A Systematic Reproducibility Study of DiSCo for Conversational Search

Stan Fris
s.c.j.fris@uva.nl
University of Amsterdam
Netherlands

Jan Hutter
jan.hutter@student.uva.nl
University of Amsterdam
Netherlands

Jan Henrik Bertrand
jan.henrik.bertrand@student.uva.nl
University of Amsterdam
Netherlands

Abstract

This reproduction study reproduces DiSCo [15] – a distillation method for first-stage retrieval based on SPLADE in an attempt to alleviate the cost and latency that comes with query rewriting through LLMs in conversational search. Reproducing the results, we find that DiSCo outperforms other methods in in-domain settings, but we observed poor MRR and nDCG@3 in out-of-domain settings, while recall remains high. DiSCo replaces an MSE on the embedding space combined with a contrastive loss with a single KL divergence on the similarity scores. However, they do not provide theoretical or empirical evidence that is replaces a contrastive incentive. Thus, we add a contrastive loss to the objective in an attempt to improve generalization, but only observe significantly positive results in MRR and nDCG@3 for in-domain data. To improve the representativeness of the samples used for distillation, we also perform ablation studies on the number of samples used for the KL divergence objective and its temperature, but do not obtain significantly positive results. Finally, conduct a study on regularization and find that even stronger regularization than used in SPLADE improves the sparsity of the representations by a large amount without affecting recall. We also find that more regularization helps keep the representations remarkably sparse for longer conversations. Our code is available at https://github.com/Janhutter/disco-conv-splade

ACM Reference Format:
Stan Fris, Jan Hutter, and Jan Henrik Bertrand. 2025. A Systematic Reproducibility Study of DiSCo for Conversational Search. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

CS Conversational Search

1 Introduction

First-stage retrieval is an important part of retrieval and search systems. It makes a broad selection of possibly relevant documents for a given user query [5, 18] before a second-stage ranker yields the final ranking order. This selection, made during the first-stage ranking, is typically drawn from a large corpus. Therefore, there are two conventionally important metrics in this task: recall [26], and computation time [14]. Achieving high recall is vital to ensure that the model effectively filters out irrelevant documents without discarding relevant ones. Maintaining low computational cost is equally important, since first-stage rankers operate over an extensive set of candidate documents. Still, in most use cases, such as conversational search or search engines, they are expected to run in fractions of a second to allow for a decent user experience.

A promising approach that balances these two objectives is the use of sparse retrievers [4, 8, 14]. Unlike dense retrieval models, which require costly vector operations, sparse retrievers leverage inverted index structures that enable efficient large-scale search. Additionally, the multiplication of sparse vectors yields efficiency gains over a dense vector system [3]. Recent advances such as SPLADE [8] and SparTerm [4] have demonstrated that sparse retrieval models can reduce computational overhead, achieving competitive or superior recall. These properties make them highly suitable for first-stage retrieval in modern information retrieval pipelines.

A promising application of sparse retrieval models is Conversational-Search (CS) [20], which centers on modeling the evolution of users' information goals over the course of the interaction. Specifically, it requires the consideration of previous turns in the conversation to contextualize the latest query correctly. A challenge of this paradigm is to remain accurate when more turns are added to the conversation, and to separate relevant from irrelevant information when contextualizing. An approach that has been shown to be effective is the use of LLMs for conversation rewriting [9, 15, 17]. However, LLM inference is a computationally expensive operation, increasing latency and cost. Therefore, recent research has investigated the use of a student-teacher setup for learning to represent rewritten conversations [9, 17]. These approaches make use of a Mean-squared Error (MSE) loss to make the student query representation approach the LLM-rewritten teacher query representation. Research by Lupart et al. [15] shows that this approach can be further improved by using a Kullback-Leibler Divergence (KLD) loss over the computed similarity scores. This raises the question of to what extent the state-of-the-art results of DiSCo are reproducible (**RQ1**). The use of KL-divergence allows the model to learn from multiple samples simultaneously and relaxes the constraint of optimizing toward a single representational vector. While showing effective results, there are several factors that remain unexplored, in particular, it is unclear how effectively KLD can replace or be combined with a contrastive loss, and how training samples can be utilized more efficiently (**RQ2**). Furthermore, Lupart et al. [15] observed a degradation in performance when longer input sentences are used. We therefore investigate whether regularization strategies can mitigate this effect (**RQ3**).

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Theoretical Background

This section starts by locating the method proposed in this work within the broader information retrieval context and ends with an explanation of DiSCo [15], which this work builds upon. Thereby, it gradually introduces a series of concepts relevant to the method at hand.

2.1 Two-Stage Retrieval

Large-scale information retrieval systems are usually organized in two stages described in the following.

First Stage. A broad selection of documents, usually called a candidate list, is made to filter the corpus for documents that are potentially relevant given the query. The ranking order in this stage is not particularly relevant, as it will be changed in the second stage. Much more important is that all potentially relevant documents from the (usually very large) corpus of all documents are retrieved, i.e., a low number of false negatives. This requirement, combined with the ranking order being not particularly relevant, yet yields recall as a highly suitable and common evaluation metric for first-stage retrieval. Given the usual size of document corpora, sparse retrieval is a common choice for this application in order to comply with the requirements regarding computational efficiency. A more detailed description of learned sparse retrieval will follow in Section Section 2.2.

Second Stage. Given the candidate list, the second stage re-ranks the candidates and yields the final ranking. Consequently, the objective of the second stage is to accurately assess the precise relevance of each selected document in relation to the other documents that have been selected by the first stage. This difference in objective compared to the first stage calls for a very different setup compared to the first stage. For this second stage, there are various methods, including cross-encoders like monoBERT [22], which take in both the query and a candidate to predict a relevance score, referred to as pointwise ranking. Another approach is pairwise re-ranking, which, given the query and two candidates, predicts which candidate is more relevant and then updates the ranking based on that. An example of this approach is duoBERT [22]. In this work, we focus on sparse first stage retrieval.

