ESPRESSO: An Effective Approach to Passage Retrieval for High-Quality Conversational Recommender Systems

Taeho Kim[1], Hyeongjun Jang[2, 3*], Juwon Yu[2, 3*], Taeuk Kim[1], Hyunyoung Lee[3], Ji-hui Im[3], Sang-Wook Kim[1†]

[1]Department of Computer Science, Hanyang University, South Korea
[2]Department of Artificial Intelligence Application, Hanyang University, South Korea
[3]KT Corporation, South Korea
{hirooms2, kimtaeuk, wook}@hanyang.ac.kr, {hyeongjun.jang, juwon1.yu, lee.hyunyoung, jihui.im}@kt.com

Abstract

Conversational Recommender Systems (CRS) aim to provide tailored recommendation responses via a chat interface, including both the user's preferred item and its accompanying explanation. However, due to its generative nature, CRS are prone to responding with factually incorrect explanations (*i.e.*, hallucinations). To solve this problem, we propose incorporating a passage retrieval module into CRS with the objective of enhancing the factuality and informativeness of system responses. Specifically, we outline essential directions for employing a passage retrieval module in CRS to address the following critical issues: (1) the risk of passage retrieval not aligning with the user preference; (2) the absence of supervision for training a passage retrieval module. As a solution, we introduce ESPRESSO, a novel passage retrieval approach for CRS, to effectively tackle the above issues with two core ideas: *adaptive item selection* and *relevance-based groupwise learning*. Our extensive experiments show that ESPRESSO effectively resolves issues, achieving up to 36% higher Hit@3 accuracy than the best of 8 competing methods. Additionally, we verify that leveraging passages retrieved by ESPRESSO significantly improves the response quality of CRS.

Code — https://github.com/Bigdasgit/ESPRESSO
Appendix — https://github.com/Bigdasgit/ESPRESSO/blob/main/AAAI_2025_appendix.pdf

Introduction

Conversational Recommender Systems (CRS) feature a chat interface that permits users to naturally express their preferences, fostering a user-friendly experience (Li et al. 2018; Zhou et al. 2020; Kim et al. 2023). The implementation of CRS usually consists of two core components: (1) the *recommendation module*, which aims to predict the items that a user is likely to prefer, and (2) the *generation module*, responsible for delivering recommendation responses (REC-responses, in short) that contain not only a recommended item but also an accompanying explanation for the item.

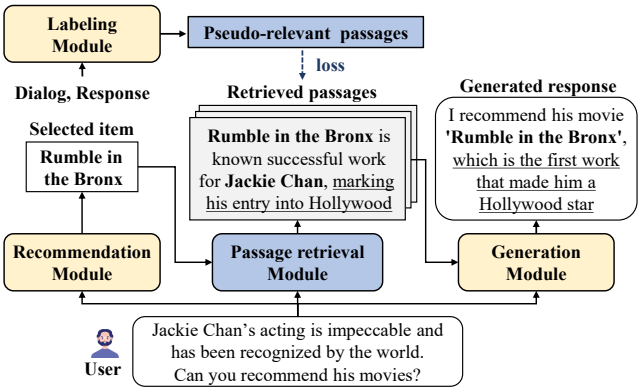

Figure 1: A retrieval-augmented CRS framework that follows our proposed directions.

The advancement of language models (Lewis et al. 2020a; Touvron et al. 2023) has made CRS increasingly feasible. However, since language models primarily rely on their intrinsic knowledge for generating responses (Lewis et al. 2020b), they are often prone to providing factually inaccurate explanations regarding recommended items (*i.e.*, hallucinations), particularly when they lack sufficient knowledge about the items. To address this issue, one could consider employing *retrieval-augmentation* strategies (Lewis et al. 2020b). Specifically, given a dialog, these strategies are designed to initially retrieve passages describing the features of recommended items and subsequently generate a REC-response by drawing upon the retrieved passages.

However, we point out that existing passage retrieval methods studied in other fields (Karpukhin et al. 2020; Sun et al. 2023; Wu et al. 2023) are insufficient for properly serving CRS in their original forms. This is because the primary objective of CRS is to present items that the user prefers along with suitable explanations (Li et al. 2018; Wang et al. 2022), but existing passage retrieval methods are neither designed nor trained to retrieve passages that align well with user preferences. Therefore, for the first time in the CRS domain, we *explore directions for effective passage retrieval to help CRS generate high-quality REC-responses* as follows.

*This work was done while Hyeongjun Jang and Juwon Yu were at Hanyang University. They are now working at KT.
†Corresponding authors.

(Direction-1) The passage retrieval module for CRS should retrieve passages related to the items that the user prefers. For instance, if a user expresses a positive impression of 'Jackie Chan' in a dialog, the passage retrieval module should retrieve passages related to his works (*e.g.*, 'Rumble in the Bronx') to improve the quality of the system's response. Therefore, as shown in Figure 1, we claim that the passage retrieval module *should utilize the items* selected by the recommendation module as its *additional input*.

(Direction-2) When generating a REC-response, the generation module in CRS needs to explain only the features of the recommended items that the user wants to know (Lu et al. 2021; Park et al. 2022). To support this, the passage retrieval module should be trained to retrieve passages that describe relevant features for the recommended items among the many passages in the corpus. However, *manually labeling* which passages are relevant for generating REC-responses *requires significant human labor and cost* (Sun et al. 2023), especially in CRS, where REC-responses often involve lengthy descriptions. Therefore, as shown in Figure 1, we advocate training the passage retrieval module by using *pseudo-relevant passages*, which are determined as relevant by the labeling module.

