# Exploiting Intent Evolution in E-commercial Query Recommendation

Yu Wang
University of Illinois Chicago
ywang617@uic.edu

Zhengyang Wang
Amazon
zhengywa@amazon.com

Hengrui Zhang
University of Illinois Chicago
hzhan55@uic.edu

Qingyu Yin
Amazon
qingyy@amazon.com

Xianfeng Tang
Amazon
xianft@amazon.com

Yinghan Wang
Amazon
yinghanw@amazon.com

Danqing Zhang
Amazon
danqinz@amazon.com

Limeng Cui
Amazon
culimeng@amazon.com

Monica Cheng
Amazon
chengxc@amazon.com

Bing Yin
Amazon
alexbyin@amazon.com

Suhang Wang
Amazon
ysuhwang@amazon.com

Philip S. Yu
University of Illinois Chicago
psyu@uic.edu

## ABSTRACT

Aiming at a better understanding of the search goals in the user search sessions, recent query recommender systems explicitly model the reformulations of queries, which hopes to estimate the intents behind these reformulations and thus benefit the next-query recommendation. However, in real-world e-commercial search scenarios, user intents are much more complicated and may evolve dynamically. Existing methods merely consider trivial reformulation intents from semantic aspects and fail to model dynamic reformulation intent flows in search sessions, leading to sub-optimal capacities to recommend desired queries. To deal with these limitations, we first explicitly define six types of query reformulation intents according to the desired products of two consecutive queries. We then apply two self-attentive encoders on top of two pre-trained large language models to learn the transition dynamics from semantic query and intent reformulation sequences, respectively. We develop an intent-aware query decoder to utilize the predicted intents for suggesting the next queries. We instantiate such a framework with an Intent-aware Variational AutoEncoder (IVAE) under deployment at Amazon. We conduct comprehensive experiments on two real-world e-commercial datasets from Amazon and one public dataset from BestBuy. Specifically, IVAE improves the Recall@15 by 25.44% and 60.47% on two Amazon datasets and 13.91% on `BestBuy`, respectively.

## CCS CONCEPTS

• **Information systems → Recommender systems**.

## KEYWORDS

Query Recommendation, Intent-aware Model, Session-based Recommendation, Pre-trained Large Language Model

**ACM Reference Format:**
Yu Wang, Zhengyang Wang, Hengrui Zhang, Qingyu Yin, Xianfeng Tang, Yinghan Wang, Danqing Zhang, Limeng Cui, Monica Cheng, Bing Yin, Suhang Wang, and Philip S. Yu. 2023. Exploiting Intent Evolution in E-commercial Query Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599821

## 1 INTRODUCTION

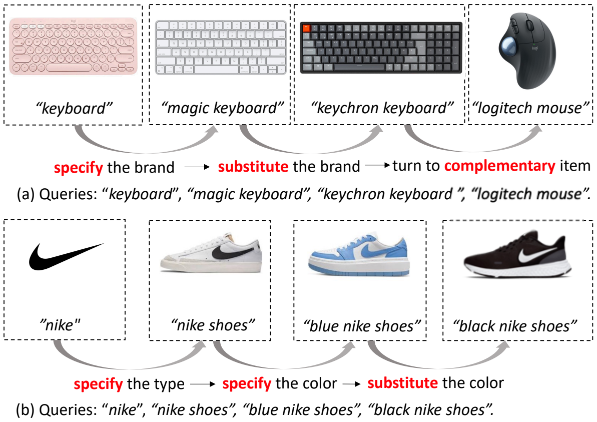
In modern online search services, users may make multiple search attempts before finding the desired products. This reformulation process could be even longer when users perform complex search tasks. To alleviate such search difficulty, query recommender systems [3, 14, 15, 29, 34] (e.g., related search widgets on Amazon) are becoming integral components by suggesting candidate queries that reflect the user search intents to help refine their queries. To better understand the user intents, session-based query recommendations that learn sequential patterns in user historical query logs have been investigated to make more precise recommendations [33, 35, 41].

However, the development of session-based query recommender systems is restricted due to the noise and the ambiguity in the query logs introduced by the imprecise articulations of hidden search intents. To alleviate such problems, recent studies resort to query reformulations to better understand the inherent logic behind the user searching processes [6, 17] since the series of reformulations are the visible manifestations of their search intents. For example, Jiang and Wang [17], Guo et al. [13], and Mitra [30] explicitly learn the reformulation vector representations by assuming that the reformulation is the semantic differences between queries by either adding/deleting terms.

Despite the success of the above methods for web search, query reformulation is under-explored in the e-commercial product search

*KDD '23, August 6–10, 2023, Long Beach, CA, USA*

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599821


“keyboard”
“magic keyboard”
“keychron keyboard”
“logitech mouse”
specify the brand → substitute the brand → turn to complementary item
”nike”
“nike shoes”
“blue nike shoes”
“black nike shoes”
specify the type → specify the color → substitute the color

(a) Queries: *“keyboard”, “magic keyboard”, “keychron keyboard ”, “logitech mouse”.*

(b) Queries: *“nike”, “nike shoes”, “blue nike shoes”, “black nike shoes”.*

**Figure 1: The query reformulation intents are complex in e-commerce: (a) two consecutive queries might be semantically unrelated but target closely related products (e.g., complementary items); (b) there are sequential patterns when users reformulate the queries in a session.**

domain, where the queries are more complicated and dynamically evolving. Concretely, a user may enter product-related terms such as brand, resulting in consecutive queries that are semantically dissimilar being relevant in terms of their desired products. Specifically, an example, as shown in Fig. 1(a) would be { *"keyboard", "magic keyboard", "keychron keyboard", "logit mouse"* }, which first **specifies** or **substitutes** the search goal, then switches to a **complementary** item. The two consecutive queries { *"keychron keyboard", "logit mouse"* } are semantically dissimilar, but we can easily observe a complementary relationship between the desired items of the two queries. Existing methods cannot reveal such hidden relationships as they merely compare the pairwise query semantics.

In addition to the above-described complicated pairwise reformulation intents, it is also observed that there exists dynamic evolution of intents when a user reformulates the search queries [6]. For example, by merely investigating the reformulation sequence in Fig. 1(b): { *"nike", "nike shoes", "blue nike shoes", "black nike shoes"* }, we can observe that the user starts by **specifying** the queries for the first two reformulation phases, then reformulates the query to find **substituted** items. As shown in Figure 2, we also observe similar patterns while conducting empirical studies on the Amazon dataset. Take the third row for instance, 0.69% of the total Substitution intents show on the first time step, while 28% of them appear on the last time step. The users may equivalently reformulate their queries at an early stage and keep specifying them. Then, they shift to other related items by substituting or generalizing their queries. We will describe such statistics in detail in Sec. 4.1.2. Understanding the sequential dynamics behind the query reformulations can provide a strong signal for predicting the **intent** of the next search and thus can narrow down the recommendation scope for the next query. However, modeling such sequential patterns of reformulation intents from the relationships between input queries and the desired products is still under-explored.

Consequently, we are interested in developing a model to bridge the above gaps by explicitly modeling the dynamic intent flow from both semantic and product-related perspectives. To achieve this goal, we should deal with two major challenges. First, the intents

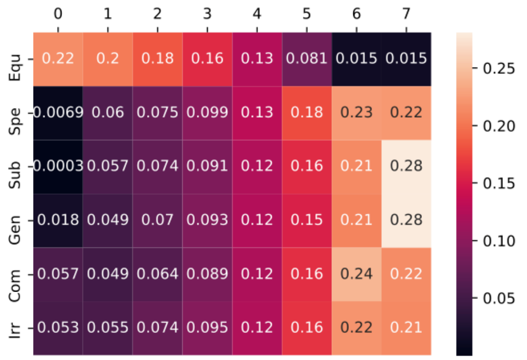


**Figure 2: Intent estimation statistics on Session-AU. The x-axis is the position in a query sequence, and the y-axis is the intent type: Equivalence, Specification, Substitution, Generalization, Complement, and Irrelevant. We obtain the intent prediction from the static intent estimator and analyze the distribution of each intent type w.r.t the corresponding position.**

for reformulating the queries in e-commercial product search are under-defined. It is challenging to get the ground-truth label of a reformulation intent merely from the semantic relations between two queries. Second, to deal with unseen queries, conventional works learn vocabulary representations from scratch [13, 17, 35]. Thus the embeddings of queries that are merely learned from such sparse and noisy domain-specific datasets encode less semantic meanings, exacerbating the difficulty of query recommendation.