2.2 Learned Sparse Retrieval

Learned Sparse Retrieval (LSR) is an approach that has proven effective in first-stage retrieval [4, 8, 15]. The goal of LSR is to learn sparse document- and query-representations that, when multiplied (i.e., when the dot-product is taken), yield a measure of similarity. Sparse means that only a small number of vector elements are non-zero. In practice, the embedding space is usually the vocabulary of a language model, where the elements represent the importance of the corresponding token in the given document or query.

A widely used approach to sparse representation learning was introduced by Bai et al. [4], where the sparse embedding space is defined using the BERT WordPiece vocabulary ($|V| = 30522$) vector of the BERT model. For the tokenized sequence $\mathbf{t} = (t_1, \ldots, t_n, \ldots, t_N$ we obtain BERT embeddings $\mathbf{h} = \{h_1, \ldots, h_i, \ldots, h_N\}$, we can compute the importance w_{ij} of token j *from the vocabulary* for index i *in the input sequence* as follows (definition from Formal et al. [8] and Bai et al. [4]):

$$w_{ij} = \text{transform}(h_i)^T E_j + b_j \quad j \in \{1, \ldots, |V|\}\,, \tag{1}$$

with E_j being the BERT input embedding for token j, b_j a token-level bias and transform(.) a linear layer with GELU activations and LayerNorm. Following this operation, the weights are combined into a single weight per index of the vocabulary vector, for example, using the definition given in Formal et al. [8]:

$$w_j = g_j \times \sum_{i \in t} \text{ReLU}(w_{ij})\,. \tag{2}$$

Here, g_j is a binary mask, which can be learned or a BOW mask to ensure occurrences of the term in the source text, and the ReLU ensures positivity of all signals. This sparse vector approach combines the fast inference of count-based models such as BM25 with the representational depth of high-dimensional vectors.

In the context of first-stage retrieval, we can define the task more formally now: given a query q and document d, determine the probability that this document is relevant given these representations, $p(R_{q,d}|q, d)$. In sparse retrieval, this is often estimated by generating embeddings $E(q), E(d)$ for the query and document, and computing the dot product to reach a similarity score, which is directly mapped to the relevance score:

$$p(R|q, d) = sim(E(q), E(d))\,, \tag{3}$$

where *sim* is any similarity measure such as the dot product [4, 8, 15]. Sorting the documents by similarity scores can be used to yield a ranking of documents in case no second stage is used.

2.3 Conversational Search

CS operates in a multi-turn setting, where at each turn, the system is given the dialogue history up to that point. This includes all previous user queries, system responses, and the current user query. Using this context, the system must retrieve the documents most relevant to the user's information need. In CS, conversational history can grow considerably large, with long turn dependencies. This differs from ad-hoc search, where queries typically consist of a limited number of words. Therefore, the main challenge in CS is contextualizing the latest user query with the conversational history to filter for information needed to address the current question [20] without omitting crucial context. Current research tries to address this challenge utilizing two approaches: (1) query reformulation [9, 15, 25, 29] and (2) representation learning [20, 23]. This research will mainly focus on representation learning.

2.4 Contrastive Loss and Hard Negative Sampling

To transform the retrieval objective in Eq. (3) into one that explicitly discriminates between relevant and irrelevant documents, we maximize similarity for target documents while minimizing it for negative ones. In practice, many approaches work with a contrastive loss, which generally does the following [4, 8]:

$$\mathcal{L}_{\text{rank}} = -\log \frac{e^{sim(q_i, d_i^+)}}{e^{sim(q_i, d_i^+)} + e^{sim(q_i, d_i^-)}}\,, \tag{4}$$

where d_i^+ and d_i^- denote embeddings of positive and negative documents, respectively, given a query q_i. This optimization is difficult, as our embeddings are forced to be more similar to positive document embeddings without a direct reference [15].

2.5 Conversational Query Rewriting

Query rewriting has been shown to be an effective method for improving representations [25, 29]. This task, which is referred to as Conversational Query Rewriting (CQR), can be carried out by human rewriters, as well as, to some extent, Large Language Model (LLM) [9, 15, 17]. However, query rewriting is computationally expensive and time-consuming, which limits its applicability to large-scale CQR systems. Research by Hai et al. [9], Mao et al. [17], Yu et al. [29] has shown that knowledge distillation can be applied to handle rewriting and retrieval tasks in a single step using only one model. We can distill from embeddings of human-rewritten queries to teach a student model to learn query representations that take relevant parts of the context into account. A common method to learn these representations is the use of the MSE loss between the student embedding vector and the teacher embedding vector [9, 17, 25, 29]. By using the MSE loss, the student embedding vector, in theory, converges to the teacher embedding vector.

2.6 Optimization Using the Kullback-Leibler Divergence

Rather than distilling on the representations themselves, Lupart et al. propose using the Kullback-Leibler divergence [12] *between the similarity scores* generated by the student and teacher models. This can be formalized as follows: given an input conversation q_{conv}, rewrite this query using a human or LLM, resulting in q_{rw}. We can then use two pre-trained encoder models E_d and E_q to compute embeddings, as well as compute an embedding for our student model $\tilde{E}_d$. We can then compute similarity scores following Eq. (3), and compute the KL-Divergence Loss on the similarity scores as follows:

$$Loss = KLD\left(sim(\tilde{E}_q(q_{conv}), E_d(d)), sim(E_q(q_{conv}), E_d(d))\right) . \quad (5)$$

This contrasts with methods preceding DiSCo, like SPLADE, which uses a combination of an MSE loss and a contrastive loss as the objective. The DiSCo objective (cf. Section 2.6) relaxes the constraints compared to the traditional combined objective as it only compares the resulting similarity scores, thereby allowing for any query representation that yields the desired similarity scores. Formally, the traditional distillation of the representations can be noted as

$$\tilde{E}_q(q_{conv}) \rightarrow E_q(q_{rw}) . \quad (6)$$

Instead, the relaxed distillation introduced by DiSCo only constrains the representations to the hyperplane that yields the required similarity scores:

$$\tilde{E}_q(q_{conv}) \rightarrow \{X \in \mathbb{R}^h \mid X^\top E_d(d) = s_{q_{rw}}\} \quad (7)$$

To stabilize training, Lupart et al. uses a cross-entropy loss weighted at 1% in combination with the KL-Divergence (weighted at 99%).

