Learning Denoised and Interpretable Session Representation for Conversational Search

Kelong Mao
mkl@ruc.edu.cn
Renmin University of China
Beijing, China

Hongjin Qian
ian@ruc.edu.cn
Renmin University of China
Beijing, China

Fengran Mo
fengran.mo@umontreal.ca
Université de Montréal
Montreal, Quebec, Canada

Zhicheng Dou
dou@ruc.edu.cn
Renmin University of China
Beijing, China

Bang Liu*
bang.liu@umontreal.ca
RALI & Mila, Université de Montréal
Montreal, Quebec, Canada

Xiaohua Cheng
Zhao Cao
Huawei Poisson Lab
Beijing, China

ABSTRACT

Conversational search supports multi-turn user-system interactions to solve complex information needs. Compared with the traditional single-turn ad-hoc search, conversational search faces a more complex search intent understanding problem because a conversational search session is much longer and contains many noisy tokens. However, existing conversational dense retrieval solutions simply fine-tune the pre-trained ad-hoc query encoder on limited conversational search data, which are hard to achieve satisfactory performance in such a complex conversational search scenario. Meanwhile, the learned latent representation also lacks interpretability that people cannot perceive how the model understands the session. To tackle the above drawbacks, we propose a sparse **Le**xical-based **Co**nversational **RE**triever (LeCoRE), which extends the SPLADE model with two well-matched multi-level denoising methods uniformly based on knowledge distillation and external query rewrites to generate denoised and interpretable lexical session representation. Extensive experiments on four public conversational search datasets in both normal and zero-shot evaluation settings demonstrate the strong performance of LeCoRE towards more effective and interpretable conversational search.

CCS CONCEPTS

• **Information systems → Query representation**.

KEYWORDS

Conversational search; context denoising; lexical retrieval

ACM Reference Format:
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 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583265

1 INTRODUCTION

Conversational search makes it possible to address users' complex information needs through multi-turn user-system interactions. It provides a brand new search experience for users and is expected to be the next generation of search engines [6]. In contrast to the traditional single-turn ad-hoc search where users mainly use a concise keyword query to express their information needs [25], conversational search systems need to deal with the whole conversational search session, which is composed of multi-turn natural language queries and system responses, for search intent understanding [15, 34, 35]. This tends to be much more difficult than the query understanding in ad-hoc search because the conversational search session is more complex and often contains tremendous noisy tokens irrelevant to understand the current search intent [2, 24]. For example, as shown in Figure 1, the real search intent of the current turn (i.e., q_3) should be "*What happens to water molecules when it freezes?*", while many tokens (e.g., q_1 and r_1) in the conversation context actually would not contribute useful information and can even have a negative effect on the session understanding.

Early conversational query rewriting methods [22, 31, 34] utilize a rewriting model to explicitly reformulate the conversational search session into a new context-independent query rewrite and then feed it into any existing ad-hoc search pipeline to finish conversational search. As illustrated in Figure 1 (a), although the rewrite shows high interpretability, the rewriting model is hard to be optimized towards the downstream search task in such a two-stage (i.e., *rewrite-then-search*) fashion since the discrete generation process in rewriting breaks the backpropagation of gradients, leading to non-ideal performance. Recently, as shown in Figure 1 (b), the end-to-end conversational dense retrieval [21, 35], which is to train a session encoder to directly encode the whole conversational search session into a latent representation, solves the above limitation and generally achieves better search effectiveness.

Nevertheless, most of the existing conversational dense retrieval models [18, 21, 35] are simply learned by fine-tuning the pre-trained ad-hoc encoders on conversational search data [1, 3]. Considering that 1) the original ad-hoc encoders are only pre-trained with short ad-hoc queries and 2) the currently available conversational search data is mainly generated by humans which is not as abundant as

[*] Canada CIFAR AI Chair.

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583265

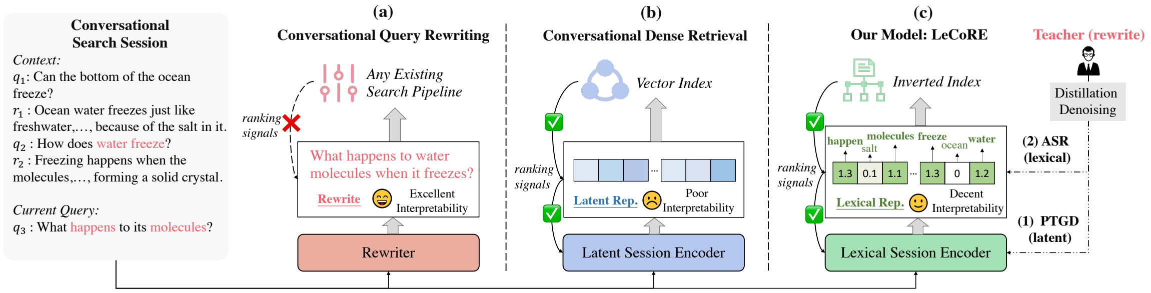
Conversational Search Session
Context:
q_1: Can the bottom of the ocean freeze?
r_1 : Ocean water freezes just like freshwater,…, because of the salt in it.
q_2 : How does water freeze?
r_2 : Freezing happens when the molecules,…, forming a solid crystal.
Current Query:
q_3 : What happens to its molecules?
(a)
Conversational Query Rewriting
Any Existing Search Pipeline
ranking signals
What happens to water molecules when it freezes?
Rewrite
Excellent Interpretability
Rewriter
(b)
Conversational Dense Retrieval
Vector Index
ranking signals
Latent Rep.
Poor Interpretability
Latent Session Encoder
(c)
Our Model: LeCoRE
Teacher (rewrite)
Inverted Index
Distillation Denoising
happen
salt
molecules
freeze
ocean
water
1.3
0.1
1.1
1.3
0
1.2
ranking signals
Lexical Rep.
Decent Interpretability
(2) ASR (lexical)
(1) PTGD (latent)
Lexical Session Encoder

Figure 1: A conceptual illustration for the three types of methods. LeCoRE has multi-level denoising approaches uniformly based on knowledge distillation to achieve more effective end-to-end optimization towards search and decent interpretability.

that in ad-hoc search since real conversational search systems have not been widely deployed in practice, such a simple fine-tuning way is hard to make the ad-hoc encoders well adapted to the noisy and complex conversational search scenario. Besides, compared with the explicit query rewrite, it is hard to know how the model understands the session from the learned latent representation. This is unfriendly to end users because no explicit readable text can be returned to users. It is also unfriendly to the model developers to make targeted improvements when the model fails to correctly understand the user's search intent.

The goal of this work is to improve the end-to-end conversational search model with better context denoising abilities to achieve consistently strong effectiveness as well as decent interpretability. Specifically, we resort to external query rewrites to enhance the model's denoising ability. Although the rewrite may not perfectly represent the real search intent [32], it has at least been refined to contain much less noise than the original session, which can provide valuable guidance for better denoising. To achieve interpretability, we abandon the dense retrieval architecture and instead adopt SPLADE [14, 19], which is a state-of-the-art sparse lexical-based retriever, as our base retrieval model. Compared with previous dense retrieval models, the lexical-based model further transforms latent representation into lexical representation in the vocabulary space. As shown in Figure 1 (c), the activated tokens and weights (e.g., *water* (1.2) and *freeze* (1.3)) of the lexical representation provide a decent hint for understanding the model's behaviors, largely improving the model interpretability while keeping the feasibility of end-to-end training towards search.