To address the above challenges, we propose a novel framework named **I**ntent-**A**ware **V**ariational **A**uto**E**ncoders (IVAE) for query recommendation tasks. IVAE consists of the following key designs: 1) We first formally define six types of reformulation intents according to the relationships between the desired items. We then train a language-model-enhanced reformulation intent estimator, which takes a pair of consecutive queries as input and predicts the corresponding intents based on extremely limited annotations. For all unlabeled pairs of consecutive queries, the predictions from the intent estimator will be used as their pseudo-labels for the subsequent learning procedure. 2) After extracting a reformulation sequence by estimating every consecutive query pair using the trained intent estimator from the original query sequence, we explicitly model the evolving dynamics of user intents using a sequential model. Then our method can predict the next reformulation intents considering the historical reformulation behaviors. We further utilize the predicted reformulation intents to facilitate the prediction of the next query. 3) We employ a frozen Pre-trained Large Language Model (PLLM) to generate the input query features instead of one-hot vocabulary indices. This not only helps to alleviate the data sparsity issue but also enables our method to handle cold-start sessions and unseen queries without the additional computational burden of fine-tuning the PLLM. We also employ an additional regularization term, named *DeepWhitening* on query embeddings to get rid of the PLLMs’ anisotropic problem, which results in poorly semantically encoded query representations [12, 22, 36]. We incorporate these key designs in a Variational AutoEncoder for its merits of controllable generation and robustness to uncertainty/noise from input queries and the intent estimator. Our designs are also compatible with other AutoEncoders, e.g. DAE, MAE. The main contributions of this paper are:

1) To our best knowledge, we are the first to explicitly model the dynamic intent flows in terms of both semantic and product-related perspectives, which is an important real-world e-commercial problem under-explored. We also inject such sequential intent estimations into the generation of the next query to improve the query recommendation performance in e-commerce.
2) We propose a new framework named IVAE to improve e-commercial query recommendations by modeling intent dynamics and addressing the issues of data noise and unseen queries simultaneously.
3) We collect two real-world datasets from Amazon and one public dataset from BestBuy, and conduct extensive experiments on real-world e-commercial datasets. IVAE consistently outperforms the baseline methods on all datasets. Especially for Recall@15, IVAE improved the performance by 25.44%, 60.47% on two Amazon datasets, and 13.91% on `BestBuy`, respectively.

# 2 RELATED WORKS

## 2.1 Session-based Query Recommendation

Session-based query recommendation, which aims to predict the next possible query according to the historical query records within the current session, has been investigated for decades in the web search areas. A line of work applies sequential models over the query logs that implicitly model the semantic transitions between query embeddings as reformulation signals. As a pioneering work, HRED [35] adopts a hierarchical encoder-decoder framework, where both the encoder and decoder are implemented with RNNs. Chen et al. [5], Dehghani et al. [9] and Mustar et al. [31] use transformer to learn the sequential patterns from the historical queries. Ahmad et al. [1] adopts a multi-task training paradigm that combines the query recommendation task with a document ranking task. Furthermore, the query recommendation task is handled with the hierarchical RNN resembling architecture in HRED [35]. Ahmad et al. [2] further enriches Ahmad et al. [1] with click information.

Another line of work explicitly learns a reformulation representation from query embeddings by assuming that the reformulations either add or delete terms from source queries. Guo et al. [13] first estimates the latent intent of each query, then learns unique representations for each query of different intent types. The final query representation is the weighted sum of representations of different types, where the weights are simply the probability estimation of each intent type. [30] learns the distributed representation of queries and uses them to implicitly represent query reformulations that can map similar query changes closer in the latent space. Jiang and Wang [17] assumes that the reformulations between consecutive queries are either adding or deleting some terms from the original query. It then feeds the difference between two consecutive queries as additional features into an RNN model for predicting the next query. However, these methods merely model intents from semantic aspects of two consecutive queries and fail to consider the complicated intents in terms of desired objective and dynamic intent evolution.

Session information also has been intensively investigated for product recommendation [18, 23, 24, 28, 37, 39, 40, 43], which apply the sequential models, e.g., RNN, LSTM or Self-attention modules over the sequence of item embeddings to learn the transition patterns from such sequences and predict next possible items.

## 2.2 Pre-trained Large Language Models in Web Search

Pre-trained Large Language Models (PLLMs) [4, 8, 10, 21, 27, 32] have become integral components in natural language processing. Recently, many works have introduced PLLMs to web search tasks [7, 16, 26, 44] and have achieved significant improvement. However, the sentence representation directly from PLLMs poorly encodes the semantic meanings [22], as they are pre-trained by optimizing word token prediction task given context. A line of works points out the problem of anisotropy [12, 19, 22, 36, 42] that the cosine similarity between arbitrary sentence representations is averagely greater than 0.9. This phenomenon hinders the application of PLLMs for sentence-level subsequent tasks based on semantic similarities. Current methods tend to design fine-tuning methods that apply contrastive learning paradigm [12, 19, 42] or post-processing approaches [22, 36] to map the sentence embeddings into the isotropic space. However, the above approaches are insufficient for session-based query recommendations as they only consider the pair-wise sentence similarity independently. Furthermore, fine-tuning and post-processing are not suitable for end-to-end optimization in inductive settings. Few works investigate transferring knowledge learned from PLLM to session-based query recommendation tasks, as sequential modeling is much more challenging than simple retrieval tasks. Mustar et al. [31] investigates the potential of fine-tuning the PLLM by maximizing the semantic similarity between the target query and the concatenation of all previous queries. They also designed a hierarchical framework using the PLLM as the query encoder.

# 3 METHODOLOGY

In this section, we introduce IVAE in detail, with a high-level illustration of IVAE presented in Fig. 3. IVAE consists of four key components: 1) We first train the **Static Intent Estimator** with extremely limited annotations from a product perspective. 2) We employ an **Dynamic Intent Encoder** with a self-attention layer to model the evolving dynamics of reformulation intents. 3) **Query Encoder**: to overcome the data sparsity issue and handle unseen queries, we employ a frozen PLLM followed by a self-attention layer to learn semantic dynamics of input queries. 4) Finally, we design the **Intent-Aware Query Decoder** with a multi-head self-attention layer scaled by intent estimations from the dynamic intent encoder. Next, we introduce the details of each component.

## 3.1 Problem Formulation

We target a real-world e-commercial query recommendation task. The input is a set of query sessions: $\mathcal{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_{|\mathcal{S}|}\}$, and a query set $Q = \{\mathbf{q}_1, \cdots, \mathbf{q}_{|Q|}\}$, where $\mathbf{s}_i$ represents the $i$-th session. Each session corresponds to a sequence of queries $\mathbf{s}_i = [\mathbf{q}_{i,1}, \cdots, \mathbf{q}_{i,T}]$, and each query consists of a sequence of words $\mathbf{q}_{i,j} = [\mathrm{w}^1_{i,j}, \cdots, \mathrm{w}^M_{i,j}]$, where $T$ and $M$ are the maximum session/query length respectively. We pad each session and query so that all sessions and queries are aligned. Then, for each session $\mathbf{s}_i$, we aim to predict the next most possible query $\mathbf{q}_{i,T+1}$. In the remaining parts, we use **bold roman symbols** to represent the raw input and the ***bold italic symbols*** to represent the corresponding vectorized representations. For instance, $\mathbf{q}_i$ denotes the raw input consisting of word tokens of the $i$-th query, whereas $\boldsymbol{q}_i$ denotes its vectorizations.