3 Method

This section introduces the objectives that will be investigated. We describe the setting of DiSCo, as well as the datasets used. Furthermore, we describe the extensions we will perform to investigate limitations found in the methods of Lupart et al. [15].

3.1 Setting

Similar to other approaches in CS [9, 15, 17, 20], DiSCo relies on two pre-trained encoder models for query and document representation. Lupart et al. [15] assume that the document encoder is sufficiently effective at modeling documents and therefore requires no additional fine-tuning. Only the query encoder is fine-tuned to improve performance on longer conversations.

3.2 Objective

When Lupart et al. propose their relaxed objective based almost only on the KL-Divergence of the similarity scores for one positive and 16 negative samples, they argue that this implies both a distillation and contrastive incentive. While using this setup might intuitively resemble a contrastive incentive to a certain extent, the objective behind the KL-Divergence is the alignment of distributions rather than contrasting positive and negative examples. We see two major issues with the DiSCo approach: (1) to the best of our knowledge, there are no theoretical guarantees for this setup and Lupart et al. also do not perform any study on the ablation of a contrastive loss. Further, (2) their approach does not allow for a fine-grained control of the contrastive objective in combination with the distillation objective. Several related work confirms our concerns, reporting improvements when adding a weighted contrastive component to the objective [27, 28].

Both the positive results obtained with the relaxed score-based distillation and our line of reasoning on the addition of a contrastive loss motivate our objective for this work: a weighted combination of a KL-Divergence based distillation loss and the InfoNCE loss as a controllable contrastive component. Formally:

$$\begin{aligned} Loss = \lambda \cdot KLD\left(sim(\tilde{E}_q(q_{conv}), E_d(d)), sim(E_q(q_{conv}), E_d(d))\right) \\ + (1-\lambda) \cdot InfoNCE_{sim} \end{aligned}$$

where $\lambda \in [0, 1]$ acts as a weighting parameter to balance the contrastive and the distillation incentive.
We hypothesize that this objective will yield better generalization resulting in positive effects on performance.

3.3 Datasets and Metrics

Table 1: Dataset statistics

Dataset	Split	# Conv	# Turns	Collection
TopiOCQA	Train	3,509	45,450	25M
	Test	205	2,514	
Trec CAsT 20	Test	25	216	38M
Trec iKAT 23	Test	25	176	77M
Trec iKAT 24	Test	17	218	77M

To train and evaluate the utilized method and baselines, we make use of various conversational passage datasets, as shown in Table 1, namely:

- TopiOCQA [1] is a large conversational dataset derived from Natural Questions (NQ) [13] that emphasizes topic switching within a Wikipedia-based corpus. In this dataset, each new Wikipedia hyperlink introduced at a given turn is treated as a topic switch.
- TREC CAsT 2020 [6], iKAT 2023, and iKAT 2024 [2] are smaller conversational datasets utilized for testing out-of-distribution performance.

This constitutes a slight deviation from Lupart et al. [15]. Due to formatting issues in the QReCC dataset, we excluded it from the present study. As a result, out-of-distribution evaluations are conducted using models trained on TopiOCQA rather than QReCC. Additionally, we replaced CAsT 2022 with iKAT 2024 due to problems encountered during the collection of the former.

3.4 Ratio of Negatives and Positives in Sampling

In addition to improving the objective, using a larger amount of positive samples could allow for a more representative distribution and thus a more informed loss. This could enable a faster training convergence through enhanced training signal. Therefore, there remains a research gap in investigating to what extent alternate sampling strategies can be used with the KL-divergence. Specifically, we can show that utilizing more positive samples from the teacher inside the KL divergence loss will lead to a better approximation of the teacher similarity scores distribution. Starting from the teacher similarity scores $T_\tau(i)$ and student similarity scores $S_\tau(i)$

$$T_\tau(i) = \frac{\exp(s_i^{(T)}/\tau)}{\sum_{j=1}^{n}\exp(s_j^{(T)}/\tau)}, \qquad S_\tau(i) = \frac{\exp(s_i^{(S)}/\tau)}{\sum_{j=1}^{n}\exp(s_j^{(S)}/\tau)}.$$

The KL divergence from teacher to student, written as an expectation, is:

$$D_{\mathrm{KL}}(T_\tau \,\|\, S_\tau) = \mathbb{E}_{i\sim T_\tau}\left[\log\frac{T_\tau(i)}{S_\tau(i)}\right] = \sum_{i=1}^{n} T_\tau(i)\,\log\frac{T_\tau(i)}{S_\tau(i)}.$$

Here we can see that from the law of large numbers, increasing the number of samples drawn from the teacher model yields a more accurate approximation of the true expectation in the KL divergence.

Furthermore, adding more samples from high-probability areas will lead to a more effective representation of the teacher distribution, as these samples are underrepresented in the training objective. This is similar to the intuition of importance sampling [24], where the sampling distribution is adjusted to place greater weight on regions that contribute most to the target expectation, thereby reducing variance and improving the accuracy of the estimator.

