# Semi-Explicit MMoE via Heterogeneous Multi-Task Learning for Ranking Relevance

Ziyang Liu*
Junqing Chen*
liuziyang7@jd.com
chenjunqing@jd.com
JD.com
Beijing, China

Yunjiang Jiang†
Yue Shang
yunjiang.jiang@jd.com
JD.com
Mountain View, CA

Wei Xiong
Sulong Xu
JD.com
Beijing, China

Zhaomeng Cheng
Bo Long
JD.com
Beijing, China

Lingfei Wu
Yun Xiao
JD.com
Mountain View, CA

Di Jin
Tianjin University
Tianjin, China

## ABSTRACT

Relevance and attractiveness are two important attributes of search ranking results. While these two attributes are well-aligned for simple (query, result) pairs, they start to diverge for more nuanced, sophisticated pairs, especially when click-bait elements enter the result presentation.

In this work, we explore how to jointly optimize both objectives in a state of the art ensemble framework, the multi-gate mixture of experts (MMoE) model, with explicit expert choice for different objectives. We also compare with less explicit expert gating mechanism.

In order to adequately quantify the model improvements, we introduce the notion of bi-metric linear AUC that takes into account both relevance and user preference metrics under a one-parameter family of model scores, which generalizes the usual ROC or Precision/Recall AUC. We argue that these fine-grained metrics are better aligned with typical search engine business requirements.

Due to the scarcity of relevance labels, we take a distillation approach, relying on state of the art NLP models such as BERT to produce high quality relevance predictions as labels. To differentiate among multiple degrees of relevance, we experiment with several extensions of cross entropy losses in order to capture the linear ordering of the relevance labels as well as their multi-categorical nature. The experimental results shows the new semi-explicit MMOE model via heterogeneous task learning often achieve the best performance. Finally, we successfully push the newly proposed model into a real-world online e-commerce search system. We find this model create more business value for the company by helping the user find the item he or she wants to buy more quickly. The source code of our work is publicly available at https://github.com/user8831222/HMMoE.

## KEYWORDS

multi-task learning, mixture of experts, relevance estimation

**ACM Reference Format:**
Ziyang Liu, Junqing Chen, Yunjiang Jiang, Yue Shang, Wei Xiong, Sulong Xu, Zhaomeng Cheng, Bo Long, Lingfei Wu, Yun Xiao, and Di Jin. 2021. Semi-Explicit MMoE via Heterogeneous Multi-Task Learning for Ranking Relevance. In *KDD '21: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 14–18, 2021, Virtual*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

For ad-hoc information retrieval, *relevance estimation*, directly affecting which items will be shown on the result page, plays an important role in the ranking model. In the last decade, neural-based deep models have been widely applied in the real-world search system or recommendation system. There are two clear research routes among them: the first one focuses on the relevance learning [1–4] while the other is oriented by the user behavior modeling [5, 6]. In the offline model training phase, relevance learning usually uses the relevance annotations by external assessors as supervised labels. Different from it, the user behavior modeling usually uses the user click-through rate or user conversation rate as the supervised information in order to simulate the real user feedback.

In a specific application scene such as web search or e-commerce search, user behavior feedback typically deviates from semantic relatedness[7]. A straightforward example of a real e-commerce search can be seen in Figure 1. Although the item of lighter fluid is not semantically equal to lighter, it satisfied more user's purchase needs and won more users' satisfaction than other items. Hidden purchase intention implied in the query phase, click-bait elements

*Both authors contributed equally to this research.
†Corresponding author.


*SIGKDD '21, August 14–18, 2021, Virtual*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

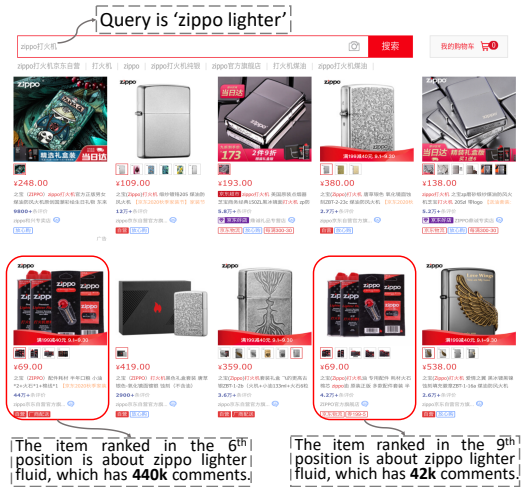



**Figure 1: A real example in the e-commerce search. The query is 'zippo lighter'. Apart from lighter, the top 10 listed items also include lighter fluid (surrounded by the red frame). Lighter fluid is not semantically equal to lighter, but it has a high sales volume which is proportional to the number of comments.**

in the listed items and other elements will result in the final mismatching phenomenon, which happens in the relevance and user feedback.