## 3.2 Static Estimations of Reformulation Intents

*3.2.1 Explicit Definitions of Reformulation Intent Types.* Existing methods model the reformulation intents directly from the differences between semantic query representations, as they assume that the query reformulations in web search either add or delete terms from the original query. However, in e-commercial product search scenarios, two consecutive queries may be semantically dissimilar but relevant in terms of the usage of their desired products, i.e., the relationships between desired products are substitution/complement, etc.

To this end, we explicitly define the following six types of reformulation intents according to underlying relations between the two desired products:

- **Equivalence**: the expected products are equivalent.
- **Specification**: the user adds attributes expecting higher retrieval precision.
- **Substitution**: the user replaces attributes expecting substitute products.
- **Generalization**: the user removes attributes expecting higher retrieval recall.
- **Complement**: the user expects complementary products.
- **Irrelevant**: the expected items are of different product types, and there are no complementary relationships.

To explicitly model the product-related reformulation intents, we need to address the following challenges: 1) It is hard to obtain ground-truth labels, as we have tens of millions of query records but extremely limited intent annotations. 2) The reformulation intent might be complicated, e.g., the transition from *"red shoes"* to *"white Adidas shoes"* contains both specification and substitution reformulation intents. To address these challenges, we use the limited intent annotations to fine-tune a PLLM that takes a pair of queries as input and predicts the corresponding reformulation intent type, and use the predicted soft logits as soft-pseudo-labels of unseen reformulations during the subsequent training procedure.

*3.2.2 Static Intent Estimator.* With the limited annotated intent labels, we wish to learn a predictive model which can estimate the intent given a pair of two consecutive queries. Specifically, we fine-tune a separate PLLM which takes a pair of queries as input and generates a vector denoting the probability that the transition between queries belongs to each reformulation intent:

$$\boldsymbol{i}_t = \mathrm{Softmax}[\mathrm{PLLM}_\phi(\mathbf{q}_{t-1}, \mathbf{q}_t)] \in \mathbb{R}^{1\times C}, \tag{1}$$

where $C$ is the number of reformulation intents ($C = 6$ as defined in Sec. 3.2). The parameters of the PLLM $\phi$ are optimized using cross-entropy loss between the reformulation intents predicted by the PLLM and the limited ground-truth labels. It is also worth noting that the static intent estimator is trained ahead of other model components.

## 3.3 Dynamic Intent Encoder

*3.3.1 Sequential Modeling of Intents.* After the intent estimator is well-trained, we freeze its parameters and use it to estimate the intents for every consecutive query pair. Thus by obtaining a sequence of estimated intents in a session $\boldsymbol{I} = [\boldsymbol{i}_1, \cdots, \boldsymbol{i}_T] \in \mathbb{R}^{T\times C}$, we hope to predict the next intent $\boldsymbol{i}_{t+1}$ given the previous intents $\boldsymbol{i}_{\leq t}$, i.e., learning an autoregressive sequential model. To reach this target, we train a masked self-attention layer (i.e., we mask the tokens after $t$ to avoid information leakage as we would like to predict the intent at $t+1$), which takes the features of previous intents as input and outputs the predicted intent of the next step:

$$\begin{aligned} \boldsymbol{\alpha}_t &= \mathrm{Softmax}\left(\left[\frac{(\boldsymbol{i}_j \boldsymbol{W}^q)(\boldsymbol{i}_t \boldsymbol{W}^k)^\top}{\sqrt{C}}\right]_{j=1}^{t}\right) \in \mathbb{R}^{1\times t}, \\ \boldsymbol{i}'_t &= \mathrm{MSA}(\boldsymbol{i}_{\leq t}) = \sum_{j=1}^{t} \alpha_{t,j} \cdot \boldsymbol{i}_j \boldsymbol{W}^v, \end{aligned} \tag{2}$$

where $\boldsymbol{W}^q, \boldsymbol{W}^k$, and $\boldsymbol{W}^v$ are the linear transformation matrices in self-attention of dimension $\mathbb{R}^{C\times C}$. MSA$(\cdot)$ is the masked-self-attention layer. Note that we also equip positional embeddings to the input intent sequence so that the model is aware of the relative positions of different intents. $\boldsymbol{i}'_t$ is the estimation of the next step intent $\boldsymbol{i}_{t+1}$, e.g., $\boldsymbol{i}'_1$ is the prediction of $\boldsymbol{i}_2$, and $\boldsymbol{i}'_T$ is the prediction of $\boldsymbol{i}_{T+1}$. For convenience, we denote the parameters of the MSA$(\cdot)$ in Intent-Encoder by $\theta_1$.

## 3.4 Query-Encoder

*3.4.1 Embedding Layer with PLLM.* For a query $\mathbf{q} = [\mathrm{w}_1, \cdots, \mathrm{w}_M]$ where each $\mathrm{w}_i$ represents a word token, we feed it into another PLLM (e.g., BERT. Other encoders also applicable.) to obtain the corresponding input query features. During the whole training procedure, we freeze the PLLM's parameters to avoid high computational costs and catastrophic forgetting. Unlike traditional query recommendation models that learn the word token representation from scratch, the utilization of the large language model benefits our method from the rich semantic information of queries. Besides, such a practice can help address the problem of cold-start queries. Specifically, when a brand new query comes into the system, our method is able to generate a meaningful query embedding instead of initializing a random embedding. We denote the pre-trained large language model by PLLM$(\cdot)$, and then apply a shallow MLP model on top of PLLM's output in order to project the embeddings into a different latent space. Formally, this can be written as

$$\boldsymbol{q}_t = \mathrm{MLP}_{\theta_2}(\mathrm{PLLM}(\mathbf{q}_t)), \tag{3}$$

where $\theta_2$ denotes the parameters of the MLP model. Besides the capacity to exploit the rich semantic information in textual data, an additional sequential model is required to learn the transition patterns in query sequences and make predictions for the next possible queries. To this end, we employ a VAE framework with both the encoder and the decoder implemented with masked self-attention (MSA) layers. We introduce the encoder in the next part and the decoder in Sec. 3.5.

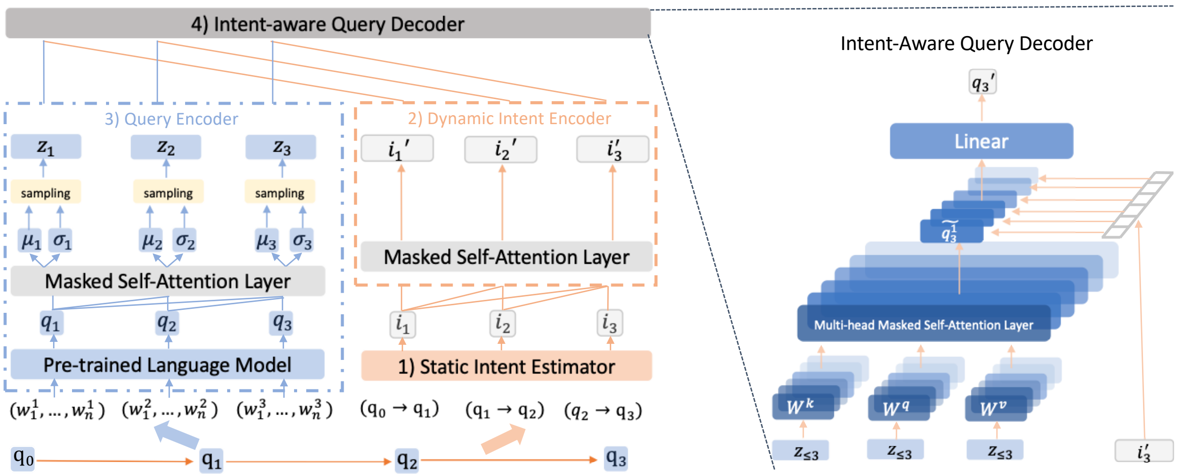

4) Intent-aware Query Decoder
3) Query Encoder
2) Dynamic Intent Encoder
$z_1$
$z_2$
$z_3$
sampling
sampling
sampling
$\mu_1$ $\sigma_1$
$\mu_2$ $\sigma_2$
$\mu_3$ $\sigma_3$
Masked Self-Attention Layer
$q_1$
$q_2$
$q_3$
Pre-trained Language Model
$i_1'$
$i_2'$
$i_3'$
Masked Self-Attention Layer
$i_1$
$i_2$
$i_3$
1) Static Intent Estimator
$(w_1^1, \dots, w_n^1)$ $(w_1^2, \dots, w_n^2)$ $(w_1^3, \dots, w_n^3)$
$(q_0 \to q_1)$ $(q_1 \to q_2)$ $(q_2 \to q_3)$
$q_0$
$q_1$
$q_2$
$q_3$
Intent-Aware Query Decoder
$q_3'$
Linear
$\tilde{q}_3^1$
Multi-head Masked Self-Attention Layer
$W^k$
$W^q$
$W^v$
$z_{\leq 3}$
$z_{\leq 3}$
$z_{\leq 3}$
$i_3'$

