43	$0.65_{\pm 0.23}$	$0.54_{\pm 0.18}$

Table 2: Results on MS-MARCO and NQ. Best results are highlighted in bold. RBO and Spearman measure the rank-correlation, and thus the coherence of the models. Results are averaged across 5 different runs.

most relevant tokens and entities, showed better coherence, comparable to dense retrieval baselines.

Other experiments comparing our approach against reformulation strategies are described in Appendix E.

4.2 Ablation study on loss components

As described in Section 3.2, our proposed loss comprises two components. The first penalizes embedding misalignment between different variations of the same query, enforcing the embeddings to be query-shape agnostic. The second acts on the margins, and enforces equivalent queries to have the same distance with positive and negative documents. The combination of these two components led to the improvement showed in the previous results. Note that our loss does not replace MNR, it extends it through an additional penalty factor. Table 3 shows document relevance and ranking coherence while using individual components of the loss in addition to standard MNR. Results highlight that the combination of query embeddings alignment and margin consistency is the key aspect, and individual components do not produce the same improvement.

4.3 Models generalization study

All previous experiments were based on MPNet due to its performance on various IR benchmarks compared to models of similar size (approx 100M parameters). To stress the generalization of the proposed loss, we tested the latter on other two popular transformer models: MiniLM-v2-12L and ModernBERT-base. MiniLM is a efficient yet effective solution for dense retrieval. It consists of 33M learnable parameters only. We considered the checkpoint pre-trained for STS[4]. ModernBERT is a recent model designed for long sequences. We considered the base version consisting of 133M parameters[5]. Given that ModernBERT was simply trained with MLM objective, we continuously

Loss	P@1	NDCG@10	RBO@5
MS-MARCO			
$\mathcal{L}_{QEA}$	22.78	41.26	$0.20_{\pm 0.16}$
$\mathcal{L}_{SMC}$	22.81	41.51	$0.22_{\pm 0.18}$
$\mathcal{L}_{CR}$	**23.01**	**41.57**	$\mathbf{0.34}_{\pm 0.24}$
Natural Questions			
$\mathcal{L}_{QEA}$	38.12	51.63	$0.66_{\pm 0.23}$
$\mathcal{L}_{SMC}$	38.88	53.22	$0.57_{\pm 0.23}$
$\mathcal{L}_{CR}$	**39.54**	**53.92**	$\mathbf{0.70}_{\pm 0.22}$

Table 3: Ablation study on loss components: Query Embedding Alignment and Similarity Margin Consistency. RBO measures ranking consistency.

[4] sentence-transformers/all-MiniLM-L12-v2.
[5] answerdotai/ModernBERT-base.

Configuration	MiniLM-v2-12L			ModernBERT-base		
	P@1	NDCG@10	RBO@5	P@1	NDCG@10	RBO@5
MS-MARCO						
Public ckpt	21.6	39.1	$0.39_{\pm 0.24}$	15.0	31.0	$0.40_{\pm 0.25}$
FT	22.6	40.5	$0.44_{\pm 0.26}$	22.8	41.6	$0.39_{\pm 0.25}$
+ Q. Augm.	22.7	40.4	$0.55_{\pm 0.27}$	21.7	39.9	$0.56_{\pm 0.26}$
+ $\mathcal{L}_{QQ}$	22.8	40.5	$\mathbf{0.57}_{\pm 0.27}$	21.9	40.6	$0.49_{\pm 0.26}$
+ $\mathcal{L}_{CR}$	**23.3**	**41.1**	$\mathbf{0.57}_{\pm 0.27}$	**23.0**	**41.9**	$\mathbf{0.56}_{\pm 0.26}$
Natural Questions						
Public ckpt	26.3	41.4	$0.53_{\pm 0.23}$	21.8	37.6	$0.58_{\pm 0.23}$
FT	34.8	48.3	$0.46_{\pm 0.23}$	36.6	50.4	$0.15_{\pm 0.19}$
+ Q. Augm.	35.4	48.1	$0.61_{\pm 0.25}$	35.9	50.2	$0.61_{\pm 0.23}$
+ $\mathcal{L}_{QQ}$	35.4	48.7	$0.44_{\pm 0.24}$	36.8	51.0	$0.38_{\pm 0.24}$
+ $\mathcal{L}_{CR}$	**36.1**	**49.2**	$\mathbf{0.65}_{\pm 0.22}$	**37.2**	**51.1**	$\mathbf{0.65}_{\pm 0.22}$

Table 4: Document relevance and coherence of MiniLM and ModernBERT. RBO measures the coherence.

trained the checkpoint on 1.5B text-similarity pairs, following the same STS training applied to MPNet and MiniLM. Details of the training are available in Appendix B.