To sum up, the passage retrieval module in CRS should utilize: (1) the output of the recommendation module as additional input and (2) the pseudo-relevant passages from the labeling module as training labels. This approach is generally promising for improving passage retrieval accuracy. However, we find that we occasionally miss opportunities for *further improvement* if we naively utilize the results of other modules (*i.e.*, the recommendation module and the labeling module). Therefore, following our directions for passage retrieval, we propose two ideas to *robustly* utilize the results of other modules by addressing potential errors arising from their mistakes.

(Idea-1) We can simply follow Direction-1 by using the top-1 item predicted by the recommendation module as an additional input for the passage retrieval module. However, if the recommendation module happens to select the top-1 item that does *not align* with user preferences, the passage retrieval module, depending only on this item, would fail to retrieve the top-K passages related to the user's preferred item. Hence, we propose a novel method called *adaptive item selection*, which selects a *variable number* of items based on the confidence of the recommendation module. Specifically, if the confidence of the recommendation module for the top-1 item is high enough, only this item is selected. Otherwise, the recommendation module selects more items until sufficient confidence is ensured. By doing so, we can increase the likelihood of including the user's preferred item while excluding less-preferred items.

(Idea-2) To follow Direction-2, we can identify passages with the highest relevance to the ground-truth responses, which are available only during training, as pseudo-relevant. Then, as in existing passage retrieval methods for model training (Karpukhin et al. 2020; Wu et al. 2023), we can use contrastive learning that individually increases the score of each pseudo-relevant passage than that of other passages. However, we find that some pseudo-relevant passages might be *mislabeled* because the labeling module cannot always be perfect. This risk necessitates a more-sophisticated learning method to robustly train the passage retrieval module. Therefore, we propose relevance-based groupwise learning, based on the intuition that the likelihood of all elements within a group of pseudo-relevant passages being irrelevant is much lower than that of a single pseudo-relevant passage. Specifically, this method first groups the pseudo-relevant passages based on their relevance, then trains the model to ensure that the average retrieval score of the grouped passages is higher than that of other passages.

In summary, we propose ESPRESSO, short for Enhanced paSsage retrieval apPROach via adaptivE item Selection and relevance-baSed grOupwise learning, for the effective passage retrieval in CRS. To the best of our knowledge, our adaptive item selection is the first method in CRS to select a variable number of items to robustly utilize the results of the recommendation module. Additionally, our relevance-based groupwise learning is the first method in CRS to apply the concept of grouping in contrastive learning to robustly train the passage retrieval module.

To validate ESPRESSO, we perform extensive experiments on two CRS datasets. Our findings show that (1) each idea of ESPRESSO enhances *passage* retrieval accuracy; (2) applying ESPRESSO to *existing* passage retrieval methods (Karpukhin et al. 2020; Izacard et al. 2021; Wu et al. 2023) *orthogonally* improves their accuracy; (3) ESPRESSO outperforms the best performer among 8 competing methods dramatically by up to 35.91% in passage retrieval accuracy; (4) the generation module can produce higher-quality REC-responses with the aid of ESPRESSO.

Related Work

CRS models. CRS models aim to accurately predict user preferences and provide natural REC-responses. ReDial (Li et al. 2018) predicts user preferences using an autoencoder-based module and generates responses through a hierarchical recurrent encoder-decoder. KBRD (Chen et al. 2019), KGSF (Zhou et al. 2020), and UniCRS (Wang et al. 2022) leverage knowledge graphs (KGs) to align users' tastes with recommendations by illustrating explicit relationships between entities. UniMIND (Deng et al. 2023) employs a unified module based on a pre-trained language model with prompt-based learning techniques. KERS (Zhang et al. 2021) introduces a passage retrieval module trained on automatically created pseudo-relevant passages. However, KERS has notable limitations: (1) it does not use the recommendation module's items to retrieve passages aligning with user preferences; and (2) it does not consider the risks that may exist in pseudo-relevant passages at all.

Passage retrieval methods. Numerous studies focus on generating factually correct answers through passage retrieval. DPR (Karpukhin et al. 2020) retrieves relevant passages using a dual-encoder architecture. RAG (Lewis et al. 2020b) introduces retrieval-augmented generation, where generation is supported by retrieved passages. DSI (Tay et al. 2022) employs a differentiable function to produce

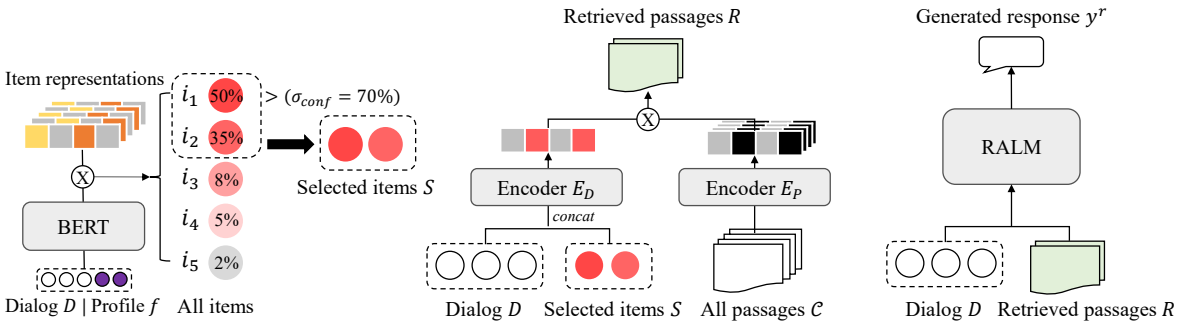

Figure 2: The overview of our CRS framework enhanced by ESPRESSO.

passage identifiers in response to queries. Contriever (Izacard et al. 2021) pre-trains the dual-encoder using contrastive learning on the Inversed Cloze Task. RankGPT (Sun et al. 2023) leverages large language models for reranking passages from BM25. CoT-MAE (Wu et al. 2023) pre-trains the dual-encoder using self-supervised and context-supervised masked auto-encoding.