**Figure 3: Illustrates the framework of IVAE. For an input session, we extract both the query sequence and the query reformulation sequence, which are fed into PLLM and intent estimator to obtain query features and reformulation intents correspondingly. $q_0$ is the artifact query, which is only used to infer the first intent type. We apply the masked self-attention layer on top of the query feature sequence and reformulation estimation sequence to learn the dynamic sequential patterns. On the right side of the figure is the detailed illustration of the intent-aware decoder, where the dynamic intent estimation based on previous intent behaviors is used as the coefficients of weighted concatenation among different heads of multi-head self-attentions.**

*3.4.2 Masked Self-Attention Layer.* The encoder of the VAE aims to learn the posterior distribution $p_{\theta_3}(z_t|\boldsymbol{q}_{\leq t}) \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$, where $\boldsymbol{z}_t$ is the hidden state of the query at time step $t$, $\boldsymbol{q}_{\leq t}$ are the queries no later than $t$, and $\Sigma_t = \text{diag}(\boldsymbol{\sigma}_t^2)$ is the diagonal covariance matrix. Formally, the mean $\boldsymbol{\mu}_t$ and standard deviation $\boldsymbol{\sigma}_t$ are estimated as

$$\boldsymbol{\mu}_t = \text{MSA}_\mu(\boldsymbol{q}_{\leq t}), \text{ and } \boldsymbol{\sigma}_t = \text{MSA}_\sigma(\boldsymbol{q}_{\leq t}), \tag{4}$$

where $\text{MSA}_\mu(\boldsymbol{q}_{\leq t})$ and $\text{MSA}_\sigma(\boldsymbol{q}_{\leq t})$ are two masked self-attention layers described in Eq. 2 to estimate the corresponding mean $\boldsymbol{\mu}_t$ and covariance $\Sigma_t$ of the hidden state $z_t$, given all the queries before $t$.

With the learned distribution, we are able to sample the latent vector for each state with the reparameterization trick:

$$z_t = \boldsymbol{\mu}_t + \boldsymbol{\varepsilon} \odot \boldsymbol{\sigma}_t, \tag{5}$$

where $\boldsymbol{\varepsilon}$ is noise sampled from standard Normal distribution.

## 3.5 Intent-aware Query-Decoder

Previous methods either entangle the intent implicitly during the query generation or train a query intent classifier first, and then use it as the coefficient of query representation learning. However, the generation of the query recommendation is a complex process, thus implicit entangling or being used as a coefficient is insufficient. To mitigate the above limitations, we devise an intent-aware decoder that takes into account both the predicted next queries (with intent estimation semantically) and the predicted next intent from the product-related perspective when generating the next query.

*3.5.1 Multi-head masked self-attention.* Let $z_{1\cdots T}$ be the sampled latent vectors of queries (with reparameterization trick), i.e., $z_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$, we first use multi-head masked self-attention mechanism to generate a series of output vectors for each time step:

$$\tilde{\boldsymbol{q}}_t^j = \text{MSA}^j(z_{\leq t}), j = 1, \cdots, C. \tag{6}$$

Note that we set the number of attention heads as the number of reformulation intents $C$, so that each head can encode the specific information for each type of intent.

*3.5.2 Intent-scaled weighted concatenation.* To inject the intent knowledge into the decoder model, we define the final estimated query representation as a weighted concatenation of the multi-head outputs up to a linear transformation:

$$\boldsymbol{q}_t' = \text{Concat}\left(\left[i_{t,(j)}' \cdot \tilde{\boldsymbol{q}}_t^j\right]_{j=1}^C\right) \boldsymbol{W}^O, \tag{7}$$

where $\boldsymbol{i}_t'$ is the predicted (normalized) intent vector at time $t$, and $i_{t,(j)}'$ is its $j$-th entry. Then with the output representation at time step $t$, i.e., $\boldsymbol{q}_t'$, we are able to compute a rating score for a target query $k$ as $r_{t,k} = {\boldsymbol{q}_t'}^\top \cdot \boldsymbol{q}_k$. We denote the parameters in Intent-aware Query-Decoder by $\theta_4$.

## 3.6 Optimization

The optimization of IVAE consists of two steps: 1) we first optimize the intent estimator, i.e., fine-tuning the large language model $\text{PLLM}_\phi$ in Eq. 1 by minimizing the cross-entropy loss between the predicted intents and limited ground-truth labels; 2) We freeze the intent estimator's parameters after the first step ends. The remaining modules are jointly optimized with the next-query (Sec. 3.6.1) and next-intent (Sec. 3.6.2) prediction tasks, together with a whitening regularization (Sec. 3.6.3) over the query representations.

*3.6.1 Next query prediction.* For the reconstruction term in VAE's optimization, we target a next-query prediction task. Given a session $\mathbf{s} = [\mathbf{q}_1, \cdots, \mathbf{q}_T]$, the reconstruction loss is defined as:

$$\mathcal{L}_{\text{rec}} = -\sum_{t=1}^{T-1} \left(\log \sigma(r_{t,t+1}) + \log(1 - \sigma(r_{t,k}))\right), \tag{8}$$

where $r_{t,t+1} = \boldsymbol{q}_t'^{\top} \cdot \boldsymbol{q}_{t+1}$ is the rating of the ground-truth positive example $\boldsymbol{q}_{t+1}$, whereas $r_{t,k} = \boldsymbol{q}_t'^{\top} \cdot \boldsymbol{q}_k$ is the rating of a randomly selected negative query $\boldsymbol{q}_k$.

For the KL-divergence term, we minimize the KL-divergence between the estimated posterior distribution $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ and standard Normal distribution every time step, and the analytical solution is:

$$\mathcal{L}_{\text{kl}} = \sum_{t=1}^{T-1} \sum_{d=1}^{D} (\sigma_{t,d}^2 + \mu_{t,d}^2 - 1 - \log \sigma_{t,d}^2), \tag{9}$$

where $D$ is the dimension of $\boldsymbol{\mu}_t$ and $\Sigma_t$.

*3.6.2 Next intent prediction.* To enable the model with the capability to predict next intent according to previous reformulation behaviors, we minimize the cross entropy loss between the output of the intent encoder at time step $t$, i.e., $\boldsymbol{i}_t'$ and the input intent of the next step $\boldsymbol{i}_{t+1}$,

$$\mathcal{L}_{\text{intent}} = \sum_{t=1}^{T-1} \ell_{\text{ce}}(\boldsymbol{i}_t', \boldsymbol{i}_{t+1}), \tag{10}$$

where $\ell_{\text{ce}}$ is the cross entropy loss.