An accurate and satisfactory e-commerce search system should consider item relevance and user preference feedback at the same time. Many studies[8–10] investigate the relationship between the user behavior model and evaluation metric, and propose some more sophisticated metrics of the search system. For example, C/W/L framework[8] formalizes the connections between the user behavior model and evaluation metric.

Recently, multi-task learning is proposed, which integrates at least two different tasks into the same learning objective. A single task is noisy, but multi-task learning (also can be viewed as one kind of data augmentation [11]) mitigates the noise by comprehensively fitting multiple tasks, and thus commonly outperforms the single-task learning. Two representative works in multi-task learning are a mixture of experts [12] (MoE) and multi-gate mixture of experts [13] (MMoE). The main difference between MoE and MMoE is the number of gating network: MoE uses one gating network to select which experts to perform computations, while MMoE allows different tasks to select different experts.

Many earlier work demonstrates that MMoE indeed can juggle both tasks of relevance learning and user behavior modeling. However, homogeneity of tasks is a requirement in existing MMoE applications, making it inflexible to solve heterogeneous tasks that are ubiquitous in the real application scene. For example, the original MMoE can solve two pointwise-data tasks in which one is for click-through rate (CTR) prediction and the other is for click conversion rate (CVR) prediction. But for the pointwise-data task (such as relevance learning) combined with the pairwise-data task (such as user preference learning), the original MMoE cannot provide an effective solution.

In summary, when using MMoE for ranking relevance estimation, the following three main problems still exist:

- **Without considering partially ordered user behavior feedback**. User preferences of items are most often relative: a click on item A does not mean item B is irrelevant or of low quality, but only that the current user prefers item A slightly more than item B. Furthermore, the positive feedback events are very sparse among all the displayed results, resulting in serious class imbalance. Thus the typical pointwise architecture in relevance learning is unsuitable for preference learning.
- **No fair metric to balance relevance learning and preference learning**. Current models [1–3] use human-labeled relatedness scores (maybe from 1 to 5, bigger number means higher relevance) or user feedback signals (such as click, purchase or wish-list behaviors) as the training label, the underlying assumption being that the definition of relevance is completely dependent on the choice between these two labels. Typically semantic relevance score is used as a filter while user feedback prediction dictates the final ranking. We argue however such a strategy is sub-optimal and a mixed ranking strategy yields better precision and recall.
- **Weak interpretability of MMoE**. In MMoE framework, the dynamic gating network is applied to optimize each learning task. The whole updating process of the gating network is like a black box, we cannot control its evolution and even cannot interpret its final predictions sometimes. In an e-commerce search, understanding the functional role of each expert model is helpful for subsequent data and training improvement, even though it is only a small step towards complete model interpretability.

To tackle the above problems, in this paper, we first investigate the relevance judgment based on human assessors. In a real-world scene such as e-commerce search, we not only hope that search system can show the most relevant items to users, but also that users would click and even purchase the displayed items in the drop-down list. Based on these requirements, we propose a new relevance judgment metric which synthetically takes into account the element of semantic relatedness between query and item, and the element of user behavior feedback.

Following the new metric, we then propose an improved MMoE to deal with two heterogeneous tasks represented by pointwise data and pairwise data respectively. The pointwise-data task captures the semantic match feature, while the pairwise-data task captures the partial order relation of user behaviors. In offline experiments, we compare the different benefits from homogeneous task leaning and heterogeneous task learning.

Another pain point of existing MMoE frameworks is that they apply implicit invocation to multi experts. To make MMoE's expert selection process more interpretable and adapt with the feature of e-commerce search, we redesign the gating network using two strategies: explicit invocation (freezing the gating network's output) and semi-explicit invocation (freezing the gating network's input). In experiments, we compare the effects brought by both settings and

verify the advantage of semi-explicit gating networks. We hope that this work can provide some help for studying a more interpretable MMoE model, even multi-task learning.

Our main contributions are three-fold:

1. We develop a new relevance evaluation metric, which can measure the relevance degree from the semantics and online feedback in a more holistic way. This metric is helpful to build a reliable and attractive search system, and also helps the subsequent researchers better study the topic of ranking relevance according to such a new measurement.