To be more specific, we propose a novel sparse **Le**xical-based **Co**nversational **RE**triever (LeCoRE), which extends the popular SPLADE model with two well-matched multi-level denoising methods without introducing any new parameters. The two denoising methods are uniformly based on knowledge distillation from external query rewrites to generate denoised lexical session representation for more effective and interpretable conversational search. Technically, the first Proxy Teacher-guided Denoising (PTGD) method leverages the knowledge of query rewrite to explicitly filter out high-confident noisy tokens at the latent level. And the second Adaptive Sparsity Regularization (ASR) directly takes effect at the final lexical level to improve the critical token weights while ensuring the overall sparsity of the output lexical representation based on the guidance of query rewrite. LeCoRE is trained by integrating the ranking objective and the two proposed denoising objectives using multi-task learning. At search time, LeCoRE encodes the conversational search session into sparse lexical session representation to retrieve passages from the pre-built inverted index.

We conduct extensive experiments on four conversational search benchmark datasets and results show that LeCoRE can consistently outperform state-of-the-art baselines in both normal evaluation and zero-shot evaluation settings. We also perform intuitive case studies to show its decent interpretability for conversational search.

2 RELATED WORK

Conversational Search. Different from traditional ad-hoc search, conversational search faces a complex conversational session understanding problem [15]. In this part, we review a few important conversational query rewriting and conversational dense retrieval methods. Specifically, without generating a completely new query rewrite, Voskarides et al. [31] propose to train a term classifier to select relevant terms from the context and combine them with the current query as the rewrite. Yu et al. [34] use GPT-2 as the rewriter model and propose a rule-based method and a self-learn method to automatically transform ad-hoc search sessions to be the training data of conversational query rewriting. Similarly, Lin et al. [22] demonstrate the effectiveness of T5 [28] to be the rewriter model. Although the high interpretability of conversational query rewriting methods is appealing, the problems of limited and expensive manual rewrite data and the gap between the rewriting and the real target (i.e., ranking) are hard to be solved. A pioneering solution is CONQRR [32], which develops a novel reward function with reinforcement learning to bridge the optimization goal of rewriting and search, but its effectiveness is still under further improvement. From another perspective, Yu et al. [35] first introduce the thinking of dense retrieval into conversational search and propose ConvDR based on knowledge distillation to enable few-shot learning. Concurrently, Lin et al. [21] develop a weak data augmentation method based on the CANARD [12] dataset to generate many session-relevance pairs as the training data for conversational dense retrieval. Although achieving relatively better performance, these models do not explicitly consider the large amount of noise in the session which may hurt the model performance. Mao et al. [24] investigate such a problem of noise and devise an effective context denoising framework COTED to enhance the model's denoising ability under the few-shot scenario, but it needs laborious manual annotations for necessary turns which would be hard to be scaled to real conversational search scenarios. Besides, the interpretability of the learned latent representation is very poor. In contrast to previous methods, we leverage the nature of lexical-based retrievers and design two well-matched multi-level denoising methods based on knowledge distillation to achieve both effective and interpretable conversational search.

PLM-based Lexical Retrieval. Compared with traditional (sparse) lexical retrieval methods (e.g., BM25 [29]), PLM-based lexical retrieval models take advantage of PLMs to inject semantic information into the lexical representation to enhance retrieval effectiveness. Such semantic injection can be done through explicit document expansion (e.g., DocT5Query [26]), document term reweighting (e.g., DeepCT [7]), contextualized vector matching (e.g., (COIL [16] and uniCOIL [20]), or direct lexical representation generation from the contextualized latent representation (e.g., SparTerm [4] and SPLADE [13, 14, 19]. Compared with the latent space of dense retrieval, the vocabulary space of lexical retrieval is much more understandable and thus has decent interpretability, which is particularly helpful in conversational search. To our knowledge, we propose the first sparse lexical-based conversational retriever by adapting the state-of-the-art SPLADE model with tailored denoising extensions for effective and interpretable conversational search.

3 METHODOLOGY

3.1 Preliminaries

3.1.1 ***Task Formulation***. In this paper, we formulate the task of conversational search as finding the relevant passage p from a large passage collection P for the current user query q_k based on the conversational context $C = \{(q_i, r_i)_{i=1}^{k-1}\}$, where q_i and r_i denote the query and the system response of each previous turn, respectively. We call the combination of the current query q_k and its conversational context C as a conversational search session $s_k = \{q_1, r_1, ..., q_k\}$. The challenge of conversational search lies in how to precisely recover the user's real current search intents from the session s_k so as to find her needed passage. For simplicity, we omit the subscript k in the rest of the paper.

3.1.2 ***Existing Two Types of Methods***. To achieve this goal, *conversational query rewriting* methods transform the session s into a de-contextualized ad-hoc query $\hat{q}$ through an expansion model which mainly performs history token selection [23, 31] or a text generation model [28, 32, 34]. By contrast, *conversational dense retrieval* methods map the session and passages into a unified dense latent space to perform denser retrieval without explicit generation of a new query:

$$\mathbf{s} \quad = \quad \mathrm{CSE}(s), \tag{1}$$

$$\mathbf{p} \quad = \quad \mathrm{PE}(p), p \in P, \tag{2}$$

where CSE and PE denote the conversational session encoder and the passage encoder, respectively. P is the passage collection. The matching score is computed as the dot product between the latent session representation $\mathbf{s}$ and the passage representation $\mathbf{p}$. The training usually adopts ranking loss based on contrastive learning:

$$\mathcal{L}_{\text{rank}} = -\log \frac{e^{(\mathbf{s}\cdot\mathbf{p}^+)}}{e^{(\mathbf{s}\cdot\mathbf{p}^+)} + \sum_{p^-\in P} e^{(\mathbf{s}\cdot\mathbf{p}^-)}}, \tag{3}$$

where p^+ and p^- are the relevant and irrelevant passages for the current turn. It is worth noting that, as the passage information has no change in conversational search or traditional ad-hoc search, it is common to directly reuse a well-trained ad-hoc passage encoder and freeze its parameters in training.