*3.6.3 Preventing anisotropic issues.* As mentioned above, the PLLMs have anisotropic issues where only a few dimensions of query embeddings are used to encode the information related to the input, resulting in high similarity scores of different queries (average cosine similarities of all query pairs are greater than 0.9) that limit the model's representation capacity. We also visualize the query feature distribution in Appendix A.1 indicating that the anisotropic problems of PLLMs indeed exist in our task. To mitigate this issue, we adopt a simple uniformity loss as regularization [38], termed as *DeepWhitening*. DeepWhitening forces the query embeddings to distribute isotropically in the latent space mapped by $MLP_{\theta_2}$ in Eq. 3. Formally,

$$\mathcal{L}_{\text{DW}} = \log \sum_{i,j} \left[ e^{-\|\boldsymbol{q}_i - \boldsymbol{q}_j\|_2^2/2} \right], \tag{11}$$

where $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are two input query embeddings within the current batch. With such regularization term, we can omit burdensome fine-tuning/post-whitening, and enable an end-to-end optimization procedure directly for query recommendation tasks.

The overall objective function is simply a summation of the terms described above. Denote the model's parameters as $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]$, then

$$\begin{aligned} \theta^* = \arg\min_{\theta} \mathcal{L}(\theta) =& \mathcal{L}_{\text{rec}}(\theta_1, \theta_2, \theta_3, \theta_4) + \mathcal{L}_{\text{kl}}(\theta_2, \theta_3) \\ &+ \mathcal{L}_{\text{intent}}(\theta_1) + \mathcal{L}_{\text{DW}}(\theta_2). \end{aligned} \tag{12}$$

# 4 EXPERIMENTS

In this section, we conduct experiments to evaluate the effectiveness of IVAE by answering the following research questions:

- **RQ1**: What's the performance of IVAE on real-world query recommendation tasks compared with other methods? (Sec.4.2)
- **RQ2**: Are the key designs in IVAE, such as the PLLM with proper regularization, the utilization of intent information beneficial for satisfactory improvement? (Sec. 4.3)
- **RQ3**: How does the performance of IVAE vary with different query frequencies and query lengths? (Sec. 4.3)

## 4.1 Experiment setups

*4.1.1 Datasets.* We use two real-world e-commercial datasets collected from search logs of Amazon, `Session-AU` and `Session-CA`, where AU and CA denote Australia and Canada, respectively. The queries within a session are first sorted according to the timestamps and then sliced into sub-sessions by purchase-leading queries. We only keep sessions longer than two. We also adopt publicly available sessions released by `BestBuy`[1]. The differences between the `BestBuy` dataset and the former two are that the sessions in `BestBuy` do not end with purchase behavior, and the sessions of `BestBuy` dataset may span several days or even months. Thus, the historical queries within a session may be more noisy and misleading for predicting the last query. Finally, we present the statistics of these datasets in Table 1. Following the common practice [11, 25], we randomly sample 10, 000 sessions from `Session-AU` and `Session-CA` for evaluation and test, respectively. For `BestBuy`, we sample 1, 000 instead. The rest of the sessions are used for training. We use the target queries that appeared in the test dataset as our recommendation candidate queries.

*4.1.2 Statistics of Intents.* In addition, we gather statistics on the intent types in our evaluation dataset from `Session-AU` as reported in Table 2. We first obtain the predictions of each consecutive query pair using the static intent estimator. We count the corresponding intent types of the last reformulation and report their ratios. The high ratio of Irrelevant intents suggests that reformulated queries may be related to historical queries instead of the current ones, necessitating the usage of historical queries. We also report the precision of the static intent estimator, which is trained ahead of other components using extremely limited annotations. Due to the imbalance and noisy nature of the intent estimator, instead of merely relying on its estimations for the prediction of the next query, we need to take intent evolution into account for the correct intent prediction. To further justify the necessity and rationality of the usage of intent dynamics, we report the distribution of each intent type w.r.t. their positions in a sequence using the static intent estimator in Fig. 2. We can observe that there exist discernible sequential patterns of each intent type. For instance, Equivalence is highly likely to appear in the early stage, while Substitution and Complement tend to appear afterward. Even if the intent estimator is not 100% correct, we can analyze the previous intent estimations and their positions to correct the next intent prediction.

**Table 1: Statistics of datasets.**

| Dataset | #sessions | #queries | avg. query freq. | avg. seq. len. |
|---|---|---|---|---|
| `Session-AU` | 1,621,374 | 1,575,659 | 6.05 | 6.23 |
| `Session-CA` | 516,117 | 724,602 | 5.72 | 4.07 |
| `BestBuy` | 83,305 | 97,690 | 5.83 | 4.97 |

*4.1.3 Baselines.* We select three types of baseline methods for a comprehensive comparison: 1) three PLLM-based methods, including DLKNN, Bert-Finetune, and SimCSE [12]; 2) two classical sequential recommendation models, SASRec [18] and Bert4Rec [37]; 3) two representative intent-aware query recommendation models, HRED [35] and RIN [17].

[1] https://www.kaggle.com/c/acm-sf-chapter-hackathon-big/data

**Table 2: Statistics of Intent types in Session-AU.**

| Intent | Equ | Spe | Sub | Gen | Com | Irr |
|---|---|---|---|---|---|---|
| Ratio | 5.53% | 8.06% | 8.65% | 6.38% | 11.81% | 59.57% |
| Estimator Precision | 95.49% | 76.61% | 72.99% | 72.31% | 92.96% | 76.99% |
| Average Position | 2.14 | 4.84 | 5.02 | 4.95 | 4.70 | 4.62 |

- **DLKNN**, **Bert-Finetune** and **SimCSE** utilize the same PLLM. DLKNN applies k-nearest neighbors directly on the output of PLLM to predict the next possible queries. Bert-Finetune finetunes the PLLM by maximizing the cosine similarity between consecutive query pairs. SimCSE enhances the Bert-Finetune with an additional contrastive learning objective.
- **SASRec** [18] first randomly initializes an embedding for each query and applies a stack of self-attention layers to predict the next possible queries. **Bert4Rec** [37]: The original implementation of Bert4Rec is similar to SASRec, but has the bi-directional self-attention layer. We enhance the Bert4Rec using PLLM with regularization for fair comparison.
- **HRED** [35] applies hierarchical RNN over query sessions. They first apply RNN to learn query representations. Another RNN is applied on top of these query representations to extract sequential patterns from query sessions. **RIN** [17] further enhances the HRED by explicitly modeling the differences between two query embeddings as reformulation. RIN also employs GNN methods to learn query embeddings from a term-query-website graph as the initialization of queries. As we do not have website information in the e-commercial scenario, we replace the query encoder components of HRED and RIN by the PLLM with regularization.

*4.1.4 Implementation Details.* We implement our model with Pytorch, and all the experiments are conducted on an Nvidia A100 GPU with 40GB memory. The model is optimized with Adam [20]. The pre-trained query encoder and static intent estimator adopt similar architecture as that of the BERT base [10] (12 layers, 768 hidden size). During training, these two models are fixed, and their outputs (query embeddings and static intent estimations) are fed into a two-layer MLP and three masked-self-attention layers (MSA) of hidden size 768. Two MSAs process the query embedding sequence to determine mean and variance separately, while one MSA processes the intent estimation sequence to predict next intent probabilities. The decoder is a six-head-self-attention layer of hidden size 768. We use a learning rate of 0.0001, batch size of 1024, and dropout rate of 0.5 for all datasets. To avoid overfitting, we employ early stopping with patience of 100 epochs. For a fair comparison, we tune the hyperparameters for all methods on the validation set.