However, increasing the number of documents per query introduces a trade-off. While incorporating more documents into the loss function raises the computational cost of each training step, due to the increased information we could increase the learning rate, reducing the amount of training steps required.

4 Experiments & Results

4.1 Baselines

Here, we compare DiSCo to a wide range of conversational search methods, using the results reported by Lupart et al. [15].

Query Rewriting methods. SPLADE-[Human/T5/Mistral]QR does retrieval using the pretrained SPLADE weights, and where the input conversation is rewritten into a shorter, single-sentence question by their respective methods[1]. These QR methods are compared to original SPLADE performance, without any rewriting. IterCQR [10] trains a conversational query reformulation method that is called once at test-time. CHIQ-Fusion [19] fuses ranking lists of different QR methods, and CHIQ FT utilizes rewrites of a QR fine-tuned T5 model. LLM4CS [16] utilizes the aggregation of different prompting techniques. The results of LLM4CS are reproduced by Lupart et al. [15].

Supervised fine-tuned methods. convSPLADE [17] and convANCE [17] are two methods fine-tuned on the whole conversation with an InfoNCE loss.

Distillation-based methods. LeCoRe [17], QRACDR [21], and ConvDR [29] learn to represent the human-rewritten query representation through distillation. Similarly to DiSCo, these models do not rely on LLM calls during inference. LeCoRe utilizes sparse representations from SPLADE, while QRACDR and ConvDR utilize dense encoders.

4.2 Implementation Details

Similarly to [15], we utilize the SPLADE++ [7] for all DiSCo and SPLADE methods. Unlike the DiSCo authors, we found in preliminary experiments that extending training from 5 to 7 epochs led to improved convergence. We utilized the same hyperparameters as the original authors; specifically, we set λ_q and λ_d to 0, as these were the values found to be used in their code.

4.3 RQ1: Reproduction

To verify the improvements of DiSCo over earlier methods in CS, we reproduce the different teachers used for DiSCo and the QR variants of SPLADE. The results of the in-domain performance are shown in Table 2, and the results for out-of-domain performance are shown in Table 3.

We observe the following: (1) DiSCo outperforms all other methods across all metrics on in-domain experiments, including approaches that apply rewriting at inference time, highlighting the effectiveness of the proposed method. (2) Utilizing DiSCo with multiple teachers, based on T5 and Mistral rewrites, does not improve performance compared to only utilizing the Mistral teacher. Only a marginal improvement from using multiple teachers is observed on the iKAT 2024 dataset, whereas performance decreases across all other datasets. (3) All reproduced methods on the out-of-domain datasets show a general decrease in MRR and nDCG@3 compared to the baselines found in the original research. However, the recall does seem slightly higher compared to the results published by the original authors. A possible explanation for this is the alignment

[1]Similar to Lupart et al. [15], we utilize Mistral mistralai/Mistral-7B-Instruct-v0.2 and T5 castorini/t5-base-canard for distillation and QR.

Table 2: In-Domain performance on TopiOCQA. DiSCo multi-teach for TopiOCQA is the combination of T_1 and T_2. Hyperscripts ⋆ are paired t-tests with $p < 0.05$ comparing multi-teacher with single-teacher DiSCo. RW denotes the LLM/human rewriting method used as input to the model. FC refers to models that do not use any rewriting at inference and just take the full context as input. # denotes the number of LLM calls used at inference.

	Method	RW	#	TopiOCQA			
				R@100	R@10	MRR	nDCG@3
Query Rewriting	SPLADE no rewrite	FC	0	0.510	0.258	0.151	0.134
	SPLADE T5QR (T_0)	T5	1	0.675	0.506	0.324	0.316
	SPLADE MistralQR (T_2)	Mistral 2	1	0.762	0.591	0.372	0.359
	IterCQR	GPT-3.5	1	0.620	0.426	0.263	0.251
	LLM4CS	GPT-3.5	5	-	0.433	0.277	0.267
	CHIQ FT	T5	1	-	0.510	0.300	0.289
	CHIQ-Fusion	Llama 2	6	-	0.616	0.380	0.370
SFT	convANCE	FC	0	0.710	0.430	0.229	0.205
	convSPLADE	FC	0	0.720	0.521	0.295	0.307
Distillation	ConvDR	FC	0	0.611	0.435	0.272	0.264
	QRACDR	FC	0	0.758	0.571	0.377	0.365
	LeCoRe	FC	0	0.735	0.543	0.320	0.314
	DiSCo T5	FC	0	0.857	0.608	0.355	0.337
	DiSCo Mistral	FC	0	0.879⋆	0.676⋆	0.409⋆	0.396⋆
	DiSCo multi-teach	FC	0	0.869	0.646	0.387	0.369

of the training set, as we train the out-of-domain models on TopiOCQA, while the original authors possibly trained on QRECC or a combination.

4.4 RQ2: Objective

In this section, we investigate components of the DiSCO loss function. To verify and extend the method, we propose 2 main changes: adding a contrastive loss component and changing the number of samples in the KL divergence loss.

4.4.1 Combining DiSCo with Contrastive Loss. Our analysis in Section 3.2 indicated that there is insufficient empirical and theoretical evidence for the use of DiSCo without a contrastive component, where other related methods do use this [27, 28] and report positive effects from doing so.

In Table 4 we show the results of training the DiSCo model with varying amounts of contrastive loss. The decimal numbers mentioned with "InfoNCE" are the weights, where the sum of the DiSCo KL divergence weight and the InfoNCE weight is always one. Hence, "DiSCo + 0.10 InfoNCE" refers to 10% InfoNCE and 90% KL divergence. The range of objectives has been evaluated on both in-domain (TopiOCQA) and out-of domain (CAsT20) data. When considering the in-domain data, we observe that using an InfoNCE component of 10% or 20% leads to significantly improved performance on ranking metrics compared to DiSCo. For recall-based metrics, we see that Recall@100 does decrease slightly, although we do not find that these decreases are significant under a 95% confidence level when tested using a paired t-test with Bonferroni correction. Recall@10 results are similar across models, with a slight improvement for a 5% InfoNCE component.