Results for a restricted set of configurations are showed in Table 4. Remarkably, both models show the same trend previously observed with MPNet. Our proposed loss improves both, the document relevance and the rank coherence, on both datasets. These results suggest how our loss generalizes over multiple models and is not tailored to a specific solution. The effect of our STS pretraining in ModernBERT is further analyzed in Appendix C.

4.4 Retrieve and Rank evaluation

DR is often a component of a more complex Question Answering or Chat pipeline. Typically, retrieve and re-rank or retrieve and generate solutions are adopted, where the top-k documents selected by a dense retrieval are further re-ranked or used as part of LLM grounding to generate an answer. Although the order of documents as input of LLM is important, as discussed in Section 2, the same becomes irrelevant in retrieve and re-rank pipelines as document re-rankers typically produce scores to each document that do not depend on the retrieval position. As long as the retrieval model can retrieve the same set of documents from different lexical variations of the input, then a document re-ranker can potentially select the same content.

To explore further this aspect, we simulated a retrieve and re-rank application where a document ranking cross-encoder takes the top-50 documents selected by a DR model, re-ranks them, and selects the most relevant one. Let $\psi_{\delta,\mathcal{D}}(q,k)$ (see Eq. 1) be the set of top-k documents (in our experiment, k=50) retrieved by a given retrieval model from a test query q of a certain cluster $\mathcal{C}$. Let $d^* \in \psi_{\delta,\mathcal{D}}(q,k)$ be the document selected by the re-ranker. We define as *re-ranking opportunity* the probability of d^* to appear in the top-k documents retrieved from any other lexical variation of the input query belonging to the same cluster $\mathcal{C}$: $opportunity(q) = \frac{1}{|\mathcal{C}|}\sum_{q_i \in \mathcal{C}} \mathbf{1}_{\psi_{\delta,\mathcal{D}}(q_i,k)}(d^*)$, where $\mathbf{1}$ is the indicator function. Given the best selection from the re-ranker, the re-ranking opportunity measures the likelihood that the same selected document would be made available by the retrieval while prompted with different equivalent questions. In this sense, the reranker has the same opportunity of selecting the same or a better document. The higher the opportunity the lower the possibility of dropping the highest re-ranked document due to a sensitivity issue. Table 5 shows the re-ranking opportunity on MS-MARCO and NQ while using a state-of-the-art document re-ranker[6].

The re-ranking opportunity showed in the table aligns to other results. Our proposed loss makes the model more coherent beyond simple retrieval. All retriever tested, beyond simple document relevancy, have higher chances to retrieve the best selection from the re-ranker, regardless the shape of the query in input. Compared to simple Fine-Tuning, our losses increases the opportunity by 9.3% on average (8.1% if we exclude ModernBERT without STS training). Note that this ex-

[6] https://huggingface.co/BAAI/bge-reranker-large

Configuration	MPNet	MiniLM	Mod. BERT	M.B. w/o STS
MS-MARCO				
Public ckpt	75.7	73.4	74.9	-
FT	79.7	78.4	75.8	71.0
+ Q. Augm.	85.9	83.7	84.5	83.1
+ $\mathcal{L}_{QQ}$	82.4	80.9	83.0	80.2
+ $\mathcal{L}_{CR}$	**87.0**	**85.7**	**85.5**	**84.0**
BM25	59.4			
SPLADE++	77.7			
Natural Questions				
Public ckpt	59.5	31.9	59.3	-
FT	58.9	52.6	11.8	7.8
+ Q. Augm.	67.5	63.6	59.4	50.5
+ $\mathcal{L}_{QQ}$	55.4	66.0	23.7	25.1
+ $\mathcal{L}_{CR}$	**70.9**	**70.4**	65.8	54.6
BM25	63.2			
SPLADE++	67.5			

Table 5: Re-ranking opportunity, how many times the best re-ranked document is retrieved in the top-50 documents from different variations of the query. BGE model was used as re-ranker (cross-encoder).

periment does not indicate whether a new/different top-ranked document is relevant or not. Here, we just highlight that the new top-1 document that the retrieve and re-rank pipeline would select with different variations of the input is different, but not necessarily worse or better.