Proposed Framework

This section elaborates on a new CRS framework enhanced via our passage retrieval approach, named ESPRESSO. As illustrated in Figure 2, the overall process of our CRS framework can be summarized as follows: (1) the recommendation module first predicts the user preferences based on the dialog and additional information (*e.g.*, profile), then selects the items the user would prefer; (2) the passage retrieval module retrieves top-K relevant passages based on the dialog and the selected items from the recommendation module; (3) the generation module produces a REC-response by referring to both the dialog and the top-K passages. Before elaborating on our framework, we first introduce the notations used in this paper.

Notations

Let i denote an item from an item set $\mathcal{I}$. A dialog between a user and the system is denoted as $D = [u_1, u_2, \cdots, u_T]$, where $u_t \in D$ denotes an utterance at the t-th turn, and T denotes the total number of utterances in the dialog D. We also leverage a passage corpus $\mathcal{C} = \{p_1, p_2, \ldots, p_{|\mathcal{C}|}\}$, where each passage $p \in \mathcal{C}$ describes an item $i \in \mathcal{I}$, and $|\mathcal{C}|$ denotes the total number of passages in $\mathcal{C}$.

Recommendation module

The recommendation module performs in two phases: (1) the preference estimation phase and (2) the *adaptive item selection* phase.

Preference estimation. Given a dialog D, the recommendation module first predicts the preference score for each item $i \in \mathcal{I}$. To this end, we can adopt a well-established recommendation approach (Deng et al. 2023; Sun et al. 2019). In this paper, we use a pre-trained language model, *i.e.*, BERT (Devlin et al. 2018; Sun et al. 2019) to encode a dialog D into a d-dimensional representation. Also, we leverage a user's profile f (*e.g.*, age and gender) to predict user preferences more accurately:

$$\hat{y}_i^{rec} = \mathbf{w}_i \cdot (\text{BERT}_{CLS}([D : f]))^{\text{T}}, \quad (1)$$

where $\hat{y}_i^{rec}$ is the preference score for item i, BERT_{CLS} is the output representation of the special token [CLS], and $\mathbf{w}_i \in R^d$ is the learnable representation for item i, which is randomly initialized.

Adaptive item selection. The recommendation module needs to select items that the user prefers and provide them to the passage retrieval module. Here, we aim to include items that the user truly prefers in the selection while excluding less-preferred ones to avoid unnecessary noise. To this end, our idea *adaptively* selects a *variable number* of items based on the confidence score for each item $i \in \mathcal{I}$, obtained from the preference estimation phase. This approach is compared to existing CRS methods (Wang et al. 2022; Deng et al. 2023), which typically select a *fixed number* of items (*e.g.*, the top-1 item). Specifically, if there is sufficient confidence for the top-1 item, it selects only this item. Otherwise, it proceeds to select more items until sufficient confidence is ensured. To do so, we calculate a confidence score $\text{C}(i)$ for each item $i \in \mathcal{I}$ based on the preference score $\hat{y}_i^{rec}$ from Eq.1:

$$\text{C}(i) = \frac{\exp(\hat{y}_i^{rec})}{\sum_{l \in \mathcal{I}} \exp(\hat{y}_l^{rec})}. \quad (2)$$

We assume that the higher the confidence score for item i, the more likely it is that the user truly prefers item i. Then, we enlarge the selected pool by incrementally including the items sorted in descending order of estimated user preferences, starting from the top-1 item, until the cumulative confidence score of the items in the selected pool exceeds a given threshold (see Figure 2-(a)). Specifically, we define the set of finally selected items S by the *adaptive item selection* as the minimal number of items satisfying:

$$\sum_{i \in S} \text{C}(i) > \sigma_{conf}, \quad (3)$$

where σ_{conf} is the pre-defined threshold, ranging from 0% to 100%. We will conduct a *sensitivity analysis* of the threshold σ_{conf} for adaptive item selection in Appendix-RQ7.

Passage retrieval module

We describe how ESPRESSO performs the top-K passage retrieval. Following Direction-1 in the Introduction, we leverage both the dialog D and selected items S from the recommendation module by concatenating them into the *enhanced dialog* $\bar{D} = [D : S]$. Then, we encode the enhanced dialog $\bar{D}$ and each passage $p \in \mathcal{C}$ by employing the dual encoder architecture (Karpukhin et al. 2020). The dual encoder is composed of two Transformer (Vaswani et al. 2017) encoders, denoted as $\mathrm{E}_D(\cdot)$ (resp, $\mathrm{E}_P(\cdot)$), to encode the enhanced dialog $\bar{D}$ (resp. a passage p) into d-dimensional representations:

$$\mathbf{h}^{\bar{D}} = \mathrm{E}_D(\bar{D}), \mathbf{h}^p = \mathrm{E}_P(p), \tag{4}$$

where $\mathbf{h}^{\bar{D}}$ (resp. $\mathbf{h}^p$) is the d-dimensional representation of the enhanced dialog $\bar{D}$ (resp. the passage p). Here, we can utilize a pre-trained encoder from existing passage retrieval methods (*e.g.*, Contriever (Izacard et al. 2021) and CoT-MAE (Wu et al. 2023)) as the initial parameters for E_D and E_P. Next, we calculate the retrieval score $\hat{y}_p^{ret}$ for each passage $p \in \mathcal{C}$ via dot-product (Kim et al. 2022; Lee et al. 2021) between the enhanced dialog representation $\mathbf{h}^{\bar{D}}$ and passage representation $\mathbf{h}^p$ (see Figure 2-(b)):

$$\hat{y}_p^{ret} = \mathbf{h}^{\bar{D}} \cdot (\mathbf{h}^p)^{\mathrm{T}}. \tag{5}$$