3.1.3 ***Recap of SPLADE***. SPLADE [13, 14] is a sparse lexical-based retrieval model. It can encode a text sequence (query or passage) $t = \{t_1, ..., t_N\}$ into a sparse lexical representation $\mathbf{v} \in \mathbb{R}^{|V|}$ by predicting token importance in the whole BERT [10] WordPiece vocabulary space (i.e., $|V| = 30522$) based on the dense latent token representations $\{\mathbf{h_1}, ..., \mathbf{h_N}\}$ generated by the underlying BERT:

$$\mathbf{w_i} \quad = \quad \mathbf{EQh_i} + \mathbf{b}, i \in [1, N], \tag{4}$$

$$\mathbf{v_i} \quad = \quad \log(1 + \mathrm{ReLU}(\mathbf{w_i})), i \in [1, N], \tag{5}$$

$$\mathbf{v} \quad = \quad \mathrm{Pooling}(\mathbf{v_1}, ..., \mathbf{v_N}), \tag{6}$$

where $\mathbf{Q} \in \mathbb{R}^{768\times768}$ and $\mathbf{b} \in \mathbb{R}^{|V|}$ are a trainable transformation matrix and a bias vector, respectively. N is the number of tokens. $\mathbf{E} \in \mathbb{R}^{|V|\times768}$ is the BERT input embedding matrix. The pooling operation can be sum, mean, or max. We adopt the max pooling in this work because: (1) It has a self-denoising effect, i.e., only the maximum weights of each dimension are considered and all the other weights are ignored, which may be helpful to conversational search where the input session is full of noises. (2) The max pooling empirically shows better performance in ad-hoc search than other pooling approaches [13].

In general, the training of SPLADE uses the similar ranking loss (Eq. (3)) as dense retrieval, where the latent representation $\mathbf{s}$ is replaced with the lexical representation $\mathbf{v}$. Besides, to encourage the sparsity of the output lexical representation for fast retrieval, an additional sparsity regularization is incorporated into the final training objective:

$$\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda_q \mathcal{L}^q_{\text{reg}} + \lambda_p \mathcal{L}^p_{\text{reg}}, \tag{7}$$

where $\mathcal{L}_{\text{reg}}$ is a sparsity regularization (e.g., L1 or FLOPS [27]). λ_q and λ_d are the regularization weights for queries and passages. Readers can refer to [14] and [19] for more details of SPLADE.

3.2 Our Model: LeCoRE

LeCoRE is an upgraded version of SPLADE which has two well-matched denoising extensions without introducing any new parameters, including Proxy Teacher-guided Denoising (at the latent level) and Adaptive Sparsity Regularization (at the lexical level), to achieve more effective and interpretable conversational search. Figure 2 shows a training overview of LeCoRE. Based on knowledge distillation from external query rewrites, LeCoRE is trained with two additional objectives (corresponding to the two denoising methods) to enhance its context denoising ability. In the following, we elaborate on these two denoising methods.

3.2.1 ***Proxy Teacher-guided Denoising***. The proxy teacher-guided denoising method helps the model learn to filter out much irrelevant information at the latent representation level to avoid their subsequent ill effects. Specifically, for an input conversational search session s, we have its corresponding query rewrite $\hat{q}$ and a SPLADE query encoder $\mathcal{T}$ which has been well-trained on large-scale ad-hoc search data as the teacher model. Note that the query

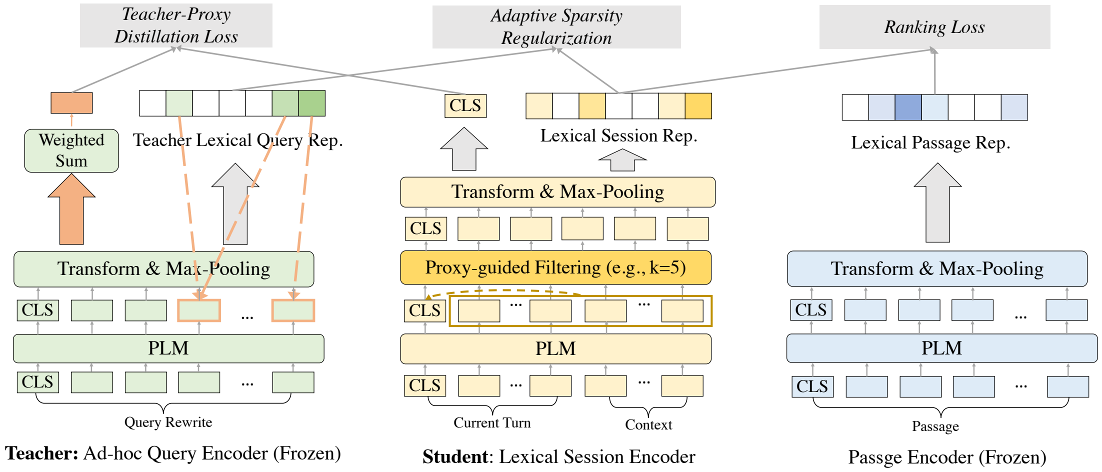
Teacher-Proxy Distillation Loss
Adaptive Sparsity Regularization
Ranking Loss
CLS
Weighted Sum
Teacher Lexical Query Rep.
Lexical Session Rep.
Lexical Passage Rep.
Transform & Max-Pooling
Proxy-guided Filtering (e.g., k=5)
CLS
...
PLM
Query Rewrite
Current Turn
Context
Passage
Teacher: Ad-hoc Query Encoder (Frozen)
Student: Lexical Session Encoder
Passge Encoder (Frozen)

Figure 2: Overview of LeCoRE. At the latent level, we distill the knowledge of the self-denoised teacher latent representation (colored orange) into the CLS (i.e., proxy) of the student to guide the noise filtering. At the lexical level, we refine the lexical session representation by leveraging the teacher lexical query representation to perform adaptive sparsity regularization.

rewrite can be manually rewritten or generated by a conversational query rewriting model. We do not expect it to perfectly express the user's real search intent since it has at least been refined to contain much less noise than the original session, which would be valuable guidance to help us reach better denoising effects. We first input the query rewrite $\hat{q} = \{\hat{q_1}, ..., \hat{q_n}\}$, where n is the number of tokens, into the teacher SPLADE model to the get the output teacher lexical query representation $\mathbf{v}^{\text{t}}$ and the intermediate latent token representations $\{\mathbf{h}^{\text{t}}_1, ..., \mathbf{h}^{\text{t}}_n\}$ of the underlying BERT:

$$\mathbf{h}^{\text{t}}_1, ..., \mathbf{h}^{\text{t}}_n = \mathcal{T}_{\text{BERT}}(\hat{q}), \tag{8}$$

$$\mathbf{v}^{\text{t}} = \mathcal{T}(\hat{q}), \tag{9}$$

where $\mathcal{T}_{\text{BERT}}$ denotes the underlying BERT of the teacher SPLADE model $\mathcal{T}$ and the superscript "t" means "teacher". As the teacher SPLADE model has been well-trained to resolve ad-hoc query rewrite, only very few tokens (i.e., indexes) of $\mathbf{v}^{\text{t}}$ will be activated with non-zero weights. In particular, the max pooling (Eq. (6)) in SPLADE acts as a kind of self-denoising which ensures that each activated token weight in $\mathbf{v}^{\text{t}}$ comes from only one latent token representation. We then leverage this self-denoising effect to obtain a self-denoised teacher latent representation for later distillation. Formally, we use I to denote the set of indexes that are non-zero in $\mathbf{v}^{\text{t}}$ and use $\mathcal{F}$ to denote a mapping function that can map an activated index i into the corresponding unique latent token index $\mathcal{F}(i)$. Then, the self-denoised teacher latent representation $\hat{\mathbf{h}}^{\text{t}}$ is computed as:

$$\hat{\mathbf{h}}^{\text{t}} = \frac{\sum_{i\in I} \mathbf{v}^{\text{t}}[i] \times \mathbf{h}^{\text{t}}_{\mathcal{F}(i)}}{\sum_{i\in I} \mathbf{v}^{\text{t}}[i]}, \tag{10}$$

where $\mathbf{v}^{\text{t}}[i]$ denotes the weight of the i-th index of $\mathbf{v}^{\text{t}}$. Since $\hat{\mathbf{h}}^{\text{t}}$ largely represents the user's search intent at the latent level and has much less noise, it is suitable to be a guide for the student model to filter out noisy input latent token representations.