These results confirm the theoretical motivation that the addition of a ranking loss improves the ranking capacity of the model and, therefore, performance in terms of MRR and nDCG@3. Additionally, we also observe that this can lead to higher Recall@k scores for lower k values, possibly due to the improved ranking of individual items, leading to a better ranking, which is less important for high-k values.

When evaluating on out-of-domain data, we observe that adding the InfoNCE loss to the DiSCo objective makes the model generally perform worse. Specifically, Recall10 and Recall100 performance scores are significantly worse when using a 10% or 20% InfoNCE component. Other performance decreases were not found to be statistically significant.

Overall, we conclude that adding an InfoNCE loss can improve the ranking performance of the DiSCO model without significantly decreasing recall performance. This implies that in settings where ranking performance is relevant, adding the InfoNCE can effectively improve the DiSCo model. The same applies for Recall@k with low k values. In cases where only the recall is important, the InfoNCE is not necessary. Thus we partially confirm the DiSCo authors intuition that using a KL divergence on the similarity scores only is sufficient following our empirical evidence.

4.4.2 Multi-Sampling in the KL divergence. As proven in Section 3.4, the number of samples drawn from the teacher models should yield a more accurate approximation of the true expectation in the KL divergence. To evaluate performance, we visualize the sample distributions on the test set to compare them to the teacher model, as shown in Fig. 3. We investigated the use of different amounts of negative samples for the KL divergence. Here, we observe that the teacher and student models are generally not aligned across different datasets. Although the models might be able to effectively learn representations leading to similar or better performance on ranking metrics, we don't find that representations exactly match those from the teacher model. However, the shape of the frequency distribution varies with the number of samples; in particular, when only a single negative sample is considered, differences between individual curves are minimal.

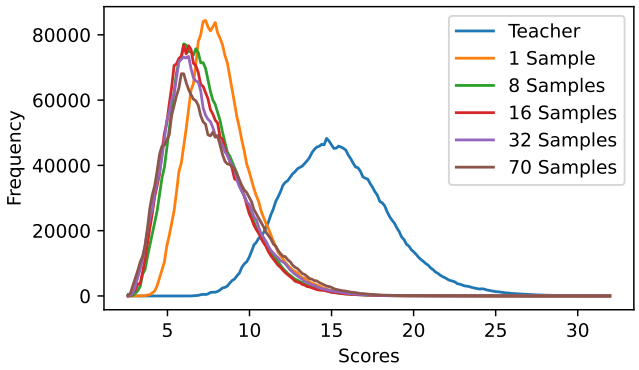

Figure 1: Distributions over the predicted similarity score for documents in the TopiOCQA test set, with teacher distribution (SPLADE MistralQR) for reference.

In addition to evaluating the KL divergence curves, we also consider increasing the amount of negative samples for the KL

Table 3: Zero-shot performance on out-of-domain performance on CAsT 2020, iKAT 2023, and iKAT 2024. The DiSCo versions are trained on TopiOCQA, while the other baselines have been trained on QReCC and are taken from the original research [15]. DiSCo multi-teach for TopiOCQA is the combination of T_1 and T_2. RW denotes the LLM/human rewriting method used as input to the model. FC refers to models that do not use any rewriting at inference and just take the full context as input. # denotes the number of LLM calls used at inference. No significant results have been found.

	Method	RW	#	CAsT 2020			iKAT 2023			iKAT 2024		
				R@100	MRR	nDCG@3	R@100	MRR	nDCG@3	R@100	MRR	nDCG@3
Query Rewriting	SPLADE HumanQR	Human	0	0.683	0.132	0.096	0.210	0.063	0.031	0.442	0.154	0.082
	SPLADE T5QR	T5	1	0.476	0.083	0.054	0.061	0.015	0.007	0.086	0.021	0.013
	SPLADE MistralQR	Mistral 2	1	0.663	0.124	0.104	0.120	0.026	0.015	0.237	0.052	0.021
	LLM4CS	GPT 3.5	5	0.504	0.618	0.444	0.133	0.154	0.099	–	–	–
	CHIQ FT	T5	1	–	0.463	0.316	–	–	–	–	–	–
	CHIQ Fusion	Llama 2	6	–	0.540	0.380	–	–	–	–	–	–
	DiSCo Fusion	Mistral 2	1	0.630	0.122	0.083	0.120	0.026	0.015	0.393	0.115	0.063
SFT	convSPLADE	FC	0	0.446	0.338	0.234	0.101	0.144	0.085	–	–	–
Distill	QRACDR	FC	0	0.324	0.442	0.303	–	–	–	–	–	–
	LeCoRe	FC	0	0.467	–	0.290	–	–	–	–	–	–
	DiSCo Mistral	FC	0	0.591	0.109	0.087	0.161	0.038	0.021	0.251	0.078	0.051
	DiSCo multi-teach	FC	0	0.548	0.096	0.068	0.117	0.041	0.025	0.255	0.078	0.052

Table 4: Results after adding a weighted contrastive loss (InfoNCE) to the KL divergence The weight of DiSCo and InfoNCE sum to 1. ⋆ is for significantly better and † for significantly worse results under a Bonferroni-corrected paired t-test with a 95% confidence level.