2. We overcome MMoE's intrinsic limitations of the homogeneous task processing and implicit expert invocation, and thus design a strengthened MMoE, which can well deal with heterogeneous tasks, including label sources and training example format, and have good interpretability.

3. Finally, by thorough comparison experiments on a large-scale public dataset and in-house dataset, we verify that 2 indeed improves the metrics introduced in 1, along with other traditional metrics.

## 2 RELATED WORK

### 2.1 Works on the search relevance metric

Three elementary metrics for binary classification tasks are precision, recall, and accuracy. While accuracy can be computed independently on how mini-batches are chosen, precision and recall depend on the choice of mini-batch.

Note that the context of user feedback data such as clicks and orders, labels are naturally binary and all three metrics above are commonly used ([14] §4.1).

We can construct higher order metrics based on these such as ROC AUC and precision/recall AUC when we fix mini-batches [15].

For human labels, since they are not necessarily binary, the above metrics do not capture the full range of label, unless we divide the labels based on some arbitrary threshold. Instead, it is common to use mulit-labeled metrics such as NDCG [10] and ERR [9], both of which are mini-batch dependent (or more precisely, session-wise). Using the equivalence between Mann–Whitney U test and ROC AUC [16], a natural generalization of the latter can also be defined in this context, namely the Kendall's Tau metric [17]. However, this has not been widely adopted presumably due to its mathematical obscurity. On the other hand, pointwise metrics with non-binary ordinal labels are highly uncommon since it would require some arbitrary choice of label transformation.

The literature on how to combine two or more different metrics into one is relatively sparse. [18] proposes counting the fraction of unanimous preferences by all the metrics as the final metric for each predictor. This works well when the metrics consist of binary components, which exclude batch-sensitive metrics like AUC, NDCG. Furthermore, it does not measure the performance of a parametric family of predictors.

Although these evaluation metrics have achieved some success in guiding how to design a good user behavior model, they just simulate user behaviors by continuously adjusting the computation method of relevance score. We argue that semantic relevance and user feedback are associated variables, so we design a new metric integrating both two variables together.

### 2.2 Deep Semantic Match

Representation-based methods and interaction-based methods are two kinds of classical semantic match works. The representation-based methods include DSSM [19], MVLSTM [20], ARC-I [2] and so on. The interaction-based methods include ARC-II [2], DRMM [21], MatchPyramid [1], K-NRM [22], DRMM-TKS [21], Conv-KNRM [23] and ESIM [24]. Duet model [3] utilizes both of the representation embedding and interaction embedding at the same time. In a real application, the representation-based methods calculate query's and document's (or title's) embedding in the offline phase and calculate their similarity score in the online phrase. While the interaction-based methods can capture more detailed match signals between query and document than the representation-based methods, they cannot calculate query's and document's embedding in advance and thus limit their online application.

In addition, the knowledge distillation framework with the teacher model of the pre-trained NLP models is popular, the representative examples among these are Tiny-BERT [25], DistilBERT [26] and BERT2DNN [4]. All of these methods use lightweight student models to imitate the teacher model's fitting ability. This solution provide a open opportunity for more sophisticated model design and exploit a new paradigm for traditional information retrieval.

### 2.3 Multi-Task Learning (MTL)

Neither traditional semantic match methods nor distillation framework are only limited to single-task learning. Considering the necessity of relevance and user preference in the web search or e-commerce search, researchers begin to focus on reforming the existing model into multi-task learning. By sharing representations between several similar tasks, Multi-Task Learning (MTL) can generalize better on the original objective task than single-task learning. The main types of MTL include: hard parameter sharing and soft parameter sharing [27]. Hard parameter sharing uses strictly shared hidden layers between all tasks, while soft parameter sharing uses different hidden layers for different tasks.

A representative work of hard parameter sharing is the Mixture of Experts (MoE) model [12]. It uses a trainable gating network to automatically select suitable experts for some specific tasks.

As a follow-up of MoE, Multi-gate Mixture of Experts (MMoE) [13] model train multiple gating networks at the same time, in order to assign different combinations of experts for different tasks. From the viewpoint of multi-modal distribution, MMoE makes each expert model more easily to focus its attention on capturing unique modal distribution.