Specifically, we would only feed the top-K similar latent token representations with respect to $\hat{\mathbf{h}}^{\text{t}}$ into the next layer and filter out the others. The similarity is computed as the dot product between $\mathbf{h}^{\text{t}}$ and $\hat{\mathbf{h}}^{\text{t}}$ and the filtering strength can be controlled by the hyper-parameter K. However, this is only feasible in the training phase because generating query rewrites in the online inference phase will make the student model performance directly related to the quality of the generated rewrites and thus become unstable and uncontrollable. Therefore, we consider that the query rewrite is unavailable in the inference phase (i.e., only the conversational search session is available), so we cannot obtain the corresponding $\hat{\mathbf{h}}^{\text{t}}$ from the teacher. To address this obstacle, we propose to distill the knowledge of $\hat{\mathbf{h}}^{\text{t}}$ into the latent representation of the special CLS token of the student model $\mathcal{S}$ in advance when training, which can then serve as a proxy of the teacher in inference to support the denoising:

$$\text{FC(s)} = [\text{CLS}] \circ q_k \circ r_{k-1} \circ q_{k-1} \circ ... \circ q_1 \circ [\text{SEP}], \tag{11}$$

$$\mathbf{h}^{\text{s}}_{\text{CLS}} = \text{Pool}_{\text{CLS}}(\mathcal{S}_{\text{BERT}}(\text{FC(s)})), \tag{12}$$

$$\mathcal{L}_{\text{tpd}} = \text{MSE}(\mathbf{h}^{\text{s}}_{\text{CLS}}, \hat{\mathbf{h}}^{\text{t}}), \tag{13}$$

where FC (which means *Flat Concatenation*) is a common input format of the conversational search session[1], $\circ$ denotes concatenation, [CLS] and [SEP] are special tokens of BERT, Pool_{CLS} is a pooling operation to only output the latent representation of the CLS token, MSE is the widely used Mean Squared Error loss function, and the superscript "s" means "student".

As shown in the middle of Figure 2, we add a proxy-guided filtering layer in the student model which only allows the CLS token and its top-K similar ones to pass. In this way, a large amount of high-confidence noise in the original input session can be filtered out in advance, which provides a "cleaner pool" for the subsequent generation of high-quality lexical session representation.

3.2.2 ***Adaptive Sparsity Regularization.*** To avoid activating too many irrelevant tokens that hurt the model performance, the

[1] Here we omit the [SEP] token after each query and response for clarity, but we actually add them in our implementation.

Table 1: Statistics of the used conversational search datasets.

Dataset		# Conversations	# Queries(Turns)	# Passages
QReCC	Train	10,823	63,501	54M
	Test	2,775	16,451	
TopiOCQA	Train	3,509	45,450	25M
	Test	205	2,514	
CAsT-19	Test	50	479	38M
CAsT-20	Test	25	208	38M

original SPLADE simply adopts L1 or FLOPS regularization on the output lexical representation to enhance its sparsity. Despite their effectiveness, these regularization methods essentially assume a strong but unrealistic prior that none of the tokens should be activated since all token weights are forced to be zero, which would result in sub-optimal performance.

Instead of using L1 or FLOPS regularization, we propose an simple yet effective Adaptive Sparsity Regularization under the conversational search scenario. Specifically, we use the teacher lexical query representation $\mathbf{v}^{\mathrm{t}}$ as the sparsity prior for the student's lexical session representation $\mathbf{v}^{\mathrm{s}}$ by performing knowledge distillation from $\mathbf{v}^{\mathrm{t}}$ to $\mathbf{v}^{\mathrm{s}}$:

$$\mathbf{v}^{\mathrm{s}} = \mathcal{S}(\mathrm{FC(s)}), \tag{14}$$

$$\mathcal{L}_{\mathrm{asr}} = \sum_{i=1}^{|V|} \left|\mathbf{v}^{\mathrm{s}}[i] - \mathbf{v}^{\mathrm{t}}[i]\right|, \tag{15}$$

where we use Mean Absolute Error (MAE) loss function to encourage sparse solutions. Compared with previous indiscriminate "all-zero regularization" to all samples, we adaptively set better sparsity priors to different samples. Since $\mathbf{v}^{\mathrm{t}}$ is sparse with only very few critical non-zero tokens, using it as the sparsity prior not only ensures the overall sparsity to shield lots of irrelevant tokens but also provides better prior weights for a few important tokens unique to each sample, which helps generate higher-quality lexical session representation. At the same time, in contrast to previous objectives that solely focus on search performance, our adaptive sparsity regularization makes the student model directly learn from the readable rewrite at the lexical level, which is also beneficial to enhance the interpretability of the lexical session representation.

3.2.3 ***Training and Inference.*** LeCoRE is trained with multi-task learning of the ranking loss $\mathcal{L}_{rank}$, the teacher-proxy distillation loss $\mathcal{L}_{\mathrm{tpd}}$, and the adaptive sparsity regularization $\mathcal{L}_{\mathrm{asr}}$:

$$\mathcal{L} = \mathcal{L}_{\mathrm{rank}} + \lambda\mathcal{L}_{\mathrm{tpd}} + \beta\mathcal{L}_{\mathrm{asr}}, \tag{16}$$

where λ and β are hyper-parameters to balance the losses. Following previous work [21, 24, 35], only the student session encoder will be trained while the passage encoder is frozen.

Similar to SPLADE, LeCoRE generates sparse lexical passage representations offline and stores them into an inverted index that has $|V|$ (i.e., 30522) keys. At the search time, LeCoRE encodes the input conversational search session into a lexical session representation and uses the standard BM25 formula for retrieval, where the retrieval score is actually equal to the dot product between the lexical session representation and the lexical passage representation.

4 EXPERIMENTAL SETUP

4.1 Datasets and Evaluation Metrics

4.1.1 ***Datasets.*** We evaluate LeCoRE in two experimental settings. The first is the normal training-test setting conducted on two popular conversational search datasets: **QReCC** [3] and **TopiOCQA** [1]. The training-test split is provided in the original datasets. We randomly select a validation set containing 500 query turns from the training set for parameter tuning. The second is the zero-shot evaluation setting, where we evaluate the models which have been trained on the QReCC training set, on two widely used small conversational search test sets: **CAsT-19** [8] and **CAsT-20** [9]. Note that QReCC, CAsT-19, and CAsT-20 provide manual oracle rewrites while TopiOCQA only provides rewrites generated by a T5-based rewriting model [28]. Table 1 presents the dataset statistics and further dataset details are provided in Appendix A.