## 4.2 Performance on Query Recommendation

We report the Top-K recommendation performance regarding Recall and NDCG of IVAE compared with other baseline methods in Table 3. Our method IVAE consistently outperforms all baseline methods in all metrics, which strongly demonstrates the efficacy of IVAE. Besides, we have the following observations: 1) Compared to DLKNN, Bert-Finetune, and SimCSE, our model IVAE consistently improves the query recommendation performance, which proves the effectiveness and necessity of utilizing all historical session information. 2) Compared to the second-best model Bert4Rec on Session-AU dataset, our model consistently improves performance by over 15%. These results strongly support the effectiveness of IVAE, as we equip Bert4Rec with PLLM and MLP, from which we can conclude that the improvement is not majorly contributed by adding more parameters. The improvements come from disentangling the latent variables using VAE to make the model more robust to noise. Moreover, the intent information, e.g., the estimation of whether the previous query is irrelevant, also improves the IVAE's robustness to the noisy and misleading query records and narrows the range of candidate queries. 3) Compared to DLKNN, Bert-Finetune performs even worse on all three datasets. This phenomenon indicates that anisotropic issues indeed exist in our task. The collapse of Bert-Finetune supports the necessity of uniformity regularization. 4) The baseline models that utilize historical queries pairwisely or sequentially do not outperform the DLKNN on the BestBuy dataset. The reason may be that the sessions in this dataset span several days or even longer, making the historical queries more noisy and misleading since the search goal is highly likely to be different from that revealed by the historical queries. Despite the noisy and misleading historical records, IVAE can still outperform strong baseline DLKNN on this dataset, as IVAE can capture the noisy interaction through VAE and irrelevant intent estimations.

## 4.3 Analysis of IVAE (RQ2 and RQ3)

*4.3.1 Ablation studies.* Our main contribution is proposing a novel query recommendation method that captures the intent dynamics using PLLMs with appropriate regularizers. Thus, the aim of the ablation study is to examine the effectiveness of 1) the Intent Encoder, 2) the PLLM, and 3) the uniformity regularizer, i.e., DeepWhitening. The comparisons, such as Transformer vs. RNN, VAE vs. AE/DAE, and Uniformity vs. post-whitening, are not directly related to the main goal of the ablation study and are omitted for brevity. For variants without intent, we delete the Intent-Encoder; for variants without PLLM, we use a learnable embedding as the input feature for each query; for variants without DeepWhitening, we remove $\mathcal{L}_{\text{DW}}$ from the final loss computation. We report the performance of different variants compared with the original IVAE on Session-AU dataset in Table 4, and we report the average performance drop in the last row compared to that of IVAE.

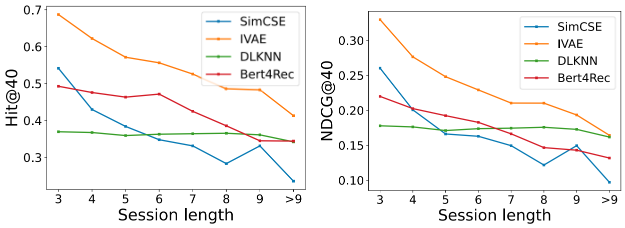
From the performance of different variants of IVAE, we have the following observations: 1) If we only remove the Intent-Encoder, the model IVAE's performance will drop by 8.54%. We also report the comparison between VAE and IVAE on all three datasets in terms of Recall@40 and NDCG@40 in Appendix A.2, which indicates that IVAE outperforms VAE on all datasets. These results strongly support the effectiveness and necessity of utilizing intent information for query recommendations. As explicitly modeling such intent information can not only narrow down the query recommendation candidates but also make the model robust to noisy and misleading interactions. Furthermore, the design of IVAE also equips the model with the capability of controllable generation. We report a case study over the controllable generation in Appendix A.3. 2) Compared to the variants without PLLMs on columns 2,

**Table 3: Overall Comparison. Boldface indicates the best performance while the underlined one indicates the second best. The proposed IVAE achieves the best performance over all datasets across a variety of metrics**

| Dataset | Metric | DLKNN | Bert-Finetune | SimCSE | SASRec | Bert4Rec | HRED | RIN | IVAE | Improv. |
|---|---|---|---|---|---|---|---|---|---|---|
| Session-AU | Recall@15 | 0.2626 | 0.1471 | 0.2841 | 0.2741 | 0.3290 | 0.2689 | 0.2333 | **0.4127** | 25.44% |
| | Recall@20 | 0.2917 | 0.1651 | 0.3144 | 0.3058 | 0.3697 | 0.3066 | 0.2719 | **0.4583** | 23.97% |
| | Recall@40 | 0.3621 | 0.2178 | 0.3938 | 0.3919 | 0.4819 | 0.4091 | 0.3739 | **0.5762** | 15.69% |
| | NDCG@15 | 0.1518 | 0.0761 | 0.1569 | 0.1615 | 0.1671 | 0.1274 | 0.1102 | **0.2188** | 30.94% |
| | NDCG@20 | 0.1587 | 0.0803 | 0.1640 | 0.1691 | 0.1767 | 0.1363 | 0.1193 | **0.2296** | 29.94% |
| | NDCG@40 | 0.1731 | 0.0911 | 0.1803 | 0.1867 | 0.1997 | 0.1572 | 0.1402 | **0.2537** | 27.04% |
| Session-CA | Recall@15 | 0.2259 | 0.1628 | 0.2325 | 0.1505 | 0.2246 | 0.1217 | 0.1553 | **0.3731** | 60.47% |
| | Recall@20 | 0.2453 | 0.1808 | 0.2513 | 0.1682 | 0.2580 | 0.1432 | 0.1829 | **0.4228** | 63.87% |
| | Recall@40 | 0.2945 | 0.2289 | 0.3055 | 0.2246 | 0.3444 | 0.2066 | 0.2665 | **0.5443** | 58.04% |
| | NDCG@15 | 0.1372 | 0.0934 | 0.1400 | 0.0850 | 0.0970 | 0.0564 | 0.0728 | **0.1899** | 35.64% |
| | NDCG@20 | 0.1417 | 0.0977 | 0.1444 | 0.0892 | 0.1129 | 0.0614 | 0.0793 | **0.2017** | 39.68% |
| | NDCG@40 | 0.1518 | 0.1075 | 0.1555 | 0.1007 | 0.1384 | 0.0744 | 0.0963 | **0.2265** | 45.65% |
| BestBuy | Recall@15 | 0.1150 | 0.0580 | 0.0890 | 0.0400 | 0.0560 | 0.0620 | 0.0680 | **0.1310** | 13.91% |
| | Recall@20 | 0.1400 | 0.0740 | 0.1120 | 0.0570 | 0.0730 | 0.0790 | 0.0860 | **0.1510** | 7.86% |
| | Recall@40 | 0.1910 | 0.1280 | 0.1650 | 0.0970 | 0.1340 | 0.1320 | 0.1600 | **0.2110** | 10.47% |
| | NDCG@15 | 0.0628 | 0.0242 | 0.0484 | 0.0159 | 0.0266 | 0.0263 | 0.0282 | **0.0640** | 1.91% |
| | NDCG@20 | 0.0680 | 0.0280 | 0.0538 | 0.0199 | 0.0307 | 0.0304 | 0.0324 | **0.0688** | 1.17% |
| | NDCG@40 | 0.0791 | 0.0389 | 0.0646 | 0.0281 | 0.0431 | 0.0411 | 0.0474 | **0.0809** | 2.28% |

4, and 6, the IVAE outperforms substantially. This could be attributed to the sparsity nature of the dataset, as the query embeddings merely learned from such highly noisy and sparse training datasets encode fewer semantics than those from PLLM. 3) From another aspect, by comparing IVAE with variants without DeepWhitening on columns 3, 4, and 5, we can conclude that DeepWhitening helps improve performance by addressing the issues of anisotropy. Although the MLP can also alleviate the anisotropic issues to a certain extent, DeepWhitening further improves by directly penalizing the anisotropic query embeddings. It is simple yet effective and provides a plug-and-play mechanism without fine-tuning the cumbersome PLLMs. We also further examine the effect of DeepWhitening in Appendix A.1. Because of the limited space, we do not report the version directly using the normalized [CLS] embedding as model input, as the model collapsed to the degenerate representation.