Lastly, we retrieve the top-K passages with the highest retrieval score $\hat{y}_p^{ret}$, denoted as $R = [p_1, p_2, \cdots, p_K]$. We assert that, by using the selected items S from adaptive item selection as additional input, the passage retrieval module can effectively retrieve the top-K passages that align with user preferences. We will analyze the effect of selected items S on passage retrieval in Evaluation-RQ1.

Generation module

Given a dialog D and retrieved passages R from ESPRESSO, the generation module aims to produce high-quality REC-responses by referring to these retrieved passages. To this end, we employ the retrieval-augmented language models (RALM) (Lewis et al. 2020b), enhanced by our passage retrieval module (*i.e.*, ESPRESSO). RALM generates a response by using both the dialog D and relevant passages R (see Figure 2-(c)):

$$\Pr(y_i|D, p, y_{1:i-1}) = \sum_{p \in R} \Pr_\eta(p|D, S) \Pr_\theta(y_i|D, p, y_{1:i-1}), \tag{6}$$

where y_i is the i-th token to be generated from RALM, $p \in R$ is a retrieved passage, η is the model parameters of ESPRESSO, and θ is the model parameters of the generative language model in RALM. Here, we can use well-known pre-trained language models (Lewis et al. 2020a; Touvron et al. 2023) as our generative language model. Also, $\Pr_\eta(p|D, S)$ represents the probability of the retrieved passage p calculated from ESPRESSO, based on retrieval score $\hat{y}_p^{ret}$ from Eq.5. $\Pr_\theta(y_i|D, p, y_{1:i-1})$ is the probability of the i-th token y_i calculated from the generative language model, given the dialog D and retrieved passage p concatenated, and having generated up to the (i-1)-th token.

As we have improved the quality of the top-K passages retrieved by ESPRESSO, our RALM is expected to generate more-informative and factually-correct REC-responses. The correlation between the quality of top-K retrieved passages and that of REC-responses will be confirmed in Evaluation-RQ4. Additionally, the superiority of ESPRESSO for response generation is detailed in Evaluation-RQ5 and Appendix-RQ8.

Training strategy

Now, we elaborate on the details of the training strategy for the passage retrieval module. First, we describe how to obtain pseudo-relevant passages. Then, we introduce our novel training strategy, *i.e.*, *relevance-based groupwise learning*.

Labeling module. Our labeling module *automatically annotates the training labels* for the relevant passages, *i.e.*, *pseudo-relevant passages*, to train the passage retrieval module. To this end, we label the passages that closely *resemble the ground-truth response* as pseudo-relevant passages (Zhang et al. 2021). This is because relevant passages, which help generate a response to a given dialog, are likely to have a high degree of keyword overlap (*e.g.*, movie title, actors, and director) with the response. Specifically, to emphasize and complement the keywords in the response r, we integrate additional information that shares the context with response r. We concatenate the response r with the target item i^r and the user's last utterance $u_T \in D$ into the *enhanced response* $\bar{r} = [r : i^r : u_T]$. Then, we use BM25 (Robertson, Zaragoza et al. 2009), which effectively captures lexical similarity and performs well without fine-tuning, to calculate the pseudo-relevance score $\hat{y}_p^{pse}$ between the enhanced response $\bar{r}$ and a passage $p \in \mathcal{C}$:

$$\hat{y}_p^{pse} = \mathrm{BM25}(\bar{r}, p). \tag{7}$$

However, we point out that relying solely on keyword overlap struggles to capture deep semantic relatedness. Therefore, for more accurate passage labeling, we also utilize large language model GPT-4o (Achiam et al. 2023) to rerank what BM25 offers (Sun et al. 2023). Concretely, we first select the top-10 candidate passages through BM25. Then, we leverage GPT-4o to rerank these candidate passages to obtain the final top-M passages as pseudo-relevant passages $Q = [q_1, q_2, \cdots, q_M]$, where $M < 10$ and M is the total number of pseudo-relevant passages for training. We validate the reliability of our labeling approach for creating pseudo-relevant passages in Appendix-RQ6.

Relevance-based groupwise learning. As in existing passage retrieval methods for model training (Karpukhin et al. 2020; Wu et al. 2023), we can use contrastive learning that trains the model to increase the score of *each* pseudo-relevant passage than that of other passages. However, as mentioned in the Introduction, some of them might be *mislabeled* because the labeling module cannot always be perfect. For this reason, unless revised, contrastive learning risks misleading the training process by treating each mislabeled passage as a positive one.