4.1.2 ***Evaluation Metrics.*** Following previous studies [1, 3, 24, 35], we adopt four widely used metrics[2]: MRR, NDCG@3, Recall@10, and Recall@100 to comprehensively evaluate the retrieval performance. The metrics are calculated using the `pytrec_eval` tool [30]. Significance tests are conducted using paired t-tests at p < 0.05 level.

4.2 Baselines

For clarity, we describe a conversational search system from two aspects: (1) the used *retriever* and (2) the *input type*. The retriever is to encode a text sequence (e.g, a single query (rewrite), a conversational search session, or a passage) and perform retrieval, and the input type specifies the format of the text sequence fed into the retriever.

For retrievers, we include: (1) **ANCE** [33]: A state-of-the-art BERT-based ad-hoc dense retriever trained with dynamic global hard negatives. (2) **Conv-ANCE** [33]: ANCE fine-tuned on conversational search data only using the ranking loss (Eq. (3)). (3) **ConvDR** [35]: ANCE fine-tuned on conversational search data using knowledge distillation between the query rewrite representation and the latent session representation. (4) **T5-Encoder** [28]: The encoder of T5 [28] fine-tuned on ad-hoc search data. (5) **BM25** [29]: A classical lexical-based retriever. (6) **SPLADE** [14]: A strong lexical-based retriever which is also the base model of LeCoRE. (7) **Conv-SPLADE** [14]: SPLADE (with L1 regularization) fine-tuned on conversational search data only using the ranking loss (Eq. (3))

For input types, we include: (1) **T5QR** [22]: The query rewrite generated by a T5-based conversational query rewriting model. (2) **CONQRR** [32]: The query rewrite generated by a reinforcement learning-based conversational query rewriting model. Note that it adopts the BM25 and T5-Encoder as the retrievers in their original paper. (3) **FC**: Flat Concatenation (i.e., Eq (11)), which is a common input format of the conversational search session. (4) **Raw**: The query of the current turn. (5) **Manual**: The manual oracle rewrite of the current turn.

[2] In particular, we deem relevance scale ≥ 2 as positive for MRR and recall on the CAsT-20 dataset following the official evaluation setting [9].

Table 2: Results of normal evaluation. The best results are in bold. † indicates significant improvements over all baselines except CONQRR. Using FC as the input of BM25 is impractical because FC is too long and makes the search latency unbearable.

Rep. Type	Retriever	Input	QReCC				TopiOCQA			
			MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Latent	ANCE	T5QR	34.5	31.8	53.1	72.8	23.0	22.2	37.6	54.4
	T5-Encoder	CONQRR	41.8	-	65.1	84.7	-	-	-	-
	Conv-ANCE	FC	47.1	45.6	71.5	87.2	22.9	20.5	43.0	71.0
	ConvDR	FC	38.5	35.7	58.2	77.8	27.2	26.4	43.5	61.1
Lexical	BM25	T5QR	33.4	30.2	53.8	86.1	12.5	10.9	23.5	46.7
	BM25	CONQRR	38.3	-	60.1	88.9	-	-	-	-
	SPLADE	T5QR	42.4	39.4	62.1	82.9	30.6	29.5	46.4	62.8
	Conv-SPLADE	FC	50.0	46.6	69.9	87.8	30.7	29.5	52.1	72.0
	LeCoRE (Ours)	FC	**51.1**†	**48.5**†	**73.9**†	**89.7**†	**32.0**†	**31.4**†	**54.3**†	**73.5**†
For Reference										
Latent	ANCE	Raw	10.2	9.3	15.7	22.7	4.1	3.8	7.5	13.8
	ANCE	FC	42.5	39.8	62.6	79.3	10.3	9.1	19.1	35.7
	ANCE	Manual	38.4	35.6	58.6	78.1	N/A	N/A	N/A	N/A
Lexical	BM25	Raw	6.5	5.5	11.1	21.5	2.1	1.8	4.0	9.2
	BM25	Manual	39.7	36.2	62.5	98.5	N/A	N/A	N/A	N/A
	SPLADE	Raw	13.4	12.3	19.8	28.0	5.7	5.2	9.3	15.8
	SPLADE	FC	48.5	45.9	67.3	84.0	15.5	14.1	25.8	47.2
	SPLADE	Manual	48.0	45.0	69.7	88.7	N/A	N/A	N/A	N/A

4.3 Implementations

4.3.1 ***LeCoRE****.* We implement LeCoRE based on the excellent public repository of SPLADE[3] using the PyTorch and Huggingface libraries. The experiments are conducted on four Nvidia Tesla v100 32G GPUs. Specifically, we adopt the Adam optimizer with a learning rate of 2e-5 and a total batch size of 128. We set the number of retained latent token representations K to 64. The loss balance weights λ and β are set to 0.1 and 1e-4, respectively. On QReCC, we use its provided manual oracle rewrite as $\hat{q}$. While on TopiOCQA which does not provide the oracle rewrites, we use its provided T5QR as $\hat{q}$. The lengths of the query, flat concatenation, and passage are truncated into 32, 256, and 256, respectively. In particular, following previous work [24, 35], only the previous queries and the last response r_{k-1} (i.e., the canonical passage) are included in FC on CAsT-20 and the length of r_{k-1} is restricted to 128. The settings of BM25 are $k_1 = 0.9, b = 0.4$ on TopiOCQA and $k_1 = 0.82, b = 0.68$ on the other datasets. The dense retrieval is performed using Faiss [17] with brute force. Code is released at https://github.com/kyriemao/LeCoRE.

4.3.2 ***Baselines****.* All ANCE and SPLADE models used in baselines and LeCoRE are uniformly initialized using the same checkpoints pre-trained on the MS MARCO ad-hoc passage retrieval dataset[4] for fair comparisons. For the ranking loss, we adopt the in-batch negative sampling plus one hard negative sample randomly selected from Top-50 retrieved passages by BM25. The T5 model used in T5QR is fine-tuned on the training data of QReCC with a batch size of 32. For ConvDR, we use T5QR as the pseudo manual oracle rewrites on TopiOCQA to enable its training of knowledge distillation. For CONQRR, we directly replicate their experimental results on QReCC because they have not released the code and their experimental settings are similar to us.

Table 3: Results of zero-shot evaluation. The best results are in bold. † indicates significant differences between LeCoRE and the second-best baselines.

Retriever	Input	CAsT-19		CAsT-20	
		NDCG@3	R@100	NDCG@3	R@100
BM25	T5QR	25.8	37.3	14.1	22.5
ANCE	T5QR	41.7	33.2	29.9	35.3
SPLADE	T5QR	**46.5**	46.9	**32.8**	39.8
ConvDR	FC	43.9	32.2	32.4	33.8
Conv-ANCE	FC	34.1	29.2	27.5	36.2
Conv-SPLADE	FC	42.0	48.3	28.1	44.5
LeCoRE (Ours)	FC	42.2	**49.4**†	29.0	**46.7**†
For Reference					
BM25	Manual	30.9	44.8	24.0	39.5
ANCE	FC	26.9	25.3	15.7	25.6
ANCE	Manual	46.1	38.1	42.2	46.5
SPLADE	FC	34.6	38.4	19.2	32.3
SPLADE	Manual	56.9	54.9	48.7	61.9

5 EXPERIMENTAL RESULTS AND ANAYLSIS

5.1 Normal Evaluation

The experimental results on QReCC and TopiOCQA are shown in Table 2, where we have the following observations:

(1) LeCoRE consistently outperforms all the other compared baselines across all four metrics and the two datasets, which demonstrates its superior effectiveness in conversational search. In particular, we observe 4.3% and 6.4% NDCG@3 relative gains over the second-best results on QReCC and TopiOCQA, respectively. This proves the strong ability of LeCoRE in the top ranks, which is particularly desired in conversational search. The superior effectiveness of LeCoRE can be attributed to the following two aspects. (i) SPLADE tends to be a more effective retriever than ANCE and BM25, which supports the superiority of LeCoRE. (ii) Our proposed two denoising methods enhance the context denoising ability of LeCoRE to achieve better performance (than Conv-SPLADE).

