A APPENDIX

A.1 DATASETS

For recommendation and product search, we conduct experiments on three domains from the Amazon review dataset (He & McAuley, 2016): Amazon-Beauty, Amazon-Sports, and Amazon-Toys. For recommendation, we keep the users and items with at least 5 interactions in their history in the Amazon review dataset. We treat the last interacted item by each user as the testing sample, the last second interacted item as the validation sample, and the previous items as training samples. For product search, to verify if the learned semantic IDs can be generalized to different downstream tasks, we keep the product corpus in the three domains the same as those in the recommendation experiments. We keep the queries in the original product search dataset (Reddy et al., 2022) which correspond to ground truth products in the product corpus. We use the original train/test split and randomly select 1/8 queries from the training set to be the validation set.

The statistics of the recommendation and product search datasets can be found in Table 6.

Table 6: Dataset Statistics

Dataset	# Items	# Users	# Rec history (train/dev/test)	# Search query (train/dev/test)	# Search labels (train/dev/test)
Amazon-Beauty	12,101	22,363	111,815 / 22,363 / 22,363	1,049 / 150 / 338	1,907 / 268 / 582
Amazon-Sports	18,357	35,598	177,990 / 35,598 / 35,598	1,299 / 186 / 443	2,209 / 311 / 764
Amazon-Toys	11,924	19,412	97,060 / 19,412 / 19,412	1,010 / 145 / 351	1,653 / 250 / 594

For document retrieval, we conduct experiments on Natural Question (NQ) (Kwiatkowski et al., 2019) and MS MACRO (Nguyen et al., 2016). For NQ, we keep the original training and testing labels and put all the documents together to form the text corpus. For MS MACRO, following Pradeep et al. (2023), we construct an MS MACRO-1M by extracting a 1 million document subset from the original collection and keeping the original training and validation labels. For TREC-DL, we merge the TREC-DL 2019 and TREC-DL 2020 datasets and keep the documents appearing in MACRO 1M. MS MACRO dev and TREC-DL Craswell et al. (2020) are used as the evaluation set for MS MACRO.

The statistics of the document retrieval datasets can be found in Table 7.

Table 7: Dataset Statistics

Dataset	# Documents	# Query (train/test)	# Search labels (train/test)
NQ320k	109,739	307,373 / 7,830	307,373 / 7,830
MACRO 1M	1,000,000	502,939 / 6,980	532,751 / 7437
TREC-DL 1M	1,000,000	502,939 / 93	532,751 / 1,069

A.2 SUMMARY OF LMINDEXER'S SELF-SUPERVISED ID LEARNING PROCEDURE

A.3 IMPLEMENTATION DETAILS

In self-supervised semantic indexer training, we use T5-base (Raffel et al., 2020) as our base model. The length of the semantic IDs is set as $T = 3$. The final position is added to distinguish documents sharing the first two position ID prefixes. For $t = 1$ and $t = 2$, we provide 50% hints and 30% hints for reconstruction respectively. We have different codebook embeddings initialized for different positions t and the size of the codebook is set to be in {512, 5,120, 51,200} depending on the size of the document corpus. We optimize the model with AdamW and search the learning rate in {1e-3, 2e-3, 5e-3}. The training epochs are set to be 30, 10, and 5 for Amazon datasets, NQ, and MS MACRO respectively. The hyper-parameter configuration for self-supervised semantic indexer training can be found in Table 8.

In the downstream recommendation task, for generative recommendation methods with semantic IDs (rq-VAE indexer, hierarchical clustering indexer, and LMINDEXER), we concatenate the textual information (title & description) of the user's previously interacted items, serve it as the input text into the generative language model and ask the model to generate the ID for next item. The baselines are using the same T5-base checkpoint. We train all the compared generative recommendation methods

Algorithm 1: Self-supervised ID Learning Procedure of LMINDEXER

Input : The document corpus $\{d\}$.

Output : The semantic IDs $\{c_d\}$ of the documents $\{d\}$. A semantic indexer SemIndexer$(\cdot)$ which contains a semantic encoder SemEnc$_\theta(\cdot)$ and codebooks $\{\boldsymbol{E}^t\}_t$. A reconstruction model Recon$_\phi(\cdot)$.

begin

  // initialize semantic encoder

  SemEnc$_\theta(\cdot) \leftarrow$ T5-base ;

  // reconstructor warm up

  $\min_\phi \mathcal{L}^0_{\text{recon}} = -\sum_d \sum_{w \in d \setminus d^0_h} \log P_{\text{recon}}(w|d^0_{\text{h}})$;

  for $t = 1, ..., T$ **do**

    // semantic encoder & codebook warm up

    $\boldsymbol{h}^t_d \leftarrow \text{SemEnc}_\theta(d, c^{<t}_d)$;

    $\boldsymbol{z}_w \leftarrow \text{Recon}_\phi(q = \{\boldsymbol{c}^{<t}_d, \boldsymbol{h}^t_d\}, k = \boldsymbol{d}^t_{\text{h}}, v = \boldsymbol{d}^t_{\text{h}})$;

    $\min_{\theta,\phi} \mathcal{L}^t = \mathcal{L}^t_{\text{recon}} + \mathcal{L}^t_{\text{contrastive}} + \mathcal{L}^t_{\text{commitment}}$;

    $\boldsymbol{h}^t_d \leftarrow \text{SemEnc}_\theta(d, c^{<t}_d)$;

    $\boldsymbol{E}^t \leftarrow \text{KMeans}(\{\boldsymbol{h}^t_d\})$;

    // whole framework training

    $\boldsymbol{z}_w \leftarrow \text{Recon}_\phi(q = \{\boldsymbol{c}^{<t}_d, \hat{\boldsymbol{c}}^t_d\}, k = \boldsymbol{d}^t_{\text{h}}, v = \boldsymbol{d}^t_{\text{h}})$;