Therefore, we propose a new training method to robustly train the passage retrieval module, even when some pseudo-relevant passages are occasionally mislabeled. Our approach

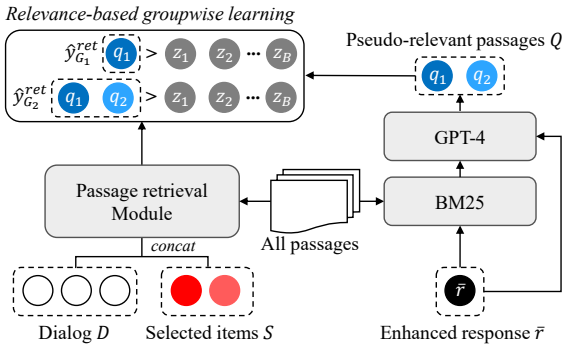

Figure 3: The training process of the passage retrieval module. For example, if the pseudo-relevant passages are $Q = [q_1, q_2]$, then the passage retrieval module is trained (1) to yield a retrieval score $\hat{y}^{ret}_{G_1}$ for G_1 $(= [q_1])$ higher than the score of each negative passage $z \in Z$, and (2) to yield an average retrieval score $\hat{y}^{ret}_{G_2}$ for G_2 $(= [q_1, q_2])$ higher than the score of each negative passage in $z \in Z$.

involves grouping each pseudo-relevant passage with more reliable ones in contrastive learning. The intuition is that, although an individual pseudo-relevant passage may be mislabeled, the likelihood of all elements within the grouped pseudo-relevant passages being mislabeled decreases exponentially. Specifically, we group a passage that is low-ranked, which is relatively more susceptible to mislabeling, with the passages having *higher* ranks as follows:

$$G_j = [q_1, q_2, \cdots, q_j], \tag{8}$$

where $G_j \subset Q$ denotes a subgroup of pseudo-relevant passages that have higher ranks than or equal to the j-th pseudo-relevant passage $q_j \in Q$. Then, we create such subgroups G_j for all pseudo-relevant passages, resulting in a total of M subgroups $\mathcal{G} = \{G_1, G_2, \cdots, G_M\}$.

Lastly, we construct a *relevance-based groupwise learning* loss $\mathcal{L}$ to train the model so that the average retrieval score of each subgroup G_j, $\hat{y}^{ret}_{G_j}$, gets higher than the retrieval score of each negative passage $z \in Z$ (see Figure 3):

$$\hat{y}^{ret}_{G_j} = \frac{\hat{y}^{ret}_{q_1} + \hat{y}^{ret}_{q_2} + \cdots + \hat{y}^{ret}_{q_j}}{|G_j|}, \tag{9}$$

$$\mathcal{L} = -\log \sum_{G_j \in \mathcal{G}} \frac{\exp(\hat{y}^{ret}_{G_j})}{\exp(\hat{y}^{ret}_{G_j}) + \sum_{z \in Z} \exp(\hat{y}^{ret}_{z})}, \tag{10}$$

where $\mathcal{L}$ is the *relevance-based groupwise learning* loss for all the subgroups G_j in $\mathcal{G}$. For the negative passages Z, we utilize both a *hard negative passage* and *in-batch negative passages* used in DPR (Karpukhin et al. 2020). We will validate the effect of relevance-based groupwise learning on passage retrieval accuracy in Evaluation-RQ2.

Evaluation

Datasets. We conducted experiments on two CRS datasets: DuRecDial2.0 (Liu et al. 2021), a public English CRS dataset, and KoRecDial, a private non-English CRS dataset. Both datasets employed human workers to annotate each response with related knowledge, indicating relevant factual information associated with the response (Liu et al. 2021). Following KERS (Zhang et al. 2021), we regard this related knowledge as the ground-truth relevant passages for evaluation. To construct the passage corpus, we gather all relevant passages in each dataset. We provide further details (*e.g.*, statistics and domains) on the datasets and the passage corpus in Appendix-Datasets.

Evaluation. We focus on evaluating the following two tasks: (1) the passage retrieval task and (2) the response generation task. For the passage retrieval task, we adopt Hit@K (K=1, 3, 5) (Karpukhin et al. 2020) to measure whether the top-K passages contain the ground-truth relevant passage. For the response generation task, following (Deng et al. 2023; Zhang et al. 2021; Ma, Takanobu, and Huang 2021), we use BLEU scores (Papineni et al. 2002). We also use GPTEval (Liu et al. 2023) to assess the quality of generated responses in terms of informativeness, relevance, and fluency.

The evaluation of the recommendation module is out of our research scope. Instead, we adopt a well-established recommendation approach (Deng et al. 2023; Sun et al. 2019), focusing on how to effectively utilize its results.

Implementation. Due to space limitations, we provide the details of implementation in Appendix-Implementation.

Results and Analysis

We conducted extensive experiments, aiming at answering the following key research questions (RQs):

- **(RQ1)** Does our *adaptive item selection* improve the accuracy of passage retrieval for CRS?
- **(RQ2)** Does our *relevance-based groupwise learning* improve the accuracy of passage retrieval for CRS?
- **(RQ3)** How much does ESPRESSO outperform state-of-the-art methods for the passage retrieval task?
- **(RQ4)** Does higher-quality passage retrieval improve the quality of response generation for CRS?
- **(RQ5)** How much does the ESPRESSO-enhanced generation module outperform state-of-the-art methods for the response generation task?