[3] https://github.com/naver/splade

[4] https://microsoft.github.io/msmarco/

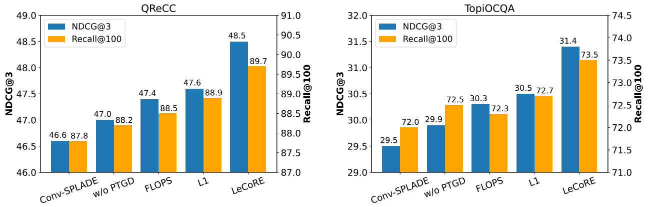

Figure 3: Performance comparisons among several variants.

(2) Through the comparisons among the baselines whose inputs are flat concatenation, we find that using the ranking loss and the knowledge distillation loss can both improve the model performance. In particular, we find that leveraging manual oracle rewrites, which express the users' search intent from the view of humans, may not achieve the best search performance. For example, (i) ConvDR performs better than Conv-ANCE on TopiOCQA while is worse on QReCC and (ii) using Manual as the input performs even worse than FC no matter adopting SPLADE or ANCE on QReCC. This is because the manual oracle rewrite is not guaranteed to be the best from the view of search.

(3) T5QR achieves significantly better performance than FC for both SPLADE and ANCE on TopiOCQA, which proves that T5 is effective for conversational query rewriting. However, as the training of the T5 rewriter is based on the manual rewrite which may not be ideal for passage ranking as illustrated in (2), T5QR does not show very good performance on QReCC. In contrast, an important advantage of our LeCoRE is that, even if the effect of the query rewrite is not very ideal that using it alone cannot bring improvements, LeCoRE can still leverage it to achieve better results thanks to its multi-task way of learning.

5.2 Zero-shot Evaluation

We also conduct zero-shot testing on CAsT datasets to evaluate the transferring abilities of the compared models which are trained on QReCC. We report the results of NDCG@3 and Recall@100 in Table 3 and we have the following findings:

(1) LeCoRE outperforms all the other baselines in terms of Recall@100 and achieves the third-best NDCG@3 results on both two datasets, which demonstrates the strong transferring ability of LeCoRE. Compared with Conv-SPLADE, LeCoRE consistently maintains its superiority in the zero-shot setting, demonstrating the generalization of our newly designed denoising methods.

(2) Jointly analyzing Table 2 and Table 3, we observe that T5QR shows to be more effective on the CAsT datasets, especially in terms of NDCG@3, which demonstrates that conversational query rewriting methods also have potential to achieve better performance than conversational dense retrieval methods for top passage ranking.

(3) From the results for reference, we find that the qualities of manual oracle rewrites of the CAsT datasets are higher than those of QReCC since Manual clearly achieves better performance than FC. Compared with their no-training counterparts, the better results of ConvDR, Conv-ANCE, and Conv-SPLADE demonstrate the effectiveness of the knowledge distillation and the ranking loss again. Besides, we also observe that the SPLADE retriever consistently keeps its advantage for conversational search on the CAsT datasets compared with BM25 and ANCE.

5.3 Ablation Study

In this section, we investigate the effects of the proxy teacher-guided denoising and the adaptive sparsity regularization proposed in LeCoRE. Specifically, we build the following variants of LeCoRE for comparisons: (1) **L1**, which replaces the adaptive sparsity regularization with L1 regularization. (2) **FLOPS**, which replaces the adaptive sparsity regularization with FLOPS regularization. (3) **w/o-PTGD**, which removes the proxy teacher-guided denoising from LeCoRE, i.e., setting λ to 0. (4) **Conv-SPLADE**, which is equivalent to removing PTGD and replacing ASR with L1 in LeCoRE.

Performance comparisons are shown in Figure 3. We observe a similar performance ranking on both datasets. Specifically, Conv-SPLADE, which is not equipped with our proposed two denoising methods, performs the worst among all the compared models, and the complete LeCoRE clearly outperforms all the others. Meanwhile, we can also observe performance improvements when only one of the denoising methods is used (i.e., L1, FLOPS, and w/o PTGD). These results verify the necessity of the proposed denoising methods for more effective conversational search.

5.4 Lexical Representation Analysis

In this section, we investigate the denoising effects of LeCoRE by analyzing the activated tokens in the output lexical session representation on the test sets of QReCC and CAsT-20.

5.4.1 *The Number of Activated Tokens.*

We count the number of activated tokens of SPLADE, Conv-SPLADE, and LeCoRE, all using FC as the input. Meanwhile, we also count the number of activated tokens of the teacher (i.e., SPLADE using Manual as the input type) for reference. The average numbers of tokens of FC for QReCC and CAsT-20 are 129.2 and 138.3, respectively. As shown in Figure 4, the good performance of the teacher indicates that the number of necessary activated tokens may not be large. Evidently, the original SPLADE, which is not fine-tuned on conversational search data, has many more activated tokens and performs significantly worse than Conv-SPLADE and LeCoRE, indicating that a large amount of noise in the original input session can seriously impair the model performance. Through the comparisons between Conv-SPLADE and LeCoRE, we find that the denoising methods in LeCoRE can further decrease the number of activated tokens while improving the model performance, which demonstrates the positive denoising effects of the two proposed denoising methods to a certain extent.

5.4.2 *Token Overlap with the Teacher.*

Then, we view the teacher's activated tokens as the "gold" (although it does not perform the best) and calculate the precision, recall, and macro F1 of the overlapped activated tokens of the three models w.r.t. to the gold. Results are shown in Table 4. We find that, while the original SPLADE has the largest recall, its precision is very low, which indicates that it suffers a lot from the noise. Compared with Conv-SPLADE, LeCoRE shows higher values on all three metrics, demonstrating that LeCoRE absorbs the teacher's knowledge to have an all-around more overlap with the teacher to improve its effectiveness. Meanwhile, LeCoRE adopts a multi-task training method to not fully imitate the teacher, thus avoiding the undesired part of the teacher which may degrade the ranking performance. In addition,

Table 4: Comparisons of Precision, Recall, and Macro F1 of the activated tokens with respect to those of the teacher.

Variants	QReCC			CAsT-20		
	P	R	F1	P	R	F1
SPLADE	0.20	0.89	0.32	0.14	0.86	0.22
Conv-SPLADE	0.31	0.55	0.38	0.26	0.55	0.35
LeCoRE	0.40	0.63	0.45	0.30	0.65	0.40
SPLADE (Manual)	1.0	1.0	1.0	1.0	1.0	1.0

more overlaps with the teacher representation generated from the readable rewrite also indicate better interpretability to some extent.