    $\min_{\theta,\phi,\boldsymbol{E}^t} \mathcal{L}^t = \mathcal{L}^t_{\text{recon}} + \mathcal{L}^t_{\text{contrastive}} + \mathcal{L}^t_{\text{commitment}}$;

    $c^t_d \leftarrow \text{argmax}_j P_s(c^t_d = j|c^{<t}_d, d)$;

  end

  return $\{c_d\}$, *SemIndexer*$(\cdot)$;

end

for 10,000 steps with the learning rate searched in {1e-2, 1e-3, 1e-4}. The batch size is set to be 32, the maximum input text length is set to be 1024 and all experiments are run on an 8 A100 40G machine. The number of beams for beam search is set to 20. The hyper-parameter configuration for generative recommendation training can be found in Table 9.

In the downstream product search task, for generative retrieval methods with semantic IDs (rq-VAE indexer, hierarchical clustering indexer, and LMINDEXER), we serve the query as the input text into the generative language model and ask the model to generate the ID for the relevant items. All baselines initially load the same T5-base checkpoint. We train all the compared generative retrieval methods for 10,000 steps with the learning rate searched in {1e-2, 1e-3, 1e-4}. The batch size is set to 32, the maximum input text length is set to be 1024 and all experiments are run on an 8 A100 40G machine. The number of beams for beam search is set to 20. The hyper-parameter configuration for generative product search training can be found in Table 10.

In the downstream document retrieval task, for generative retrieval methods with semantic IDs (rq-VAE indexer, hierarchical clustering indexer, and LMINDEXER), we serve the query as the input text into the semantic indexer and ask the model to generate the ID for the relevant documents. Following (Wang et al., 2022), we use docT5query (Nogueira et al., 2019) to generate pseudo queries for each document in NQ and MS MACRO for training augmentation. The number of pseudo queries for each document is set to be 15 and 20 respectively. We train all the compared generative retrieval methods for 250,000 and 500,000 steps in NQ and MS MACRO respectively, with the learning rate searched in {5e-4, 1e-3, 5e-3}. The batch size is set to 2048, the maximum input text length is set to 32 and all experiments are run on an 8 A100 40G machine. The number of beams for beam search is set to 20. All baselines initially load the same T5-base checkpoint. The hyper-parameter configuration for generative document retrieval training can be found in Table 11.

A.4 SEMANTIC ID LENGTH STUDY

In this section, we analyze how the length of the semantic IDs affects the downstream recommendation performance. We conduct experiments with the length of item semantic IDs to be 1, 2, and 3. The results on the Amazon-Beauty, Amazon-Sports, and Amazon-Toys datasets are shown in Figure 6. From the result, we can find that the model performance increases as the semantic ID length increases. The result is intuitive, since the longer the semantic ID is, the more semantic information it can contain.

Table 8: Hyper-parameter configuration for self-supervised semantic ID learning.

Parameter	Amazon-Beauty	Amazon-Sports	Amazon-Toys	NQ	MACRO-1M
Optimizer	Adam	Adam	Adam	Adam	Adam
Adam ϵ	1e-6	1e-6	1e-6	1e-6	1e-6
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Batch size	128	128	128	128	128
Max epochs	30	30	30	10	5
Max sequence length	512	512	512	512	128
ID length	3	3	3	3	3
Codebook size	512	512	512	5120	51200
Hint ratio	50%, 30%	50%, 30%	50%, 30%	50%, 30%	50%, 30%
Learning rate	searched in {1e-3, 2e-3, 5e-3}				
Backbone LM	T5-base				

Table 9: Hyper-parameter configuration for generative recommendation.

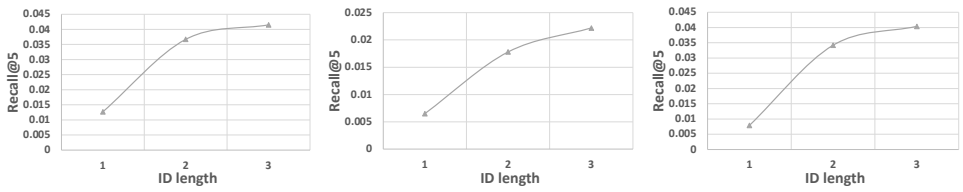
Parameter	Amazon-Beauty	Amazon-Sports	Amazon-Toys
Optimizer	Adam	Adam	Adam
Adam ϵ	1e-6	1e-6	1e-6
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Batch size	32	32	32
Max steps	10,000	10,000	10,000
Max sequence length	1024	1024	1024
Bean size	20	20	20
Learning rate	searched in {1e-2, 1e-3, 1e-4}		
Backbone LM	T5-base		

Table 10: Hyper-parameter configuration for generative product search.

Parameter	Amazon-Beauty	Amazon-Sports	Amazon-Toys
Optimizer	Adam	Adam	Adam
Adam ϵ	1e-6	1e-6	1e-6
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Batch size	32	32	32
Max steps	10,000	10,000	10,000
Max sequence length	1024	1024	1024
Bean size	20	20	20
Learning rate	searched in {1e-2, 1e-3, 1e-4}		
Backbone LM	T5-base		

Table 11: Hyper-parameter configuration for generative retrieval.

Parameter	NQ	MACRO-1M
Optimizer	Adam	Adam
Adam ϵ	1e-6	1e-6
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)
Batch size	2,048	2,048
Max steps	250,000	500,000
Max sequence length	32	32
Bean size	20	20
Learning rate	searched in {5e-4, 1e-3, 5e-3}	
Backbone LM	T5-base	



(a) beauty (b) sports (c) toys

Figure 6: Semantic ID length study on recommendation.