Other findings on a simple retrieve & generate application are discussed in Appendix F.

4.5 Retrieval complexity

We hypothesize that coherent models are particularly valuable for queries where multiple documents share similar relevance scores. To investigate this, we focused our analysis on original queries (non-generated) from MS-MARCO and NQ where the difference in retrieval scores between the top-1 and 50th ranked document was less than 0.1. Such cases represent complex information needs where the retrieval task becomes particularly challenging, as multiple documents exhibit comparable relevance to the query with minimal score differences. For instance, in MS-MARCO, we observed this phenomenon with queries like *"What constitutional amendment granted American women suffrage?"*, *"Can you describe the gallbladder's position in the human anatomy?"* and *"What is the specific location for viewing the total solar eclipse?"*. Similarly, in NQ, queries such as *"where does the great outdoors movie take place"* and *"who was the declaration of independence written for"* demonstrated this characteristic. In these cases, multiple documents received nearly identical relevance scores, making the final ranking highly sensitive to small score variations. This underscores the importance of maintaining coherence in the ranking process, as minimal differences in retrieval scores can significantly impact the final document order.

Configuration	MS-MARCO	NQ
Public ckpt	$0.16_{\pm 0.14}$	$0.41_{\pm 0.21}$
FT	$0.17_{\pm 0.14}$	$0.25_{\pm 0.17}$
+ Gen. Qs	$0.32_{\pm 0.23}$	$0.43_{\pm 0.24}$
+ $\mathcal{L}_{QQ}$	$0.24_{\pm 0.18}$	$0.30_{\pm 0.20}$
+ $\mathcal{L}_{CR}$	$0.34_{\pm 0.24}$	$0.49_{\pm 0.23}$
Full	$0.38_{\pm 0.25}$	$0.52_{\pm 0.24}$
BM25	$0.07_{\pm 0.14}$	$0.36_{\pm 0.27}$
SPLADE++	$0.23_{\pm 0.21}$	$0.48_{\pm 0.26}$

Table 6: RBO@5 (coherence) on a subset "most complex" queries, i.e. queries where the retrieval score of the top-1 and the 50-th document is differs less than 0.1.

Table 6 shows the results of this evaluation. As expected, the coherence measured through RBO is generally much lower compared to the full set (see Table 2), corroborating our conjecture on the retrieval complexity. Our proposed loss has a drastic contribution, especially on MS-MARCO, improving coherence from 0.16 to 0.38 (+138% relative).

4.6 Examples

Table 7 contains two examples showing how out models affect the coherence. Each example consists of two equivalent queries (indicated as Q_1 and Q_2) and the two associated top-1 retrieved documents (D_1 and D_2) using MPNet with standard fine-tuning. The retrieval is incoherent as the model returns two different documents for two semantically equivalent questions. Most importantly, one of the documents is not or weakly relevant. In contrast, the CR-trained model consistently retrieves D_1 for both queries, highlighting higher coherence and final relevance.

5 Conclusions

This work analyzes the ranking-coherence of Dense Retrieval (DR) models, that is their ability of retrieving the same content when prompted with dif-

Example 1	
Q_1	What is the average lifespan of a flea?
Q_2	Can you explain the typical duration it takes for a flea to complete its life cycle?
D_1	How long is the life span of a flea? 30-90 Days (Average). A flea might live a year and a half under ideal conditions. These include the right temperature, food supply, and humidity. Generally speaking, though, an adult flea only lives for 2 or 3 months. Without a host for food, a flea's life might be as short as a few days. But with ample food supply, the adult flea will often live up to 100 days.
D_2	When you initiated treatment, it can be assumed that eggs were laid earlier that day. It takes around 2 days for those eggs to hatch, 7 days for the larvae to pupate, and another 7 days until the adult stage is reached
Example 2	
Q_1	What mechanism allows some shark species to retain warmth internally?
Q_2	How does a select group of sharks maintain a higher body temperature than their surroundings?
D_1	White sharks have a unique system called a "counter current heat exchange", which keeps their body tempreture +/- 7C above the surrounding water temperature. All sharks have an incredibly unique system on the tip of their nose called the "ampillae of Lorenzini". These are small pores filled with a gel that transmits the electrical currents in the water to the sharks brain so that it can assess its environment.
D_2	But the advantages of endothermy are costly. To maintain a warm body in cold water, a mackerel shark must burn fuel like a blast furnace. A warm-bodied shark may need more than ten times as much food as a cold-bodied shark of the same size.