*4.3.2 Impacts of session length.* In this section, we split the test sessions into eight groups according to their length. We compare four models' performance over different groups and report the experiment results in Fig. 4. We chose DLKNN, SimCSE and Bert4Rec for comparison, as they achieve second best performances on three datasets respectively. Within Fig. 4, we can observe that: 1) IVAE outperforms the other baseline models over all groups on both Recall@40 and NDCG@40. These results validate the effectiveness of IVAE. 2) The performance of all models except DLKNN decreases as the length of sessions increases. The reason might be that there are more noisy queries for the long sessions. Without the capability of being robust to the noise, the performance of models like Bert4Rec decreases dramatically. DLKNN only recommends queries based on semantic similarity without considering historical queries. Thus the performance stays invariant to the session length. 3) SimCSE performs better than Bert4Rec and DLKNN on short sessions, as it introduces additional uniformity loss over query representations and fine-tunes the PLLM by looking one step back. These factors make SimCSE good at short sessions. This phenomenon also supports IVAE as we add uniformity loss and train the model by looking back to all previous queries. Thus, IVAE performs much better for short sessions but is also more robust to the noise in long sessions.



**(a) Recall@40** **(b) NDCG@40**

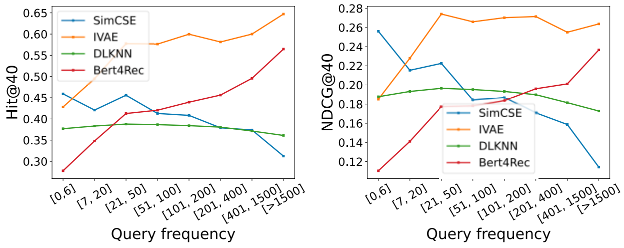
**Figure 4: Performance comparison w.r.t. session lengths.**

*4.3.3 Impacts of query frequency.* We divide test sessions into eight groups based on the frequency of target queries in the training dataset. We then compare the performance of second-best models (SimCSE, DLKNN, and Bert4Rec) with our model. We report the results in Fig. 5 and have the following observations: 1) IVAE has better results for all groups of queries that appear more than seven times in the training dataset. 2) DLKNN and SimCSE outperform IVAE for queries with low frequency, as these rare queries usually have specific descriptions and search goals. Thus, historical information is highly likely to be irrelevant. IVAE and Bert4Rec that look back to all previous queries can extract little signal for such query recommendation, resulting in decreased performance. 3) The performance of IVAE and Bert4Rec improves as the frequency of queries increases, while SimCSE performs the opposite. The reason

**Table 4: Ablation Study**

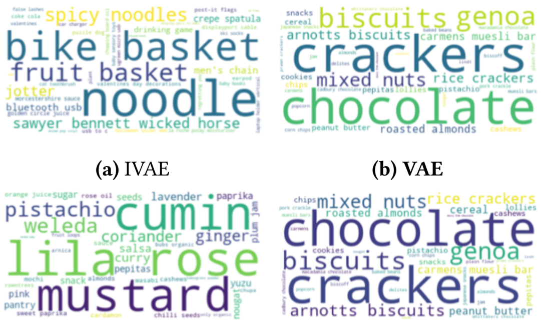
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Components | Intent | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| | PLLM | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |
| | DeepWhitening | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ |
| Session-AU | Recall@15 | 0.3781 | 0.1385 | 0.3408 | 0.1355 | 0.3338 | 0.1289 | **0.4127** |
| | Recall@20 | 0.4274 | 0.1632 | 0.3874 | 0.1589 | 0.3793 | 0.1531 | **0.4583** |
| | Recall@40 | 0.5415 | 0.2339 | 0.4994 | 0.2267 | 0.4918 | 0.2166 | **0.5762** |
| | NDCG@15 | 0.1908 | 0.0688 | 0.1703 | 0.0665 | 0.1655 | 0.0620 | **0.2188** |
| | NDCG@20 | 0.2024 | 0.0746 | 0.1813 | 0.0720 | 0.1762 | 0.0677 | **0.2296** |
| | NDCG@40 | 0.2258 | 0.0890 | 0.2042 | 0.0858 | 0.1992 | 0.0807 | **0.2537** |
| | avg. drop | -8.54% | -64.26% | -17.03% | -65.33% | -18.79% | -67.03% | 0% |

might be that the frequent queries are more likely to be relevant to the historical queries that are two or more time steps back, and IVAE and Bert4Rec can learn the global relevance from all historical queries. IVAE is superior to Bert4Rec as it is more robust to the noisy and misleading queries when considering all previous queries.



**(a) Recall@40** **(b) NDCG@40**

**Figure 5: Performance comparison with different target query frequencies. IVAE performs better as the target query's frequency increases.**



**(a)** IVAE **(b) VAE**

**(c) DLKNN** **(d) Bert4Rec**

**Figure 6: Top-40 Ranked Queries from IVAE (our method), VAE, DLKNN and Bert4Rec. Even though the input queries (in Table 5) are noisy, IVAE can still give correct predictions.**

## 4.4 Case Study

Finally, we analyze specific cases of the predicted queries by comparing the top-40 ranked queries from: IVAE, DLKNN, Bert4Rec, and VAE (a variant of our model without intent information). We report the recommended queries of these four models in the form of the word cloud, as shown in Fig. 6, where the font size corresponds to the rank of the query. We also report the input information of these four models in Table 5. From the input data, we can observe that there is a noisy query *dog toy*, and the target query *noodle* is a complementary reformulation from the previous query. In Fig. 6, we can observe that the IVAE correctly predicts the next potential query *noodle*, even though (as shown in Table 5) the static intent estimation is incorrect, as the dynamic intent estimator can re-estimate the potential intent considering the intent evolution. Furthermore, the intent information also introduces diversity into the query recommendation, e.g., *fruit basket, and drinking game*, which are not semantically relevant to the historical queries but might be complementary and inspiring queries.

**Table 5: Case Study of Input Data**

| | |
|---|---|
| Historical Queries | dog toy, Genoa Foods, apricot |
| Target Query | noodle |
| Static intent estimator | Irrelevant |
| Dynamic intent estimator | Complement |

## 5 CONCLUSION

In this paper, we have proposed IVAE for explicitly modeling the complex reformulation intent evolving from both semantic and product-related perspectives. To reach such desiderata, we first explicitly define the six types of reformulation intents according to the relationships between the desired items. Then, we extract an additional reformulation intent sequence from the original query sequence and apply the self-attention module over these two sequences, respectively, to learn the sequence transitions of queries and intents. In the end, we propose an intent-aware decoder that can generate candidate queries using the dynamic next intent estimations and next query estimations. Extensive experiments on real-world query recommendation datasets demonstrate the efficacy of the proposed method.

## 6 ACKNOWLEDGEMENT

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

# REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.*

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.*

[3] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. 2009. Query suggestions using query-flow graphs. In *Proceedings of the 2009 workshop on Web Search Click Data.* 56–63.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. 1877–1901.

[5] Jia Chen, Jiaxin Mao, Yiqun Liu, Ziyi Ye, Weizhi Ma, Chao Wang, Min Zhang, and Shaoping Ma. 2021. A Hybrid Framework for Session Context Modeling. 1–35.

[6] Jia Chen, Jiaxin Mao, Yiqun Liu, Fan Zhang, Min Zhang, and Shaoping Ma. 2021. Towards a better understanding of query reformulation behavior in web search. In *Proceedings of the Web Conference 2021.* 743–755.

[7] Xiaokai Chu, Jiashu Zhao, Lixin Zou, and Dawei Yin. 2022. H-ERNIE: A Multi-Granularity Pre-Trained Language Model for Web Search. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval.* 1478–1489.

[8] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

[9] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017.*

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

[11] Rui Feng, Chen Luo, Qingyu Yin, Bing Yin, Tuo Zhao, and Chao Zhang. 2022. CERES: Pretraining of Graph-Conditioned Transformer for Semi-Structured Session Data.

[12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021.* 6894–6910.

[13] Jiafeng Guo, Xueqi Cheng, Gu Xu, and Xiaofei Zhu. 2011. Intent-aware query similarity. In *Proceedings of the 20th ACM international conference on Information and knowledge management.* 259–268.

[14] Qi He, Daxin Jiang, Zhen Liao, Steven CH Hoi, Kuiyu Chang, Ee-Peng Lim, and Hang Li. 2009. Web query recommendation via sequential query prediction. In *2009 IEEE 25th international conference on data engineering.* IEEE, 1443–1454.