Loss	MRR	R@10	R@100	nDCG@3
TopiOCQA (In-Domain)				
DiSCo	0.409	0.676	**0.879**	0.396
DiSCo + 0.05 InfoNCE	0.414	**0.687**	0.875	0.400
DiSCo + 0.10 InfoNCE	0.421*	0.673	0.877	0.408*
DiSCo + 0.20 InfoNCE	**0.424***	0.676	0.874	**0.412***
CaST20 (Out-of-Domain)				
DiSCo	**0.109**	**0.250**	**0.591**	**0.087**
DiSCo + 0.05 InfoNCE	0.104	0.226	0.582	0.084
DiSCo + 0.10 InfoNCE	0.100	0.201†	0.562†	0.077
DiSCo + 0.20 InfoNCE	0.102	0.197†	0.548†	0.074

divergence to improve performance. The results of these experiments, as well as the KL divergence values, are shown in Table 5. Contrary to our hypothesized improvements in performance for increased sample amounts, we observe that performance does not increase with a larger sample size in practice. A sample amount of 16 negatives (as was used in the original DiSCo paper), leads to the best in-domain results, with performance being significantly better than other models in most settings. Looking at the KL divergence, we observe this does become lower when using more samples, indicating that distributions do move closer when using more samples, confirming our intuition from Section 3.4. Looking at out of domain performance, we observe that there is a clear increase in performance when using more samples, with 70 negative samples leading to the best scores across all metrics. However, none of these increases in performance were found to be significant compared to the amount of 16 negative samples used in the original DiSCo setup.

A possible explanation for our observations is that the positive sample is under-represented when using many samples in the in-domain setting. In the KL divergence loss, the positive sample is included in the negative sample distribution and given the score of the highest sample. By including more samples, the influence of the positive sample on the distribution and therefore the loss decreases. This means that gradient updates will be less focused on the positive sample. This can lead to less fine-grained learning for this sample and ultimately decreased performance. As it is likely that positive samples are distributed differently in the CAsT20 dataset, this effect is not observed here.

4.4.3 Effect of Temperature on Model Performance. To examine how temperature affects the performance of different loss objectives, we conduct an ablation study. Adjusting the temperature is expected to help when working with noisy labels, in our case, similarity scores Kim et al. [11]. We observed no significant performance change when varying the temperature parameter in the softmax over similarity scores. The results are reported in Appendix A.

4.5 RQ3: Sparsity and Inference Speed

In first-stage retrieval, inference speed remains an important factor and a primary reason to use sparse retrieval methods. We investigate the inference speed of DiSCo, evaluating whether adding more regularization can improve sparsity and therefore inference speed.

In Table 7, we show the results of adding increasing amounts of regularization. We have included the DiSCo λ-values of 0, as well as the SPLADE λ-values $\lambda_d = 0.0005$ and $\lambda_q = 0.001$. The exact lambda values are described in **??**. Here, we observe that with increasing regularization, there is little change in performance

Table 5: Results with different amounts of negatives in the KL divergence. There is always one positive. ⋆ is for significantly better and † for significantly worse results under a Bonferroni-corrected paired t-test with a 95% confidence level.

Sample Amount	MRR	R@10	R@100	nDCG@3	D_{KL}
TopiOCQA (In-Domain)					
1	0.348†	0.613†	0.859†	0.326†	3.501
8	0.394†	0.657†	0.874	0.381†	3.861
16	0.409	**0.676**	**0.879**	**0.396**	3.698
32	**0.410**	0.662†	0.874	0.395	3.592
70	0.395†	0.655†	0.860†	0.381†	3.221
CAsT20 (Out-of-Domain)					
1	0.094	0.207	0.500†	0.067	4.679
8	0.098	0.202†	0.548	0.077	6.784
16	0.109	0.250	0.591	**0.087**	6.605
32	0.110	0.255	0.562	0.079	6.563
70	**0.111**	**0.264**	**0.620**	**0.087**	6.668

while substantially improving the FLOPS. For the "High" setting, only Recall10 shows a statistically significant degradation, while all other metrics remain within the significance bounds. For the higher λ setting, all values are within the same threshold, while reaching a FLOPS score that is two times lower. When looking at out-of-domain data performance, we make the same observation for higher lambdas. From this, we can derive that a substantial speed-up can be achieved in DiSCo without sacrificing model performance significantly.

Table 6: Mapping of the mentioned regularization settings to the actually used λ-values.

Reg. Setting	λ_d	λ_q
DiSCo	0	0
Splade Setting	5×10^{-4}	1×10^{-3}
High	1×10^{-3}	5×10^{-3}
Higher	1×10^{-2}	5×10^{-2}
Highest	5×10^{-2}	1×10^{-1}

Lupart et al. [15] show that the amount of activated embedding dimensions increases as the amount of conversation turns increases. We investigate how this increase appears in our findings, and compare it with regularized versions. In Fig. 2, we show sparsity across conversations with different amounts of turns. When we compare the original DiSCo setting with versions where increased regularization was applied, we see a significant improvement in sparsity of representations. Furthermore, we find that while DiSCo becomes proportionally much sparser with increased conversation lengths compared to the higher λ settings.

Furthermore, we also investigate performance across different numbers of turns, as shown in Fig. 3. Here, we see that for all models, performance generally decreases as the number of conversation turns increases. This can be explained by the fact that increased-length conversations are generally more complex and require the model to encode more information. Looking at the performance of regularized methods compared to DiSCo we observe that performance for "High" and "Higher" regularization is very similar to DiSCo performance. This indicates that regularization can be applied effectively to both long and shorter conversations, as performance changes compared to a non-regularized version are small.

Table 7: Results after adding regularization to. DiSCo uses 0 regularization. A table mapping the setting names to the actual λ-values used for regularization can be found in Table 6. ⋆ is for significantly better and † for significantly worse results under a Bonferroni-corrected paired t-test with a 95% confidence level.