Table 7: Examples of coherence. The simple FT model returns document D_i when prompted Q_i. Differently, our coherent model returns D_1 for both variations.

ferent lexical variations of the input query. Our experiments show that classical FT based on popular losses and hard-negatives mining leads to poor coherence. As countermeasure, simple data augmentation or multitask training (query-query similarity and query-document alignment) have proven to increase coherence while keeping comparable accuracy. On top of that, our loss function, which jointly (i) penalizes embeddings distance between equivalent queries and (ii) enforces margin between different queries and the same positive/negative documents to be the same, further improves both, accuracy and coherence. Our results, conducted on multiple benchmarks by using different models indicates high generalization.

6 Limitations

The main focus of this work is the coherence of DR models. However, DRs are just a component of state-of-the-art pipelines based on retrieval (typically lexical+dense) and LLMs to generate answers. How DR coherence affects the entire pipeline is not deeply explored in this work. Experiments in Section 4.4 show early evidence on how a state-of-the-art document re-ranker may take benefits from a more coherent DR. However, coherence of the re-ranker itself is outside the scope of this work. Regarding LLMs' coherence, related work (Lauriola et al., 2025) showed that popular models are poorly coherent, and the input query shape heavily affects the final result. Based on these premises, an exhaustive evaluation of an end-to-end pipeline requires different work outside the scope of this paper.

The improvement of document relevancy may seem limited: (i) +0.14 P@1 and +0.47 NDCG@10 on MS-MARCO, (ii) +0.65 P@1 and +0.43 NDCG@10, (iii) +0.48 NDCG@10 on BEIR (average across IR tasks), and +0.68/+0.21 NDCG@10 on TREC-DL (Appendix D) from the best baseline. As discussed before, one main motivation behind coherence optimization is based on previous work evidence, where more coherent models are showed to improve relevancy by *recovering* errors from *unfavorable* input shape. However, although we observed a significant improvement in ranking overlap, the same improvement is not directly translated into relevance. It is worth to notice that the desired outcome from this work is not a accuracy improvement but producing models with higher coherence. As mitigation, we would like to highlight that all experiments conducted on multiple datasets (MS-MARCO, NQ, 11 BEIR

benchmarks, and TREC-DL) with different models (MPNet, MiniLM, ModernBERT with and without STS training) are aligned and show similar trends.

Finally, our results in Section 4.5 suggest that coherence may gain importance in scenario where there are many similar documents, where small input differences can cause a drastic change in the retrieval score, and thus the final rank. This evidence raises the question, how does coherence impact real-world applications, based on web indexes with Billions of documents?

References

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*.

Ingeol Baek, Jimin Lee, Joonho Yang, and Hwanhee Lee. 2025. Crafting the path: Robust query rewriting for information retrieval. *IEEE Access*.

J Jansen Bernard, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a session on web search engines: Research articles. *Journal of the American Society for Information Science and Technology*, 58(6):862–871.

Stefano Campese, Ivano Lauriola, and Alessandro Moschitti. 2023. QUADRo: Dataset and models for QUestion-answer database retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15573–15587, Singapore. Association for Computational Linguistics.

Daniel Campos, ChengXiang Zhai, and Alessandro Magnani. 2023. Noise-robust dense retrieval via contrastive alignment post training. *arXiv preprint arXiv:2304.03401*.

Kaibo Cao, Chunyang Chen, Sebastian Baltes, Christoph Treude, and Xiang Chen. 2021. Automated query reformulation for efficient search based on query logs from stack overflow. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1273–1285. IEEE.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Aditi Chaudhary, Karthik Raman, and Michael Bendersky. 2024. It's all relative! – a synthetic query generation approach for improving zero-shot relevance prediction. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1645–1664, Mexico City, Mexico. Association for Computational Linguistics.

Haitian Chen, Qingyao Ai, Xiao Wang, Yiqun Liu, Fen Lin, and Qin Liu. 2024. Unsupervised dense retrieval with conterfactual contrastive learning. *arXiv preprint arXiv:2412.20756*.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational question answering on heterogeneous sources. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 144–154.