(RQ1) Effectiveness of adaptive item selection. We analyze the effect of *adaptive item selection* on passage retrieval by comparing the three variants of selected items used as additional input: (1) top-1 item (*i.e.*, S=top-1), (2) a fixed number of top-N items (*i.e.*, S=top-N), and (3) items selected by our *adaptive item selection* (*i.e.*, S=top-σ_{conf}). For an in-depth analysis, we evaluate passage retrieval accuracy across three cases: (1) when the top-1 item aligns with the ground-truth item (*i.e.*, G=top-1), (2) when it does not (*i.e.*, G≠top-1), and (3) both cases combined (*i.e.*, Overall).[1]

Table 1 shows the passage retrieval accuracy in Hit@3 for the three cases (*i.e.*, G=top-1, G≠top-1, and Overall) when we utilize the three variants of selected items (*i.e.*,

[1]In the DuRecDial2.0 dataset, cases of G=top-1 (resp. G≠top-1) account for 79.4% (resp. 20.6%) of all test dialog samples.

Variants	G=top-1	G≠top-1	Overall
S=top-1	**0.815**	0.003	0.648
S=top-N	0.749	**0.160**	0.628
S=top-σ_{conf}	0.810	0.156	**0.675**

Table 1: Passage retrieval accuracy at Hit@3 in DuRecDial2.0 with three variants of item selection methods: top-1 item (*i.e.*, S=top-1), top-N items (*i.e.*, S=top-N), and *adaptive item selection* (*i.e.*, S=top-σ_{conf}). In the table, G=top-1 (resp. G≠top-1) refers to the case where the top-1 item aligns (resp. does not align) with the ground-truth item.

S=top-1, S=top-N, and S=top-σ_{conf}), where $N = 2$ and $\sigma_{conf} = 70\%$. First, when the top-1 item aligns with user preferences (*i.e.*, G=top-1), the result shows that S=top-1 exhibits the highest accuracy, while S=top-N exhibits the lowest accuracy. This indicates that, in such a case, using more items (*i.e.*, top-N items) may introduce noise into the passage retrieval module, harming the passage retrieval accuracy. Next, when the top-1 item does not align with user preferences (*i.e.*, G≠top-1), S=top-1 fails to retrieve relevant passages, whereas S=top-N exhibits the highest accuracy. This indicates that enlarging the pool of selected items S improves passage retrieval accuracy when the predicted top-1 item does not match user preferences. Finally, our method (*i.e.*, S=top-σ_{conf}) consistently shows comparable accuracy to the best method *in both cases*. Consequently, it outperforms the other variants in overall accuracy, exhibiting a gain of 4.2% (resp. 7.5%) for S=top-1 (resp. S=top-N) in the overall accuracy. This result indicates that we can obtain robust accuracy in various cases via *adaptive item selection*.

Also, we conduct a *sensitivity analysis* of the threshold σ_{conf} for *adaptive item selection* in Appendix-RQ7.

(RQ2) Effectiveness of relevance-based groupwise learning. We validate whether our *relevance-based groupwise learning* enhances passage retrieval accuracy by comparing three different learning strategies: (1) contrastive learning (*i.e.*, CL(M)), (2) groupwise learning (*i.e.*, GL(M)), and (3) relevance-based groupwise learning (*i.e.*, RGL(M)). Here, M denotes the number of utilized pseudo-relevant passages. CL(M) assigns a higher probability to a pseudo-relevant passage over other passages. GL(M) and RGL(M) belong to the category of grouping methods, but they have different grouping strategies. Specifically, GL(M) creates a *single* group that contains *all* pseudo-relevant passages without considering their pseudo-relevance ranks. In contrast, RGL(M) forms multiple subgroups, each corresponding to a pseudo-relevant passage grouped only with others that have higher pseudo-relevance ranks.

Figure 4 shows the accuracy of passage retrieval of ESPRESSO with different learning strategies, namely CL(M), GL(M), and RGL(M), measured at Hit@3 when M is changed from 1 to 3. First, when M is larger than 1, grouping methods (*i.e.*, GL(M) and RGL(M)) always outperform the non-grouping method (*i.e.*, CL(M)). This validates that the grouping strategy ensures stable accuracy, even in the cases where pseudo-relevant passages are occasionally mislabeled. Next, RGL(M), which represents our relevance-based grouping strategy, outperforms all the other variations. This demonstrates the effectiveness of our relevance-based grouping strategy, which groups the passages suspected to be irrelevant only with those deemed more reliable, leading to a further enhancement.

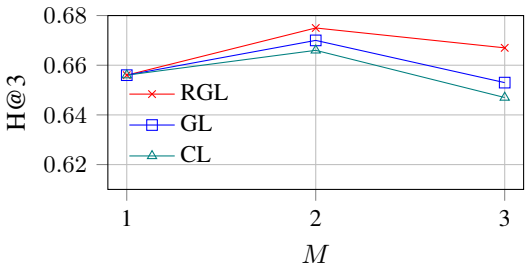

Figure 4: Passage retrieval accuracy at Hit@3 in DuRecDial2.0 with different learning strategies: contrastive learning (*i.e.*, CL), groupwise learning (*i.e.*, GL), and our *relevance-based groupwise learning* (*i.e.*, RGL).

Methods	DuRecDial2.0			KoRecDial		
	H@1	H@3	H@5	H@1	H@3	H@5
BM25	0.260	0.425	0.529	0.059	0.142	0.206
DPR	0.384	0.485	0.538	0.127	0.277	0.372
RAG	0.100	0.120	0.122	0.005	0.007	0.007
KERS	0.291	0.415	0.460	0.091	0.177	0.219
DSI	0.376	0.453	0.484	0.059	0.091	0.108
Contriver	0.393	0.497	0.550	0.123	0.278	0.371
RankGPT	0.339	0.460	0.529	0.101	0.164	0.206
CoT-MAE	0.406	0.504	0.561	0.134	0.283	0.392
OURS+DPR	0.514	0.675	0.725	0.153	0.314	0.414
OURS+Contriever	0.520	0.685	0.735	0.141	0.313	0.423
OURS+CoT-MAE	**0.523**	**0.685**	**0.738**	**0.154**	**0.315**	**0.423**

Table 2: Comparison of ESPRESSO and 8 competitors on the passage retrieval task using Hit ratios.