Loss / Setting	MRR	R@10	R@100	nDCG@3	FLOPS
TopiOCQA (In-Domain)					
DiSCo (0)	0.409	**0.676**	0.879	0.396	3.790
SPLADE	0.405	0.659†	0.877	0.391	3.140
High	**0.412**	0.667	0.872	**0.398**	1.840
Higher	**0.412**	0.662	0.859	0.394	1.370
Highest	0.385†	0.619†	0.825†	0.371†	**0.470**
CaST20 (Out-of-Domain)					
Disco (0)	**0.109**	**0.250**	0.591	0.087	2.145
SPLADE	0.103	0.226	0.591	0.084	1.780
High	0.106	0.216	0.567	0.085	1.500
Higher	0.105	0.207†	0.572	**0.093**	1.110
Highest	0.076†	0.168†	0.486†	0.057†	**0.390**

Overall, we observe that regularization can be an effective method for the improvement of the inference speed of DiSCo. We observe that regularization leads to especially large improvements for longer conversations without suffering from decreased performance over DiSCo.

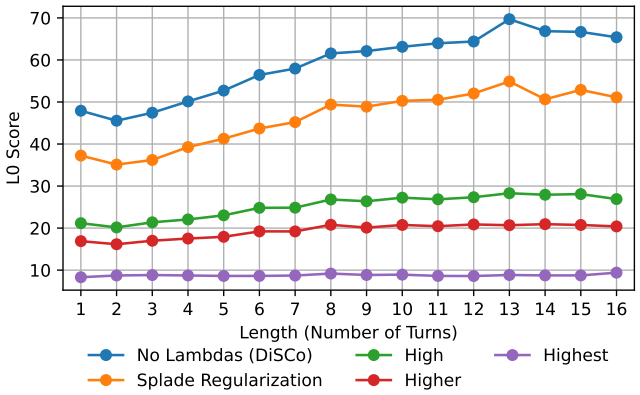

Figure 2: The L0 Score as a measure of sparsity for different conversation lengths (measured by the number of turns) across a range of regularization settings. Lower is better.

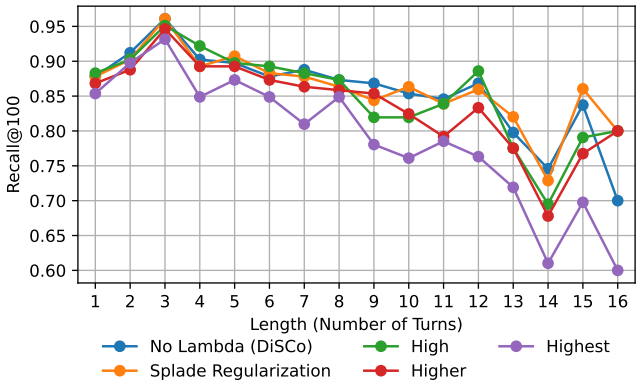

Figure 3: Recall@100 for different conversation lengths (measured by the number of turns) across a range of regularization settings. Higher is better.

5 Discussion & Future Work

This section discusses the outcomes of our reproduction study, with a focus on discrepancies between the results originally reported by Lupart et al. [15] and those obtained in our experiments. We analyse potential sources of variation, including dataset preprocessing and formatting choices.

5.1 Reproduction

We observed that the reproduced out-of-domain ranking performance was lower than the performance reported in the original work, which may be attributable to differences in dataset formatting and preprocessing. Despite this discrepancy, the recall remained consistently high relative to both competing methods and the original reported results, indicating that the model retained a strong ability to identify relevant candidates even under distributional shift.

Secondly, due to indexing issues, we relied on the pre-built SPLADE indexes provided for the iKAT dataset [2].

5.2 Future Work on Extensions

Regarding the extensions considered in our study, there are several considerations and suggestions for future work to take into account. First, the contrastive loss is incorporated using a simple linear weighting scheme with fixed weights. While this is a common and often effective method, we have not explored adaptive weighting strategies or alternative contrastive objectives, which could lead to more robust gains across metrics. Similarly, our temperature ablation is limited in scope and does not consider interactions between temperature, sample size, and loss composition. Second, in the analysis of multi-sampling for the KL divergence, we only vary the number of negative samples while keeping the sampling strategy fixed. Different negative sampling methods or explicitly re-weighting the positive sample could alter the observed effects, especially in the in-domain setting where larger sample sizes appear to under-emphasize positives. Finally, inference efficiency is evaluated using FLOPS and sparsity metrics rather than end-to-end latency in a deployed retrieval system. While these proxies are informative and regularly used in the evaluation of First-Stage retrieval systems, performance can differ in real-world settings. Future work should therefore validate these findings in larger-scale and more diverse retrieval settings and with system-level efficiency measurements. We also leave it to future work to explore the use of combined regularization, where more regularization can be used for longer queries or queries with higher complexity.

6 Conclusion

This reproduction study reproduces DiSCo finding that it outperforms the other methods in in-domain settings, while showing poor MRR and nDCG@3 in out-of-domain settings. Recall however remains high. The biggest hurdle in reproducing the authors' results was the pre-processing of the data, which consumed about one-third of the project's resources. Specifically, we were not able to reproduce the performance improvements regarding a multi-teacher setup reported by the DiSCo authors. Beyond reproducing the DiSCo experiments, we conducted a series of ablation studies in order to improve DiSCo. In that, we add a contrastive loss to the objective but only observe significantly positive results in MRR and nDCG@3 for in-domain data. Moreover, we investigate the number of samples used for the KL divergence objective and its temperature. However, we do not obtain any significantly positive results while doing so. Finally, we conduct a study on regularization, finding that adding significant amounts of regularization compared to previous methods improves the sparsity of the representations by a large amount without negative effects on recall. When investigating sparsity and recall for different conversation lengths, we find that more regularization helps keep the representations remarkably sparse for longer conversations.