Mengtian Guo, Mutasem Al-Darabsah, Choon Hui Teo, Jonathan May, Tarun Agarwal, and Rahul Bhagat. 2025. Learning to rewrite negation queries in product search. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 575–582, Abu Dhabi, UAE. Association for Computational Linguistics.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task. *arXiv preprint*.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1443–1452.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

Bernard J Jansen, Amanda Spink, and Jan Pedersen. 2005. A temporal comparison of altavista web searching. *Journal of the American Society for information Science and Technology*, 56(6):559–570.

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.

Ivica Kostric and Krisztian Balog. 2024. A surprisingly simple yet effective multi-query rewriting method for conversational passage retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2271–2275.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *arXiv preprint arXiv:2403.06789*.

Ivano Lauriola, Stefano Campese, and Alessandro Moschitti. 2025. Analyzing and improving coherence of large language models in question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11740–11755, Albuquerque, New Mexico. Association for Computational Linguistics.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023a. On the robustness of generative retrieval models: An out-of-distribution perspective. *arXiv preprint arXiv:2306.12756*.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023b. Black-box adversarial attacks against dense retrieval models: A multi-view contrastive learning method. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1647–1656.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus. *Preprint*, arXiv:1911.02782.

Sheng Lu, Hendrik Schuff, and Iryna Gurevych. 2024. How are prompts different in terms of sensitivity? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5833–5856, Mexico City, Mexico. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. 2022. Augtriever: Unsupervised dense retrieval by scalable data augmentation. *arXiv preprint arXiv:2212.08841*.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

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

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Hongjin Qian and Zhicheng Dou. 2022. Explicit query rewriting for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4725–4737, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby Tavor. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 138–154, Singapore. Association for Computational Linguistics.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems. *arXiv preprint arXiv:2407.10670*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Praveena Sunkara. 2024. Enhancing question answering systems with rephrasing strategies: A study on bert sensitivity and refinement techniques.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6287–6310, Bangkok, Thailand. Association for Computational Linguistics.

Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. 2023. Pre-trained language models and their applications. *Engineering*, 25:51–65.

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

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Xiao Wang, Craig Macdonald, and Iadh Ounis. 2020b. Deep reinforced query reformulation for information retrieval. *arXiv preprint arXiv:2007.07987*.

Yaxuan Wang, Hanqing Lu, Yunwen Xu, Rahul Goutam, Yiwei Song, and Bing Yin. 2021b. Queen: Neural query rewriting in e-commerce.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are neural ranking models robust? *ACM Transactions on Information Systems*, 41(2):1–36.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore. Association for Computational Linguistics.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1933–1936.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

A Queries generation

To generate equivalent queries, we uses Phi-3[7], 3.8B parameters. For each query in the MS-MARCO and NQ datasets, we generated 10 semantically equivalent reformulations, maintaining

[7] microsoft/Phi-3-mini-128k-instruct

the original intent while introducing linguistic diversity.

The generation process utilized the following prompt:

```
You are a powerful question rephraser and question generation system.
Given a question coming from the MSMARCO dataset, your task is to generate 10 EQUIVALENT questions using different styles coming from other different datasets.
Two questions are defined EQUIVALENT, is they (i) are asking for exact same thing even if they contain a very different wording, and (ii) they require the same answer.

You can use the following dataset styles:
     - SQuAD-style: Starts with an introductory phrase and focuses on a specific piece of information.
     - MS-MARCO-style: Framed as a request for information, with a more open-ended tone.
     - DuoRC-style: Asks about typical or common symptoms/indicators, using "if" to set up the context.
     - HotpotQA-style: Combines a request for key indicators with a follow-up on how to identify them.
   - NQ-style question: Concise and direct, focused on a specific piece of information. Typically starts with question words like "what", "who", "where", etc.
     - TriviaQA-style question: More open-ended and sometimes requires more detailed or nuanced answers. May include additional context or keywords to guide the response.
     - WebQA-style question: Framed as a request for a list or set of information related to the topic. Often starts with phrases like "Can you provide...", "List the...", or "Identify the..."

Your task is to produce a well formatted and parsable JSON containing the EQUIVALENT questions.
The produced output must be EXACTLY AS FOLLOWS:
```\{
    "original_question": \$INPUT_QUESTION,
    "equivalent_questions":[
         \{'question': \$EQUIVALENT_QUESTION_1,
'style': \$EQUIVALENT_QUESTION_STYLE_1\},
         \{'question': \$EQUIVALENT_QUESTION_2,
'style': \$EQUIVALENT_QUESTION_STYLE_2\},
        \dots,
        \{'question': \$EQUIVALENT_QUESTION_10,
'style': \$EQUIVALENT_QUESTION_STYLE_10\}
    ],
\}```

Where the \$EQUIVALENT_QUESTION_NTH and \$EQUIVALENT_QUESTION_STYLE_NTH are the generated question and the used style.