ViLBERT [28] uses the interaction form of the co-attentional transformer layers to process image-material data and text-material data together. VilBERT model is pretrained on two tasks: 1) masked multi-modal modeling, i.e., inferring the semantics of masked words or image patch; 2) and multi-modal alignment prediction, i.e., inferring the relatedness between a sequence and an image.

Alibaba Taobao group proposes a Multi-IPW method [5] to concatenate click-through rate (CTR) task and click conversion rate (CVR) task by parameter sharing. However, both CTR and CVR tasks belong to the user preference learning domain. Different from it, we integrate two different-domain tasks into the MMoE model together.

# 3 MODEL DESCRIPTION

## 3.1 SUM: Semantic relatedness and User feedback Metric

Recall that for e-commerce search ranking, semantic accuracy and user feedback popularity are both important for the business. The former ensures smooth and reasonable user experience, while the latter more directly generates growth and revenue.

One common way to combine two metrics into one is via some simple bi-variate formula. A typical example is given by F1 score, which combines precision and recall of a binary classification model by taking the ratio of their geometric and arithmetic means:

$$F1 = \frac{2PR}{P+R}.$$

While precision or recall alone depends heavily on the threshold of the prediction score, the F1 score removes some of that uncertainty, and presents a more balanced evaluation of model accuracy. Inspired by the F1 score, we define a kind of relatively simple bi-metric named as SUM (Semantic relatedness and User feedback Metric):

$$\text{SUM} = \frac{2M_{\text{pt}}M_{\text{pr}}}{M_{\text{pt}}+M_{\text{pr}}} \tag{1}$$

where $M_{\text{pt}}$ and $M_{\text{pr}}$ are respectively the metrics of pointwise-data and pairwise-data task.

When we have a 1-parameter family of predictors $P(\eta)$, indexed by $\eta$, we can look at how the two metrics vary against one another by sweeping $\eta$ from 0 to 1. This is the principle behind metrics such as ROC AUC (Area Under the Curve) [29], as well as precision/recall AUC [15], and the less well-known negative precision/recall AUC, where the parameter $\eta$ takes the role of a threshold between positive and negative predictions. We shall abuse the notation $M_{\text{pt}}(\eta) = M_{\text{pt}}(P(\eta))$ and similarly for $M_{\text{pr}}$.

Let $\{(M_{\text{pt}}(\eta_1), M_{\text{pr}}(\eta_1)), \ldots, (M_{\text{pt}}(\eta_n), M_{\text{pr}}(\eta_n))\}$ be the set of all possible pairs of the two metrics obtained by varying the model parameter $\eta$. Let $H := \{\eta_1, \ldots, \eta_n\}$ be sorted in ascending order. We can then compute the area under the piece-wise linear interpolation using the trapezoid rule:

$$\text{AUC}^H_{\text{pt}} = \sum_{i=1}^{n} \frac{|M_{\text{pt}}(\eta_{i+1}) - M_{\text{pt}}(\eta_i)|(M_{\text{pr}}(\eta_i) + M_{\text{pr}}(\eta_{i+1}))}{2}. \tag{2}$$

Note that the formula is well-defined and agrees with geometric intuition even if $M_{\text{pt}}(\eta_i) = M_{\text{pt}}(\eta_j)$ for some $i \neq j$; in that case the trapezoid will have zero area.

For symmetry, we define the transposed version of the above AUC:

$$\text{AUC}^H_{\text{pr}} = \sum_{i=1}^{n} \frac{|M_{\text{pr}}(\eta_{i+1}) - M_{\text{pr}}(\eta_i)|(M_{\text{pt}}(\eta_i) + M_{\text{pt}}(\eta_{i+1}))}{2}, \tag{3}$$

Lastly we define the bi-metric linear AUC to be their average:

$$\text{BML-AUC}^H := \frac{\text{AUC}^H_{\text{pt}} + \text{AUC}^H_{\text{pr}}}{2}. \tag{4}$$

Note that the above finite sum definitions are of course the finite-difference trapezoid approximation of the Riemann-Stieltjes integral, where $dv$ stands for the total variation measure under $M_{\text{pt}}$ and $M_{\text{pr}}$:

$$\text{AUC}_{\text{pt}} := \frac{1}{2}\left(\int_0^1 M_{\text{pr}}(\eta)dv_{\text{pt}}(\eta) + \int_1^0 M_{\text{pr}}(\eta)dv_{\text{pt}}(\eta)\right) \tag{5}$$

$$\text{AUC}_{\text{pr}} := \frac{1}{2}\left(\int_0^1 M_{\text{pt}}(\eta)dv_{\text{pr}}(\eta) + \int_1^0 M_{\text{pt}}(\eta)dv_{\text{pr}}(\eta)\right). \tag{6}$$

Note also that if $M_{\text{pt}}$ and $M_{\text{pr}}$ are continuous of finite total variations (e.g., Lipschitz), only one of the two integrals are needed in each of the formulas above.