(RQ3) Comparison with competitors for passage retrieval. We compared ESPRESSO against 8 state-of-the-art passage retrieval methods: BM25 (Robertson, Zaragoza et al. 2009), DPR (Karpukhin et al. 2020), RAG (Lewis et al. 2020b), KERS (Zhang et al. 2021), DSI (Tay et al. 2022), Contriever (Izacard et al. 2021), RankGPT (Sun et al. 2023), and CoT-MAE (Wu et al. 2023). For neural retrieval models that require training labels (*i.e.*, DPR, KERS, DSI, Contriever, and CoT-MAE), we provided the pseudo-relevant passages under the *same conditions* as ESPRESSO. In this context, OURS+DPR, OURS+Contriever, and OURS+CoT-MAE represent neural retrieval models equipped with the two ideas of ESPRESSO, each initialized with the checkpoints of DPR, Contriever, and CoT-MAE, respectively.

Table 2 shows the passage retrieval accuracies of ESPRESSO and 8 competitors across two datasets in Hit

Methods	Passage retrieval			Response generation		
	H@1	H@3	H@5	BLEU2	BLEU3	BLEU4
BART(DPR)	0.384	0.485	0.538	0.167	0.115	0.076
BART(Contriever)	0.393	0.497	0.550	0.171	0.117	0.081
BART(CoT-MAE)	0.406	0.504	0.561	0.170	0.118	0.080
BART(OURS+DPR)	0.514	0.675	0.725	0.195	0.135	0.091
BART(OURS+Contriever)	0.520	0.685	0.735	0.194	0.134	0.092
BART(OURS+CoT-MAE)	**0.523**	**0.685**	**0.738**	**0.197**	**0.137**	**0.093**

Table 3: Impact of various passage retrieval modules on the quality of response generation in DuRecDial2.0.

ratios. First, neural retrieval methods (*i.e.*, DPR, Contriever, and CoT-MAE), which are fine-tuned with our pseudo-relevant passages, outperform non-fine-tuned methods such as RAG, BM25, and RankGPT. Notably, CoT-MAE outperforms RankGPT, which uses the large language model GPT-4o, by up to 19.76% in DuRecDial2.0. This result underscores the importance of *supervision* for the passage retrieval module in CRS, as claimed in Direction-2. Next, applying our two core ideas (*i.e.*, OURS) to the neural retrieval models (*i.e.*, DPR, Contriever, and CoT-MAE) *orthogonally and substantially* improves their performance. In particular, OURS+CoT-MAE outperforms CoT-MAE (*i.e.*, best competitor), exhibiting *large* gains of 28.82%/35.91%/31.55% and 14.93%/11.31%/7.91% in terms of Hit@1/Hit@3/Hit@5 on the DuRecDial2.0 and KoRecDial datasets, respectively.

(RQ4) Impact of passage retrieval on response generation. We compare the response generation quality of BART-large equipped with different passage retrieval models: DPR, Contriever, CoT-MAE, OURS+DPR, OURS+Contriever, and OURS+CoT-MAE. Table 3 shows the results of both passage retrieval and response generation for BART variants with passage retrieval methods, in terms of Hit ratios for the passage retrieval task and BLEU scores for the response generation task. The results show that, as the accuracy of passage retrieval increases, the quality of generated responses increases as well. In particular, BART(OURS+CoT-MAE) enhances the quality of generated responses by exhibiting gains of up to 16.25% compared to BART(CoT-MAE). This verifies the crucial role of a thoughtfully designed retrieval engine in achieving the objectives of CRS.

(RQ5) Comparison with competitors for response generation. We compared the generation module with ESPRESSO against 2 CRS methods (KERS (Zhang et al. 2021) and UniMIND (Deng et al. 2023)), and 6 language generation models (GPT2-base, GPT2-large (Radford et al. 2018), BART-base, BART-large (Lewis et al. 2020a), RAG (Lewis et al. 2020b), and LLaMa2 (Touvron et al. 2023)). For KoRecDial, the accuracy of GPT2-large, BART-large, and LLaMa2 could not be obtained due to the absence of non-English variants of the underlying language models. To assess the impact of ESPRESSO on response quality, we evaluated BART-large (BART-base in KoRecDial) and LLaMa2 integrated with ESPRESSO (*i.e.*, BART(ESPRESSO) and LLaMa2(ESPRESSO)). Here, ESPRESSO refers to the OURS+CoT-MAE model in RQ3 chosen for its superiority.

Methods	DuRecDial2.0			KoRecDial		
	BLEU2	BLEU3	BLEU4	BLEU2	BLEU3	BLEU4
GPT2-base	0.080	0.040	0.020	0.065	0.032	0.018
GPT2-large	0.142	0.093	0.061	-	-	-
BART-base	0.088	0.044	0.015	0.077	0.040	0.022
BART-large	0.132	0.083	0.051	-	-	-
RAG	0.161	0.108	0.072	0.100	0.056	0.034
KERS	0.115	0.072	0.047	0.083	0.044	0.026
UniMIND	0.147	0.083	0.055	0.092	0.050	0.028
LLaMa2	0.144	0.094	0.060	-	-	-
BART(ESPRESSO)	0.197	0.137	0.093	**0.123**	**0.073**	**0.045**
LLaMa2(ESPRESSO)	**0.209**	**0.147**	**0.103**	**-**	**-**	**-**

Table 4: Comparison of ESPRESSO-enhanced generation module and 8 competitors on the response generation task using BLEU scores.