References

[1] [n. d.]. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching | Transactions of the Association for Computational Linguistics | MIT Press. https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00471/110550/TopiOCQA-Open-domain-Conversational-Question

[2] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffrey Dalton, and Leif Azzopardi. 2024. TREC iKAT 2023: A Test Collection for Evaluating Conversational and Interactive Knowledge Assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Washington DC USA, 819–829. doi:10.1145/3626772.3657860

[3] Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. 2021. Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection. doi:10.48550/arXiv.2109.10739 arXiv:2109.10739 [cs].

[4] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval. doi:10.48550/arXiv.2010.00768 arXiv:2010.00768 [cs].

[5] Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J. Shane Culpepper. 2017. Efficient Cost-Aware Cascade Ranking in Multi-Stage Retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Shinjuku Tokyo Japan, 445–454. doi:10.1145/3077136.3080819

[6] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The Conversational Assistance Track Overview. doi:10.48550/arXiv.2003.13624 arXiv:2003.13624 [cs].

[7] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid Spain, 2353–2359. doi:10.1145/3477495.3531857

[8] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New

York, NY, USA, 2288–2292. doi:10.1145/3404835.3463098

[9] Nam Le Hai, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarski, and Laure Soulier. 2024. CoSPLADE: Contextualizing SPLADE for Conversational Information Retrieval. doi:10.48550/arXiv.2301.04413 arXiv:2301.04413 [cs].

[10] Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. IterCQR: Iterative Conversational Query Reformulation with Retrieval Guidance. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 8121–8138. doi:10.18653/v1/2024.naacl-long.449

[11] Taehyeon Kim, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun. 2021. Comparing Kullback-Leibler Divergence and Mean Squared Error Loss in Knowledge Distillation. doi:10.48550/arXiv.2105.08919 arXiv:2105.08919 [cs].

[12] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (March 1951), 79–86. doi:10.1214/aoms/1177729694 Publisher: Institute of Mathematical Statistics.

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (Aug. 2019), 453–466. doi:10.1162/tacl_a_00276

[14] Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid Spain, 2220–2226. doi:10.1145/3477495.3531833

[15] Simon Lupart, Mohammad Aliannejadi, and Evangelos Kanoulas. 2025. DiSCo: LLM Knowledge Distillation for Efficient Sparse Retrieval in Conversational Search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Padua Italy, 9–19. doi:10.1145/3726302.3729966

[16] Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. doi:10.48550/arXiv.2303.06573 arXiv:2303.06573 [cs].

[17] Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. 2023. Learning Denoised and Interpretable Session Representation for Conversational Search. In *Proceedings of the ACM Web Conference 2023*. ACM, Austin TX USA, 3193–3202. doi:10.1145/3543507.3583265

[18] Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, Seattle Washington USA, 437–444. doi:10.1145/1148170.1148246

[19] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual History Enhancement for Improving Query Rewriting in Conversational Search. doi:10.48550/arXiv.2406.05013 arXiv:2406.05013 [cs].

[20] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2025. A Survey of Conversational Search. *ACM Transactions on Information Systems* 43, 6 (Nov. 2025), 1–50. doi:10.1145/3759453

[21] Fengran Mo, Chen Qu, Kelong Mao, Yihong Wu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. 2024. Aligning Query Representation with Rewritten Query and Relevance Judgments in Conversational Search. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 1700–1710. doi:10.1145/3627673.3679534

[22] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. doi:10.48550/ARXIV.1910.14424

[23] Leila Tavakoli, Johanne R. Trippas, Hamed Zamani, Falk Scholer, and Mark Sanderson. 2025. Online and Offline Evaluation in Search Clarification. *ACM Transactions on Information Systems* 43, 1 (Jan. 2025), 1–30. doi:10.1145/3681786

[24] Surya T. Tokdar and Robert E. Kass. 2010. Importance sampling: a review. *WIREs Computational Statistics* 2, 1 (2010), 54–60. doi:10.1002/wics.56 _eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.56.

[25] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational Query Rewriting for Retrieval with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 10000–10014. doi:10.18653/v1/2022.emnlp-main.679

[26] Yan Xiao, Yixing Fan, Ruqing Zhang, and Jiafeng Guo. 2023. Beyond Precision: A Study on Recall of Initial Retrieval with Neural Representations. In *Information Retrieval*, Yi Chang and Xiaofei Zhu (Eds.). Springer Nature Switzerland, Cham, 76–89. doi:10.1007/978-3-031-24755-2_7

[27] Yingrui Yang, Shanxiu He, Yifan Qiao, Wentai Xie, and Tao Yang. 2023. Balanced Knowledge Distillation with Contrastive Learning for Document Re-ranking. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '23)*. Association for Computing Machinery, New York, NY, USA, 247–255. doi:10.1145/3578337.3605120

[28] Yingrui Yang, Shanxiu He, and Tao Yang. 2024. On Adaptive Knowledge Distillation with Generalized KL-Divergence Loss for Ranking Model Refinement. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, Washington DC USA, 81–90. doi:10.1145/3664190.3672522

[29] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-Shot Conversational Dense Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event Canada, 829–838. doi:10.1145/3404835.3462856

A KL-Divergence Temperature Ablation Study

Table 8: Results on experiments with different temperatures for the KL-Divergence.

Run Name / Temperature	MRR	R@10	R@100	nDCG@3
TopiOCQA (In-Domain)				
Temperature 0.5	0.394	0.655	0.870	0.379
Temperature 1.0	0.409	**0.676**	**0.879**	0.396
Temperature 2.0	**0.411**	0.664	0.873	**0.399**
CaST20 (Out-of-Domain)				
Temperature 1.0	**0.109**	0.250	0.591	**0.087**
Temperature 0.5	0.105	**0.260**	**0.601**	0.072
Temperature 2.0	0.091	0.207	0.538	0.068