The anchor-free version of the bi-metric linear AUC is thus defined by

$$\text{BML-AUC} := \frac{\text{AUC}_{\text{pt}} + \text{AUC}_{\text{pr}}}{2}. \tag{7}$$

The natural parameter we use for the baseline is the linear combination weight between the individually trained pointwise and pairwise model, i.e.,

$$F_\eta := \eta F_{\text{pt}} + (1-\eta)F_{\text{pr}}. \tag{8}$$

Similar formula applies to the two tower outputs of the HMMoE model.

In theory, we need infinitely many $\eta_i$ points to compute the exact BML-AUC, but when the two metrics $M_{\text{pt}}$ and $M_{\text{pr}}$ are both discrete, such as in the case of precision/recall/accuracy or the classical ROC AUC, a finite number of points suffices.

More precisely, let $N$ be the number of test examples, $\mu_{\text{pt}}, \mu_{\text{pr}}$ be the two discrete metrics, and $L_i^{\text{pt}}, L_i^{\text{pr}}$ be pointwise and pairwise task labels, and $s_i, t_i$ be the scores of the two model outputs, for $i \in [N]$. In the case of AUC, each example consists of a complete test mini-batch, such as all items under a single query. Then we have

$$M_{\text{pt}}(\eta_i) = \mu_{\text{pt}}(L_i^{\text{pt}}, \eta_i s_i + (1-\eta_i)t_i), \tag{9}$$

and similarly for $M_{\text{pr}}(\eta_i)$. We only need to choose $\eta_i$'s so that $M_{\text{pt}}(\eta), M_{\text{pr}}(\eta)$ are constant for $\eta \in (\eta_i, \eta_{i+1})$ for all $i < n$.

Let $H_{\text{pt}}$ and $H_{\text{pr}}$ be the minimal sets satisfying the above condition, then $H := H_{\text{pt}} \cup H_{\text{pr}}$ will also be a minimal set for both metrics.

Below we present two algorithms to compute the minimal $H$'s in the case of accuracy and ROC AUC.

*Definition 3.1.* An increasing collection of anchors points $H = \{\eta_1 < \ldots < \eta_Q\}$ is said to be **saturated** with respect to the list of score pairs $\{(s_i, t_i) : i \in [N]\}$ and corresponding reward functions $R_i$ if the following holds

$$\frac{d}{d\eta}\sum_i R_i(s_i\eta + t_i(1-\eta)) = 0, \qquad \forall \eta \in [0,1] \setminus H. \tag{10}$$

Given a sequence of binary labels $L_i \in \{\pm 1\}$, let the accuracy reward be given by

$$R_i(s) = \mathbb{1}_{sL_i > 0}. \tag{11}$$

Similarly we define the ROC AUC reward function

$$R_B(\vec{s}) = \sum_{i,j \in B} \mathbb{1}_{s_i > s_j}\mathbb{1}_{L_i > L_j}. \tag{12}$$

Lemma 3.2. *For any binary label sequence $L_i \in \{\pm 1\}$, $i \in [N]$,*

**Algorithm 1** Accuracy Anchors

**Input:** $N$: the number of test examples
**Input:** $\Pi := \{(s_i, t_i) : i \in [N]\}$: the scores from two predictors
**Output:** $Q \in \mathbb{N}$: number of Riemann sum anchors points
**Output:** $H = \{\eta_k : k \in [Q]\}$: saturated w.r.t. $\Pi$ and $R_i$.