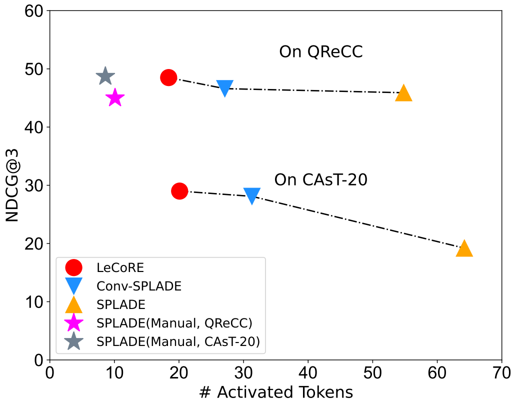

Figure 4: Relationships between the number of activated tokens and NDCG@3 of SPLADE (FC), Conv-SPLADE, LeCoRE, and SPLADE with Manual input (i.e. the teacher model).

5.5 Case Study: Interpretability

Inherited from SPLADE and enhanced by the two proposed denoising methods, LeCoRE achieves decent interpretability for conversational search. The activated tokens and weights provide a hint for us to understand how the model understands the conversational search session. In this section, we show some concrete examples in Table 5 to intuitively demonstrate the interpretability of LeCoRE.

In the first example, LeCoRE successfully recovers the user's search intent by activating the corresponding tokens (e.g., "heater", "cheese", "steak", "different", and "normal") with relatively large weights. "philadelphia" and "inventing" are also weakly related to the current search intent as the regular cheesesteak is invented by two Philadelphians. Besides, we show the corresponding activated token weights of Conv-SPLADE and SPLADE in Table 6 of Appendix B due to the limited space. We find that the activated tokens of Conv-SPLADE and SPLADE contain more unreasonable tokens and weights from the view of session understanding (e.g, only 0.36 weight for the important token "cheese" in Conv-SPLADE), which intuitively demonstrates the denoising effects in LeCoRE for generating more interpretable lexical session representation.

While in the second example, LeCoRE does not put the desired passage at the top rank because it fails to accurately recover the search intent "UNLV" (abbreviation of "University of Nevada, Las Vegas"), which specifies the transfer of Odom. This is difficult since UNLV not appears in the context and can only be implicitly inferred based on the background knowledge of Odom.

Table 5: Two concrete examples. The blue tokens stand for the search intents correctly predicted by LeCoRE while the red tokens stand for the unrecovered intents.

ID	Conversational Search Session	Token weights of LeCoRE
1	**Context**: q_1: What ingredients are in a philly cheesesteak? r_1: Philly Cheesesteak is a sandwich made with super thinly sliced ribeye steak, caramelized onion, and provolone cheese. q_2: Did the sandwich come from Philadelphia? r_2: A popular regional fast food, the cheesesteak has its roots in the U.S. city of Philadelphia, Pennsylvania. q_3: Who invented the sandwich? r_3: Philadelphians Pat and Harry Olivieri are often credited with inventing the cheesesteak by serving chopped steak on an Italian roll in the early 1930s. q_4: What kind of variations are there? r_4: Variations of the cheesesteak include the chicken cheesesteak, the pizza steak, the cheesesteak hoagie, the vegan cheesesteak, and the Heater. **Current Query**: q_5: How is the heater different? **Manual Oracle Rewrite**: $\hat{q}_5$: How is the heater different from a regular cheesesteak?	('heat', 2.09), ('cheese', 1.62), ('phil', 1.51), ('steak', 1.48), ('##er', 1.33), ('different', 1.29), ('variations', 0.84), ('chicken', 0.73), ('philadelphia', 0.63), ('normal', 0.52), ('sandwich', 0.47), ('##venting', 0.27), ('roots', 0.10), ('1930s', 0.08), ('roll', 0.06), ('##eye', 0.06), ('pizza', 0.03)
2	**Context**: q_1: Who was Lamar Odom inspired by? r_1: Lamar Odom drew inspiration from his maternal grandmother, a nurse who had raised five children and returned to school to earn her degree in 1980 at the age of 56. **Current Query**: q_2: Where did he transfer to? **Manual Oracle Rewrite**: $\hat{q}_2$: Where did Lamar Odom transfer to from UNLV?	('##dom', 2.07), ('o', 1.68), ('transfer', 1.18), ('1980', 1.16), ('inspired', 1.02), ('lamar', 0.65), ('inspiration', 0.57), ('was', 0.4), ('56', 0.39), ('1978', 0.31), ('school', 0.29), ('nurse', 0.25), ('degree', 0.23), ('maternal', 0.11), ('1976', 0.07)

6 CONCLUSION

In this paper, we propose the first sparse lexical-based conversational retriever LeCoRE. By extending SPLADE with two well-matched multi-level denoising methods, LeCoRE effectively filters out many noisy signals in the raw conversational search session to get higher-quality lexical session representation. Extensive experiments conducted on four public datasets demonstrate that LeCoRE outperforms state-of-the-art baselines and achieves decent interpretability. In the future, we plan to investigate better ways of integrating lexical/latent representation and explicit query rewrites to further improve both effectiveness and interpretability.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by the National Natural Science Foundation of China No. 62272467 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China NO. 22XNKJ34, and Public Computing Cloud, Renmin University of China. The work was partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE.

REFERENCES

[1] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain Conversational Question Answering

with Topic Switching. *Transactions of the Association for Computational Linguistics* 10 (2022), 468–483.

[2] Mohammad Aliannejadi, Manajit Chakraborty, Esteban Andrés Ríssola, and Fabio Crestani. 2020. Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. 33–42.

[3] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. In *NAACL-HLT*. Association for Computational Linguistics, 520–534.

[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. *arXiv preprint arXiv:2010.00768* (2020).

[5] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *EMNLP*. Association for Computational Linguistics, 2174–2184.

[6] J Shane Culpepper, Fernando Diaz, and Mark D Smucker. 2018. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR Forum*, Vol. 52. ACM New York, NY, USA, 34–90.

[7] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Term Weighting For First Stage Passage Retrieval. In *SIGIR*. ACM, 1533–1536.

[8] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The conversational assistance track overview. In *In Proceedings of TREC*.

[9] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. CAsT 2020: The Conversational Assistance Track Overview.. In *In Proceedings of TREC*.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

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

[12] Ahmed Elgohary, Denis Peskov, and Jordan L. Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 5917–5923.

[13] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *SIGIR*. ACM, 2353–2359.

[14] 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*. 2288–2292.

[15] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural Approaches to Conversational Information Retrieval. *arXiv preprint arXiv:2201.05176* (2022).

[16] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. In *NAACL-HLT*. Association for Computational Linguistics, 3030–3042.

[17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* 7, 3 (2021), 535–547.

[18] Sungdong Kim and Gangwoo Kim. 2022. Saving Dense Retriever from Shortcut Dependency in Conversational Search. *arXiv preprint arXiv:2202.07280* (2022).