To produce the JSON you MUST respect the following rules:
+ The generated questions should be short and concise when possible.
+ Remember: two questions are equivalent if (i) they are asking for exact same thing, and (ii) they require the same answer.
+ Remember: each generated question must follow a different style.
+ Remember: the output must be a valid JSON ready to be used without further post-processing.

Here you can find an example:
INPUT_QUESTION:
symptoms of a dying mouse

OUTPUT JSON:
\{
     "original_question": "symptoms of a dying mouse",
    "equivalent_questions":[
       {"question":"What are the typical symptoms that indicate a mouse is dying?", "style":"NQ"},
      {"question":"Identify the most common signs that a mouse is approaching the end of its life.", "style":"TriviaQA"},
       {"question":"Can you provide a list of the primary indicators that a mouse is in the process of dying?", "style":"WebQA"},
       {"question":"According to medical experts, what are the primary symptoms that indicate a mouse is nearing the end of its life?", "style":"SQuAD"},
      {"question":"I need to know the most common signs that a mouse is dying. Can you provide me with that information?", "style":"MS-MARCO"},
        {"question":"What are the key indicators that a mouse is in the process of dying, and how can these be identified?", "style":"HotpotQA"},
        {"question":"If a mouse is showing signs of dying, what are the typical symptoms that would be observed?", "style":"DuoRC"},
       {"question":"How does the appearance of a mouse's coat change when it's approaching death?", "style":"NQ"},
        {"question":"What changes in eating and drinking habits suggest a mouse is near death?", "style":"NQ"},
        {"question":"As a mouse approaches death, it may show this sign related to body temperature. What is it?", "style":"TriviaQA"}
    ]
\}

Remember, just return the JSON, no additional text.

Here is the input:
{question}

Please provide your JSON output
```

We validated our prompt by manually evaluating the correcteness and the equivalence of 100 random questions and their 10 generated variations. The analysis show an accuracy of 100% of the generative model. We reported some examples of the generated queries for both MSMARCO and NQ in Table 8.

## B STS training - ModernBERT

Building upon from a public checkpoint of ModernBERT, 140M parameters, we performed extensive additional pre-training using diverse datasets focused on semantic text matching tasks. The training data encompassed various tasks including text
```

MS-MARCO
Q: *What is the typical function of simple epithelium*
G1: Could you explain the main function of simple epithelium?
G2: What role does simple epithelium play in the body?
G3: simple epithelium purpose
Q: *what is federal prevailing wage*
G1: Can you explain the concept of federal prevailing wage?
G2: federal prevailing wage definition
G3: What does federal prevailing wage refer to?
G4: Can you explain what "federal prevailing wage" is?
Natural Questions
Q: *when did the san francisco giants win their first world series*
G1: When did the San Francisco Giants first win the World Series?
G2: In what year did the San Francisco Giants first win the World Series?
G3: ancisco Giants first win the World Series? When?
Q: *when is shameless us season 8 coming out*
G1: When is the eighth season of Shameless US scheduled to air?
G2: If I'm looking for the premiere date of Shameless US Season 8, when should I expect it?
G3: When will Shameless US Season 8 be available for viewing?

Table 8: Examples of original queries and their generated variations using Phi-3.

similarity detection, answer matching, document understanding, and content abstraction. The training leveraged multiple large-scale datasets such as The Semantic Scholar Open Research Corpus (Lo et al., 2020), PAQ (Lewis et al., 2021), AmazonQA (Gupta et al., 2019), WikiHow (Koupaee and Wang, 2018), and others. Overall, these resources contain more than $\approx$ 1.5B semantically related text pairs.

Our training approach focused on semantic similarity learning, following established practices in dense retrieval training (Reimers and Gurevych, 2019). The model was trained to distinguish between semantically related and unrelated text pairs. We implemented FP16 precision training with a MultipleNegativesRanking loss function, employing a learning rate of 2e-5. The training process utilized a batch size of 2048 samples, with input sequences capped at 128 tokens. We used 8x Nvidia H100 GPUs.

C Effect of STS pre-training

In this section, we investigate the impact of semantic similarity pre-training on ModernBERT's performance. This comparison allows to understand how semantic similarity knowledge acquired during STS pre-training affects both relevance ranking and coherence of the rank.