1: $k \Leftarrow 1$
2: $\eta_1 \Leftarrow 0$
3: $H \Leftarrow \{\eta_1\}$
4: **while** $\eta_k < 1$ **do**
5: $\quad \Psi_{i,k} := s_i\eta_k + t_i(1-\eta_k)$
6: $\quad \Delta := \{\Psi_{i,k}/(t_i - s_i) : \Psi_{i,k}(t_i - s_i) > 0, i \in [N]\}$.
7: $\quad$ **if** $\Delta = \emptyset$ **then**
8: $\quad\quad \eta_{k+1} \Leftarrow 1$
9: $\quad$ **else**
10: $\quad\quad \eta_{k+1} \Leftarrow \min\{\eta_k + \min\Delta, 1\}$
11: $\quad$ **end if**
12: $\quad H \Leftarrow H \cup \{\eta_{k+1}\}$
13: $\quad k \Leftarrow k+1$
14: **end while**
15: $Q \Leftarrow k$

**Algorithm 2** ROC AUC Anchors

**Input:** $N$: the number of test examples
**Input:** $\Pi := \{(s_i, t_i) : i \in [N]\}$: the scores from two predictors
**Input:** $\mathcal{P} \vdash [N]$: a partition of test examples into mini-batches (or sessions)
**Output:** $Q \in \mathbb{N}$: number of Riemann sum anchors points
**Output:** $H = \{\eta_k : k \in [Q]\}$: saturated w.r.t $\Pi$ and $R_B$.

1: $k \Leftarrow 1$
2: $\eta_1 \Leftarrow 0$
3: $H \Leftarrow \{\eta_1\}$
4: **while** $\eta_k < 1$ **do**
5: $\quad \Psi_{i,j,k} := (s_i\eta_k + t_i(1-\eta_k)) - (s_j\eta_k + t_j(1-\eta_k))$
6: $\quad D_{i,j} = s_j - s_i + t_i - t_j$
7: $\quad \Delta := \{\Psi_{i,j,k}/D_{i,j} : \Psi_{i,j,k}D_{i,j} > 0,\ i, j \in B \text{ for some } B \in \mathcal{P}\}$.
8: $\quad$ **if** $\Delta = \emptyset$ **then**
9: $\quad\quad \eta_{k+1} \Leftarrow 1$
10: $\quad$ **else**
11: $\quad\quad \eta_{k+1} \Leftarrow \min\{\eta_k + \min\Delta, 1\}$
12: $\quad$ **end if**
13: $\quad H \Leftarrow H \cup \{\eta_{k+1}\}$
14: $\quad k \Leftarrow k+1$
15: **end while**
16: $Q \Leftarrow k$

(1) *The output anchor points in Algorithm 1 are saturated with respect to the score pairs* $\Pi = \{(s_i, t_i) : i \in [N]\}$*, and accuracy reward sequence* $\{R_i : i \in [N]\}$.
(2) *Similarly anchor points in Algorithm 2 are saturated with respect to* $\Pi$ *and the batch AUC reward sequence* $\{R_B : B \in \mathcal{P}\}$*, where* $\mathcal{P} \vdash [N]$ *is a fixed contiguous partition of the positive integers up to* $N$.

*Furthermore, both algorithms terminate in finite steps.*

Proof. We prove the case for Accuracy metric only. The argument for ROC-AUC is similar. First we establish finite step termination. Note that $\Delta\eta_k > 0$ for all $k$. By definition of $\Delta\eta_k$ (Algorithm 1, line 6), for every $k \in \mathbb{N}$, there is an $i \in [N]$ such that

$$s_i\eta_k + t_i(1-\eta_k) = 0. \tag{13}$$

For fixed $s_i$, $t_i$, (13) has a unique solution for $\eta_k$ unless $s_i = t_i = 0$, which were excluded by the condition $\Psi_{i,k}(s_i - t_i) > 0$ from Operation 6. Thus there are only finitely many $\eta_k$. The condition that $\Delta \neq \emptyset$ then guarantees that the while loop eventually terminates.

To show saturation, it suffices to take some $k < Q$ and show that $R_i(s_i\eta + t_i(1-\eta))$ is constant for all $\eta \in (\eta_k, \eta_{k+1})$ and $i \in [N]$, which is a stronger statement (note that we are not aiming for the minimal saturated anchor set).

In the case of Algorithm 1, we see from the definition of $\Delta$ (line 6) and the fact that $\eta_{k+1} - \eta_k < \min\Delta$, that $s_i\eta + t_i(1-\eta)$ has the same sign for $\eta \in [\eta_k, \eta_{k+1})$. Thus $R_i(s_i\eta + t_i(1-\eta))$ is indeed constant in that interval by (11), which contains $(\eta_k, \eta_{k+1})$.