[19] Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. In *SIGIR*. ACM, 2220–2226.

[20] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *CoRR* abs/2106.14807 (2021).

[21] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized Query Embeddings for Conversational Search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[22] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909* (2020).

[23] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–29.

[24] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval. In *Proceedings of the 45th International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*.

[25] 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*.

[26] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6 (2019).

[27] Biswajit Paria, Chih-Kuan Yeh, Ian En-Hsu Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing FLOPs to Learn Efficient Sparse Representations. In *ICLR*. OpenReview.net.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[29] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[30] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. In *SIGIR*. ACM.

[31] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*. 921–930.

[32] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2022. CONQRR: Conversational Query Rewriting for Retrieval with Reinforcement Learning. (2022).

[33] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[34] 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 (SIGIR)*. 1933–1936.

[35] 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 (SIGIR)*.

A DETAILS OF DATASETS

In this section, we introduce more details of the four used conversational search datasets.

QReCC is a large-scale dataset for conversational question answering, which contains 14K information-seeking conversations with 80K query-answer pairs originated from the training set of CAsT-19 [8], QuAC [5], and NQ [5] with manually generated follow-up queries. Each query has a response answer and a corresponding human rewrite. The entire text corpus for retrieval includes 54M passages and the query-passage relevance is labeled through a heuristic span-matching method based on the answer.

TopiOCQA contains around 4K information-seeking conversations in open domains based on the Wikipedia corpus. Each conversation is generated by using a real search query in NQ [5] as the start query and replenishing the subsequent turns in a wizard-of-oz fashion. It additionally provides the topic information for each query but does not provide the human rewrite. The text corpus contains 25M passages and the relevance is labeled by humans.

CAsT-19 and **CAsT-20** are two widely used conversational search datasets released by TREC Conversational Assistance Track (CAsT). There are only 50 and 25 human-written information-seeking conversations in CAsT-19 and CAsT-20, respectively, so they are hard to support training and are suitable to be used as the evaluation datasets. The query turns in CAsT-19 can only depend on the previous query turns. While in CAsT-20, the query turns may also depend on the previous system response. Each query turn in both CAsT-19 and CAsT-20 has a corresponding human rewrite and CAsT-20 additionally provides a canonical response passage for each query turn. The text corpus consists of 38M passages from MS MARCO [25] and TREC Complex Answer Retrieval [11]. More fine-grained query-passage relevance labels are generated by the experts of TREC.

B SUPPLEMENT OF CASE STUDIES

We show the comparison among the activated tokens of LeCoRE, Conv-SPLADE, and SPLADE for the first example in Table 6 to supplement the Section 5.5.

Table 6: The comparison among the activated tokens of LeCoRE, Conv-SPLADE, and SPLADE. The blue tokens stand for the search intents correctly predicted by the compared models.

Conversational Search Session	Activated Tokens and Weights		
	LeCoRE	Conv-SPLADE	SPLADE
Context: q_1: What ingredients are in a philly cheesesteak? r_1: Philly Cheesesteak is a sandwich made with super thinly sliced ribeye steak, caramelized onion, and provolone cheese. q_2: Did the sandwich come from Philadelphia? r_2: A popular regional fast food, the cheesesteak has its roots in the U.S. city of Philadelphia, Pennsylvania. q_3: Who invented the sandwich? r_3: Philadelphians Pat and Harry Olivieri are often credited with inventing the cheesesteak by serving chopped steak on an Italian roll in the early 1930s. q_4: What kind of variations are there? r_4: Variations of the cheesesteak include the chicken cheesesteak, the pizza steak, the cheesesteak hoagie, the vegan cheesesteak, and the Heater. **Current Query**: q_5: How is the heater different? **Manual Oracle Rewrite**: $\hat{q}_5$: How is the heater different from a regular cheesesteak?	(’heat’, 2.09), (’cheese’, 1.62), (’phil’, 1.51), (’steak’, 1.48), (’##er’, 1.33), (’different’, 1.29), (’variations’, 0.84), (’chicken’, 0.73), (’philadelphia’, 0.63), (’normal’, 0.52), (’sandwich’, 0.47), (’##venting’, 0.27), (‘roots’, 0.10), (’1930s’, 0.08), (’roll’, 0.06), (’##eye’, 0.06), (’pizza’, 0.03)	(’heat’, 2.03), (’phil’, 2.01), (’##ste’, 1.62), (’different’, 1.51), (’##er’, 1.48), (’pat’, 1.26), (’variations’, 1.05), (’##venting’, 1.02), (’philadelphia’, 0.92), (’##zen’, 0.83), (’1930s’, 0.76), (’##agi’, 0.62), (’##leg’, 0.54), (’name’, 0.50), (’##lone’, 0.44), (’chicken’, 0.42), (’##bra’, 0.40), (’cheese’, 0.36), (’latitude’, 0.36), (’birth’, 0.35), (’include’, 0.33), (’##sur’, 0.28), (’,’, 0.28), (’fast’, 0.28), (’nationality’, 0.27), (’vega’, 0.21), (’##quin’, 0.19), (’##eye’, 0.17), (’##cy’, 0.16), (’motown’, 0.15), (’early’, 0.1), (’traditional’, 0.08), (’##bil’, 0.06), (’how’, 0.04), (’sandwich’, 0.02)	(’invented’, 2.1), (’philadelphia’, 2.01), (’sandwich’, 1.74), (’roots’, 1.73), (’heat’, 1.7), (’variations’, 1.69), (’chopped’, 1.69), (’vega’, 1.67), (’steak’, 1.66), (’##venting’, 1.64), (’regional’, 1.59), (’pizza’, 1.59), (’phil’, 1.49), (’pennsylvania’, 1.46), (’ingredients’, 1.45), (’chicken’, 1.44), (’who’, 1.4), (’1930s’, 1.4), (’roll’, 1.39), (’onion’, 1.38), (’fast’, 1.38), (’popular’, 1.34), (’cheese’, 1.32), (’italian’, 1.32), (’include’, 1.29), (’pat’, 1.27), (’##ak’, 1.21), (’different’, 1.19), (’how’, 1.15), (’olivier’, 1.14), (’made’, 1.1), (’sliced’, 1.09), (’##ste’, 1.09), (’##agi’, 1.08), (’early’, 1.06), (’serving’, 0.99), (’##eye’, 0.98), (’super’, 0.97), (’often’, 0.94), (’food’, 0.9), (’was’, 0.89), (’credited’, 0.88), (’come’, 0.86), (’thin’, 0.86), (’city’, 0.85), (’##er’, 0.81), (’u’, 0.78), (’harry’, 0.76), (’##lone’, 0.73), (’##ly’, 0.66), (’##mel’, 0.64), (’rib’, 0.61), (’pro’, 0.58), (’origins’, 0.54), (’##ns’, 0.48), (’from’, 0.44), (’ho’, 0.41), (’##vo’, 0.4), (’##ized’, 0.37), (’with’, 0.34), (’has’, 0.33), (’there’, 0.3), (’cara’, 0.3), (’on’, 0.28), (’kind’, 0.23), (’in’, 0.22), (’##i’, 0.08), (’##n’, 0.07), (’sub’, 0.04)