Results in Table 9 demonstrate clear advantages of using pre-trained ModernBERT-base compared to its non-pre-trained counterpart across both MS-MARCO and NQ datasets. On MS-MARCO, the pre-trained model with standard fine-tuning (FT) achieved notable improvements in all metrics, with P@1 increasing from 20.2% to 22.8%, NDCG@10 from 37.5 to 41.6, and RBO@5 from 0.33 to 0.39. When applying our proposed $\mathcal{L}_{CR}$ loss, the pre-trained model maintained its superior performance, showing further marginal improvements across all metrics (P@1: 23.0%, NDCG@10: 41.9, RBO@5: 0.56).

The advantages of pre-training are even more pronounced on the Natural Questions dataset, where the pre-trained model demonstrated substantial gains in effectiveness. With standard fine-tuning, pre-training improved P@1 by approximately 11 percentage points (from 25.7% to 36.6%) and NDCG@10 by 13.5 points (from 36.9 to 50.4). The addition of $\mathcal{L}_{CR}$ loss further enhanced these results, with the pre-trained model achieving the best overall performance (P@1: 37.2%, NDCG@10: 51.1, RBO@5: 0.65). Notably, RBO@5 showed substantial improvement with pre-training, particularly when combined with $\mathcal{L}_{CR}$, suggesting that pre-training helps the model develop more consistent and coherent ranking behavior.

D BEIR and TREC-DL evaluation

To assess the generalization capabilities of our model and its performance across diverse domains, we conducted extensive experiments using TREC-DL '19[8] and '20[9] benchmarks, and 11 datasets

[8] https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019

[9] https://microsoft.github.io/msmarco/TREC-Deep-Learning-2020.html

Configuration	ModernBERT-base			Pre-trained ModernBERT-base		
	P@1	NDCG@10	RBO@5	P@1	NDCG@10	RBO@5
MS-MARCO						
FT	20.2	37.5	$0.33_{\pm 0.23}$	22.8	41.6	$0.39_{\pm 0.25}$
+ $\mathcal{L}_{CR}$	20.5	37.83	$0.52_{\pm 0.25}$	**23.0**	**41.9**	$\mathbf{0.56}_{\pm 0.26}$
Natural Questions						
FT	25.7	36.9	$0.08_{\pm 0.13}$	36.6	50.4	$0.15_{\pm 0.19}$
+ $\mathcal{L}_{CR}$	30.4	42.6	$0.48_{\pm 0.24}$	**37.2**	**51.1**	$\mathbf{0.65}_{\pm 0.22}$

Table 9: Comparison of document relevance and coherence of ModernBERT and STS Pre-trained ModernBERT. RBO measures the coherence.

from the BEIR benchmark. The results, presented in Table 10 and 11, demonstrate interesting patterns across different configurations.

On TREC benchmarks, our proposed approach achieves best NDCG on both versions of TREC test data, +0.68 and +0.21 compared to the strongest baseline, +1.37 and + 0.30 compared to the simple FT model.

Our $\mathcal{L}_{CR}$ configuration achieved the best overall performance with an average score of 44.94 across all BEIR datasets, showing consistent improvements over the public checkpoint (43.56) and standard fine-tuning (44.46). Notable improvements were observed on several key datasets as Scifact, HotPotQA, Arguana, and NFCorpus, demonstrating enhanced capability in handling complex, multi-hop questions as well as maintaining robust retrieval capabilities across specialized content.

Interestingly, while the Full configuration, showed strong performance on specific datasets like HotpotQA (55.22) and Quora (88.87), it didn't achieve the best overall average (42.96). This suggests that the combination of all components might lead to some interference effects in certain domains. We used a model trained on MS-MARCO. Thus, we only focus on the average result metric as individual benchmarks would require fine-tuning.

E Reformulation experiments

As discussed in this paper, query reformulation is out of the scope of this work as it introduces strong drawbacks, including cost and latency, to the retrieval pipeline. However, for completeness we ran two simple train-free reformulation approaches as additional baselines.