[15] Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. 2003. Relevant term suggestion in interactive web search based on contextual information in query session logs. 638–649.

[16] Haoming Jiang, Tianyu Cao, Zheng Li, Chen Luo, Xianfeng Tang, Qingyu Yin, Danqing Zhang, Rahul Goutam, and Bing Yin. 2022. Short Text Pre-training with Extended Token Classification for E-commerce Query Understanding.

[17] Jyun-Yu Jiang and Wei Wang. 2018. RIN: Reformulation inference network for context-aware query suggestion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 197–206.

[18] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018.* 197–206.

[19] Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations.

[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

[22] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models.

[23] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020.* 322–330.

[24] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. 2021. Lightweight Self-Attentive Sequential Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.* 967–977.

[25] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference.* 689–698.

[26] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 3365–3375.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

[28] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. 2021. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval.* 1608–1612.

[29] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query suggestion using hitting time. In *Proceedings of the 17th ACM conference on Information and knowledge management.* 469–478.

[30] Bhaskar Mitra. 2015. Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval.* 3–12.

[31] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2021. On the study of transformers for query suggestion. 1–27.

[32] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

[33] Christian Sengstock and Michael Gertz. 2011. Conquer: A system for efficient context-aware query suggestions. In *Proceedings of the 20th international conference companion on World wide web.* 265–268.

[34] Yang Song, Dengyong Zhou, and Li-wei He. 2012. Query suggestion by constructing term-transition graphs. In *Proceedings of the fifth ACM international conference on Web search and data mining.* 353–362.

[35] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management.* 553–562.

[36] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval.

[37] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.* 1441–1450.

[38] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119).* 9929–9939.

[39] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. 2022. ContrastVAE: Contrastive Variational AutoEncoder for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 2056–2066.

[40] Qitian Wu, Chenxiao Yang, Shuodian Yu, Xiaofeng Gao, and Guihai Chen. 2021. Seq2Bubbles: Region-Based Embedding Learning for User Behaviors in Sequential Recommenders. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021.* 2160–2169.

[41] Xiaohui Yan, Jiafeng Guo, and Xueqi Cheng. 2011. Context-aware query recommendation by learning high-order relation in query logs. In *Proceedings of the 20th ACM international conference on Information and knowledge management.* 2073–2076.

[42] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer.

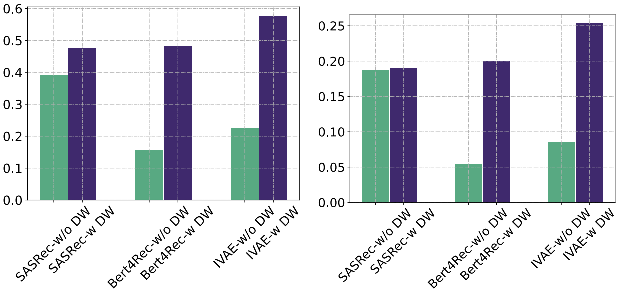
[43] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019.* 4320–4326.

[44] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in Baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 4014–4022.

# A ADDITIONAL EXPERIMENTS

## A.1 Effects of DeepWhitening

DeepWhitening provides a simple yet effective mechanism for high-level sentence tasks e.g. query recommendation. It provides the possibility to combine the PLLMs and the sequential models while avoiding anisotropy. It plays a plug-and-play interface and can also be applied to existing sequential approaches. In this section, we equip the SASRec, Bert4Rec with the PLLM using the DeepWhitening, and report the results in Fig. 7, where SASRec-w/o DW, Bert4Rec-w/o DW represent the variants without DeepWhitening, SASRec-DW, and Bert4Rec-DW with DeepWhitening on the contrary. Compared to the vanilla version that learns query embedding from scratch using the training data, the performance in terms of both Recall@40 and NCDCG@40 of models with DeepWhitening improves substantially. These phenomena prove the potential of utilizing PLLMs and the effectiveness of DeepWhitening.



(a) Recall@40 (b) NDCG@40

**Figure 7: The effect of DeepWhitening. We equip the DeepWhitening technique to different backbone models and compare their corresponding performance. DeepWhitening can boost the performance of all backbones.**

To investigate whether the proposed DeepWhitening can really mitigate the anisotropic issue, we further plot and compare the query feature distribution of `Session-CA` with and without DeepWhitening in Fig. 8. As demonstrated in the figure, without the regularization of DeepWhitening, the query features tend to be concentrated, thus, are hard to discriminate. By contrast, DeepWhitening encourages query features to distribute uniformly in the hypersphere.

## A.2 Effects of Intents

We further compare IVAE with VAE that simply removes the Intent-Encoder of IVAE on all three datasets. From the experimental results reported in Fig. 9, we can observe that the intent information improves the query recommendation performance consistently on all datasets in terms of Recall@40 and NDCG@40. Especially for dataset `Session-CA`, which has the most sparse sessions, the IVAE managed to improve the Recall@40 compared to VAE is improved from 0.2957 to 0.5443.

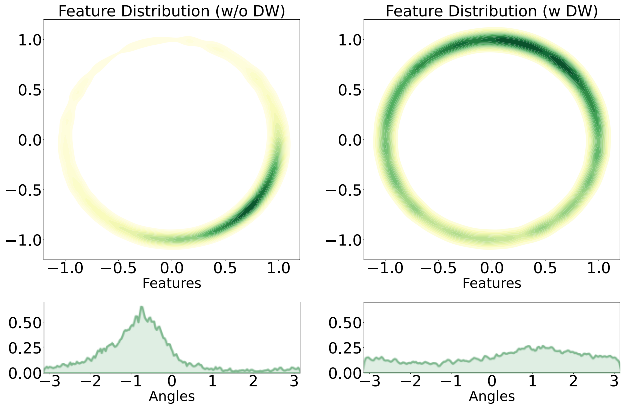
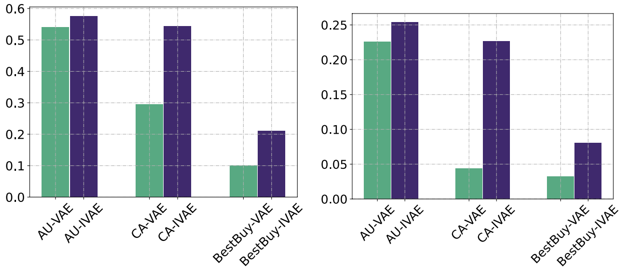


**Figure 8: We plot the query feature distributions (the first two dimensions) with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ and von Mises-Fisher (vMF) KDE on angles for randomly selected 10,000 queries from `Session-CA` dataset. Left is the feature distribution without DeepWhitening, while the right is that with DeepWhitening.**



(a) Recall@40 (b) NDCG@40

**Figure 9: The effect of Intent. We compare the performance of our method with (IVAE) and without (VAE) the intent information on `Session-AU`, `Session-CA` and `BestBuy`, respectively. Adding the intent information can improve the model's performance on all datasets.**

**Table 6: Case Study of Controllable Generation from `Session-AU`. We report the top-5 ranked queries from** IVAE **and Bert4Rec.**

| rank | IVAE with Equivalent intent estimation | Bert4Rec |
|---|---|---|
| 1 | **face mask reusable** | pm 2.5 filters for face mask |
| 2 | face masks virus protection | kids face mask |
| 3 | face mask | **face mask reusable** |
| 4 | face mask disposable | reusable face mask |
| 5 | face masks disposable | weddingstar face masks |

## A.3 Controllable Generation

The design of IVAE also equips the model with the capability of controllable generation. Specifically, we can affect the query recommendation process of the intent-aware decoder by manipulating its input from the dynamic intent encoder. For example, we manually enlarge the probability estimation of equivalent reformulation intent and report the top-5 ranked queries for recommendation in Tab. 6. For this case, we have historical queries as {*face masks virus protection, face mask disposable kids, face mask reusable kid, face mask disposable*} and the target query is *face mask reusable*. From Tab. 6, we can observe that the top-ranked queries from IVAE are all equivalent reformulations compared to those from Bert4Rec.