Table 4 displays the performance of 8 competitors in terms of BLEU scores (Papineni et al. 2002). First, the method that leverages retrieved passages for generating responses (*i.e.*, RAG) outperforms the other baselines. Note that it even outperforms GPT2-large and LLaMa2, which contain a larger number of parameters. Next, BART(ESPRESSO) significantly outperforms RAG (*i.e.*, the best competitor) with gains of 22.36%/26.85%/29.17% and 23.00%/30.36%/32.35% in the DuRecDial2.0 and KoRecDial datasets, respectively. Lastly, LLaMa2's response quality is substantially enhanced when referencing retrieved passages through ESPRESSO (*i.e.*, LLaMa2(ESPRESSO)).

In addition, we conducted experiments using GPTEval (Liu et al. 2023), which demonstrated that our approach significantly outperforms competitors in terms of informativeness, fluency, and relevance. For more details, please refer to Appendix-RQ8. Additionally, we conducted a case study on response generation in Appendix-RQ9.

Conclusions

We argue that the passage retrieval module in CRS should leverage (1) selected items from the recommendation module and (2) pseudo-relevant passages as training labels. However, it is challenging to effectively utilize the results by other modules since they occasionally make mistakes in practice. Therefore, we propose ESPRESSO, featuring *adaptive item selection* and *relevance-based groupwise learning*. Extensive experiments validate the effectiveness of ESPRESSO, showing that CRS can deliver higher-quality REC-responses through improved passage retrieval. For future work, we will explore how to enable RALM to robustly reference retrieved passages, even if they contain noise.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, Q.; Lin, J.; Zhang, Y.; Ding, M.; Cen, Y.; Yang, H.; and Tang, J. 2019. Towards Knowledge-Based Recommender Dialog System. 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)*, 1803–1813.

Deng, Y.; Zhang, W.; Xu, W.; Lei, W.; Chua, T.-S.; and Lam, W. 2023. A Unified Multi-task Learning Framework for Multi-goal Conversational Recommender Systems. *ACM Transactions on Information Systems*, 41(3): 1–25.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. *arXiv preprint arXiv:2112.09118*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2004.04906*.

Kim, T.; Kim, Y.; Lee, Y.-C.; Shin, W.-Y.; and Kim, S.-W. 2022. Is It Enough Just Looking at the Title? Leveraging Body Text To Enrich Title Words Towards Accurate News Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4138–4142.

Kim, T.; Yu, J.; Shin, W.-Y.; Lee, H.; Im, J.-h.; and Kim, S.-W. 2023. LATTE: A Framework for Learning Item-Features to Make a Domain-Expert for Effective Conversational Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1144–1153.

Lee, Y.-C.; Kim, T.; Choi, J.; He, X.; and Kim, S.-W. 2021. M-BPR: A novel approach to improving BPR for recommendation with multi-type pair-wise preferences. *Information Sciences*, 547: 255–270.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020a. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020b. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, R.; Ebrahimi Kahou, S.; Schulz, H.; Michalski, V.; Charlin, L.; and Pal, C. 2018. Towards Deep Conversational Recommendations. *Advances in Neural Information Processing Systems*, 31.

Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; and Zhu, C. 2023. Gpteval: Nlg Evaluation Using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634*.

Liu, Z.; Wang, H.; Niu, Z.-Y.; Wu, H.; and Che, W. 2021. DuRecDial 2.0: A Bilingual Parallel Corpus for Conversational Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4335–4347.

Lu, Y.; Bao, J.; Song, Y.; Ma, Z.; Cui, S.; Wu, Y.; and He, X. 2021. RevCore: Review-augmented conversational recommendation. *arXiv preprint arXiv:2106.00957*.

Ma, W.; Takanobu, R.; and Huang, M. 2021. CR-Walker: Tree-Structured Graph Reasoning and Dialog Acts for Conversational Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1839–1851.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Park, S.-J.; Chae, D.-K.; Bae, H.-K.; Park, S.; and Kim, S.-W. 2022. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 784–793.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving Language Understanding by Generative Pre-Training.

Robertson, S.; Zaragoza, H.; et al. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1441–1450.

Sun, W.; Yan, L.; Ma, X.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; and Ren, Z. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Tay, Y.; Tran, V.; Dehghani, M.; Ni, J.; Bahri, D.; Mehta, H.; Qin, Z.; Hui, K.; Zhao, Z.; Gupta, J.; et al. 2022. Transformer Memory as a Differentiable Search Index. *Advances in Neural Information Processing Systems*, 35: 21831–21843.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. LLaMa 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, X.; Zhou, K.; Wen, J.-R.; and Zhao, W. X. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1929–1937.

Wu, X.; Ma, G.; Lin, M.; Lin, Z.; Wang, Z.; and Hu, S. 2023. ConTextual Masked Auto-Encoder for Dense Passage Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4738–4746.

Zhang, J.; Yang, Y.; Chen, C.; He, L.; and Yu, Z. 2021. KERS: A knowledge-Enhanced Framework for Recommendation Dialog Systems with Multiple Subgoals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 1092–1101.

Zhou, K.; Zhao, W. X.; Bian, S.; Zhou, Y.; Wen, J.-R.; and Yu, J. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1006–1014.