The first approach, here indicated as *Centroid*, was introduced by Kostric and Balog (2024). In short, the DR model first computes the embeddings of the original query $\mathbf{e}_q$ and all of its k reformulations $\mathbf{e}_{r_1} \dots \mathbf{e}_{r_k}$. Then, the embedding used to compute the similarity against indexed documents is defined as the average of all query embeddings $\frac{1}{k+1}(\mathbf{e}_q + \sum_i \mathbf{e}_{r_i})$. The motivation of the approach is that the centroid of these rewrites adds robustness to the DR model as the center of mass of multiple reformualtions will likely correspond better to the user's information need than a single rewrite. Note that, in the original work, a waighted average is used, where each reformulation has a score depending on the conversation history, not available on our task. As second baseline, we ran the DR model on all available reformulation and selected the documents with highest scores with respect to the reformulated query. By doing so, the model can select documents that receive low score with the original query but high score with a reformulation.

Configuration	NDCG@10	RBO@5
TREC-DL '19		
Public ckpt	64.35	$14.07_{\pm 0.12}$
FT	69.77	$14.57_{\pm 0.13}$
+ Gen. Qs	70.46	$15.59_{\pm 0.15}$
+ $\mathcal{L}_{QQ}$	69.39	$16.07_{\pm 0.14}$
+ $\mathcal{L}_{CR}$	71.14	$19.47_{\pm 0.14}$
TREC-DL '20		
Public ckpt	63.36	$15.56_{\pm 0.13}$
FT	65.52	$16.11_{\pm 0.14}$
+ Gen. Qs	65.49	$16.35_{\pm 0.15}$
+ $\mathcal{L}_{QQ}$	65.61	$16.30_{\pm 0.14}$
+ $\mathcal{L}_{CR}$	65.82	$19.09_{\pm 0.12}$

Table 10: Results on TREC DL benchmarks.

We tested these approaches on TREC-DL '19 and '20 benchmarks. Results are showed in Table 12.

Both reformulation approaches show poor performance. We conjecture these methods highlight their benefits on conversational settings rather

Configuration	Scifact	Arguana	Nfcorpus	Webis-touche2020	Fiqa	Dbpedia-entity	Climate-Fever	Quora	Scidocs	Trec-covid	HotPotQA	AVG
Public ckpt	59.98	51.01	32.11	16.55	47.42	35.31	25.35	87.69	17.39	58.77	47.57	43.56
FT	59.08	47.72	31.94	23.89	46.89	36.72	25.16	88.15	16.88	60.60	52.04	44.46
+ Gen. Qs	59.56	46.86	32.26	23.82	44.42	35.86	26.51	86.81	16.65	58.48	53.54	44.07
+ $\mathcal{L}_{QQ}$	59.56	46.74	31.93	24.52	46.38	36.71	25.39	88.91	16.88	55.23	54.50	44.25
+ $\mathcal{L}_{CR}$	60.46	48.50	32.77	23.54	46.16	36.62	26.04	88.61	17.24	60.26	54.19	**44.94**
Full	59.95	45.42	32.61	23.88	43.18	36.53	25.95	88.87	16.99	44.00	55.22	42.96

Table 11: Results on BEIR benchmark. The model, MPNet, is trained on MS-MARCO. Dataset as CQAdupstack, BioASQ, Signal1m, Trec-news, Robust04 are not included sice they were not available.

Configuration	P@1	NDCG@10
TREC-DL '19		
No reformulation	83.72	69.77
Centroid	76.74	67.16
Best	82.80	65.02
TREC-DL '20		
No reformulation	81.48	65.52
Centroid	78.22	61.68
Best	80.95	65.47

Table 12: Results of reformulation approaches on TREC DL benchmarks.

than single turn Q&A. Other authors (Wang et al., 2020b) applied more complex techniques, where a reformulator model is trained and rewarded to generate queries to achieve higher retrieval performance. However, results showed a very limited improvement on TREC-DL benchmarks, less than 0.1% NDCG@10.

F Retrieve and Generate

In order to further explore the contribution of our DR models on downstream applications, we simulated a RAG pipeline.

We used an LLM, Mistral-7B-Instruct-v0.2, to generate an answer while using the top-5 documents retrieved by various DR models. Specifically, we used the MPNet fine-tuned on MS-MARCO with MNRL and our CR losses. The LLM, evaluated on KILT benchmarks (Petroni et al., 2020), showed an average improvement in accuracy by +0.4%, from 60.0 (MNRL) to 60.4 (our CR loss).

This quick experiment is not meant to be exhaustive as the focus of this work is improving quality and coherence of DR models. The aim is to provide an intuition of downstream effects in terms of accuracy. These results need to be explored further.