□

As a corollary, we have

Theorem 3.3. *Let* $M_{\text{pr}}$ *and* $M_{\text{pt}}$ *be metrics of either accuracy or ROC-AUC type, and let* $L_i^{\text{pr}}, L_i^{\text{pt}}$ *be arbitrary binary label sequences. Let* $H_{\text{pt}}$ *and* $H_{\text{pr}}$ *be the anchor points from Algorithm 1 and 2 respectively,* $H' = H_{\text{pt}} \cup H_{\text{pr}}$ *and* $H := H' \cup \{(\eta_k + \eta_{k+1})/2 : k < Q'\}$*. Then BML-AUC*$^H$ *as defined in* (4) *agrees with the following anchor-free expression:*

$$\begin{aligned}\text{BML-AUC} &:= \frac{1}{2}\left(\text{AUC}_{\text{pr}} + \text{AUC}_{\text{pt}}\right)\\ &= \frac{1}{4}\left[\left(\int_0^1 + \int_1^0\right)\left(M_{\text{pr}}(\eta)dv_{\text{pt}}(\eta) + M_{\text{pt}}(\eta)dv_{\text{pr}}(\eta)\right)\right]\end{aligned}$$

*where* $dv_{\text{pt}}(\eta), dv_{\text{pr}}(\eta)$ *stand for the total variation measure of* $M_{\text{pt}}$ *and* $M_{\text{pr}}$ *respectively.*

Proof. By definition of total variation integrals, they can be arbitrarily closely approximated by finite difference sums over any sequence of partitions of the unit interval $[0, 1]$, provided the maximal mesh size goes to 0. So in particular we can refine the partition given by $0 = \eta_1 < \eta_2 < \ldots < \eta_Q = 1$ from $H$. By definition of Riemann-Stieltjes,

$$\int_{\eta_k}^{\eta_{k+1}} M_{\text{pr}}(\eta)dv_{\text{pt}}(\eta) = \lim_{\Delta\zeta\to 0}\sum_{i=1}^{T-1} M_{\text{pr}}(\zeta_i)|M_{\text{pt}}(\zeta_{i+1}) - M_{\text{pt}}(\zeta_i)|,$$

where the limit is over all partitions $\eta_k = \zeta_1 < \ldots < \zeta_T = \eta_{k+1}$ of $[\eta_k, \eta_{k+1}]$, with the mesh size $\Delta\zeta := \max\zeta_{i+1} - \zeta_i$ tending to zero. The interior points do not contribute to the right hand side sum.

If $\eta_k \in H'$, $\eta_{k+1} \notin H'$, thus the last summand also vanishes. Since $M_{\text{pt}}$ is locally constant away from $H'$, we deduce that

$$\begin{aligned}\int_{\eta_k}^{\eta_{k+1}} M_{\text{pr}}(\eta)dv_{\text{pt}}(\eta) &= M_{\text{pr}}(\eta_k)|M_{\text{pt}}(\zeta_2) - M_{\text{pt}}(\zeta_1)|\\ &= M_{\text{pr}}(\eta_k)|M_{\text{pt}}(\eta_{k+1}) - M_{\text{pt}}(\eta_k)|.\end{aligned}$$

Similarly if $\eta_{k+1} \in H'$, only the last summand survives, and we have

$$\int_{\eta_k}^{\eta_{k+1}} M_{\mathrm{pr}}(\eta) dv_{\mathrm{pt}}(\eta) = M_{\mathrm{pr}}(\zeta_{T-1})|M_{\mathrm{pt}}(\eta_{k+1}) - M_{\mathrm{pt}}(\zeta_{T-1})| = M_{\mathrm{pr}}(\eta_k)|M_{\mathrm{pt}}(\eta_{k+1}) - M_{\mathrm{pt}}(\eta_k)|.$$

Thus we have shown that under any refinement $\zeta \vdash [\eta_k, \eta_{k+1}]$,

$$\sum_{i=1}^{T-1} M_{\mathrm{pr}}(\zeta_i)|M_{\mathrm{pt}}(\zeta_{i+1}) - M_{\mathrm{pt}}(\zeta_i)| = M_{\mathrm{pr}}(\eta_k)|M_{\mathrm{pt}}(\eta_{k+1}) - M_{\mathrm{pt}}(\eta_k)|.$$

The conclusion of the Theorem follows by unravelling the definition of BML-AUC. □

In practice, we found that the number of steps required in Algorithm 2 can grow quadratically with the mini-batch size. So we choose the anchor points $H$ to evenly divide the unit interval $[0, 1]$. The final BML-AUC results are highly stable with respect to $H$, provided it is large enough (usually $|H| \geq 10$ is sufficient).

## 3.2 HMMoE: Heterogeneous Multi-gate Mixture of Experts

We propose a new MoE model that is compatible with the heterogeneous tasks and has strong model interpretability. We call this model HMMoE. There are four module structures in HMMoE: interaction match module, $n$ expert networks, $t$ tower networks, and $t$ gating networks (i.e., $t$ tasks).

**Interaction match module.** For two text sequences $A$ and $B$, the word-level interaction match calculates the dot product between their embeddings and then reshape the result into a one-dimensional vector:

$$Match(\{\boldsymbol{E}_A^i\}_{i=1}^{l_A}, \{\boldsymbol{E}_B^i\}_{i=1}^{l_B}) = reshape([\boldsymbol{E}_A^1, ..., \boldsymbol{E}_A^{l_A}]^{\mathrm{T}} \times [\boldsymbol{E}_B^1, ..., \boldsymbol{E}_B^{l_B}]) \tag{14}$$

where $l_A$ is the length of the sequence A, and $\boldsymbol{E}_A^i$ is the $i$-th word's embedding in $A$.

For the pointwise-data task, we obtain its expert networks' input $\boldsymbol{E}_{\mathrm{pt}}$ by concatenating query $q$ and title $t$'s interaction match result, $q$'s sequence embedding $\boldsymbol{E}_q$ and $t$'s sequence embedding $\boldsymbol{E}_t$:

$$\boldsymbol{E}_{\mathrm{pt}} = [Match(\{\boldsymbol{E}_q^i\}_{i=1}^{l_q}, \{\boldsymbol{E}_t^i\}_{i=1}^{l_t}) \parallel \boldsymbol{E}_q \parallel \boldsymbol{E}_t]. \tag{15}$$

Similarly, for the pairwise-data task, we calculate and obtain its positive example's and negative example's results as the expert network's inputs:

$$\boldsymbol{E}_{\mathrm{pr}}^{pos} = [Match(\{\boldsymbol{E}_q^i\}_{i=1}^{l_q}, \{\boldsymbol{E}_{t_{pos}}^i\}_{i=1}^{l_{tpos}}) \parallel \boldsymbol{E}_q \parallel \boldsymbol{E}_{t_{pos}}] \tag{16}$$

$$\boldsymbol{E}_{\mathrm{pr}}^{neg} = [Match(\{\boldsymbol{E}_q^i\}_{i=1}^{l_q}, \{\boldsymbol{E}_{t_{neg}}^i\}_{i=1}^{l_{tneg}}) \parallel \boldsymbol{E}_q \parallel \boldsymbol{E}_{t_{neg}}]. \tag{17}$$

**Expert networks.** To make different groups of experts to capture the influence of different word patterns, we set $N_p$ groups of experts where each group has $N_e$ experts. The forward pass process of the expert networks can be represented as:

$$E_i(x) = \mathrm{ReLU}(Expert_i(x)), i = 1, \cdots, N_p \cdot N_e \tag{18}$$

where $x \in (\boldsymbol{E}_{\mathrm{pt}}, \boldsymbol{E}_{\mathrm{pr}}^{pos}, \boldsymbol{E}_{\mathrm{pr}}^{neg})$. $Expert_i(\cdot)$ can be designed as any neural networks, here we use multi-layer perceptron to replace it.

**Gating networks.** Considering the importance of relevance learning and preference learning, we integrate the corresponding pointwise-data task and pairwise-data task into HMMoE's learning objectives. Suppose the $i$-th expert's output is $E_i(x)$, then the input of the poinwise-data task tower is decided by the output of the pointwise-task gating $G_{\mathrm{pt}}(Map(q))$:

$$\boldsymbol{H}_{\mathrm{pt}} = \sum_{i=1}^{N_p \cdot N_e} G_{\mathrm{pt}}(Map(q))_i E_i(\boldsymbol{E}_{\mathrm{pt}}) \tag{19}$$

Similarly, the pairwise-data task tower's input can be derived by the output of the pairwise-task gating $G_{\mathrm{pr}}(Map(q))$:

$$\boldsymbol{H}_{\mathrm{pr}}^{pos} = \sum_{i=1}^{N_p \cdot N_e} G_{\mathrm{pr}}(Map(q))_i E_i(\boldsymbol{E}_{\mathrm{pr}}^{pos}) \tag{20}$$

$$\boldsymbol{H}_{\mathrm{pr}}^{neg} = \sum_{i=1}^{N_p \cdot N_e} G_{\mathrm{pr}}(Map(q))_i E_i(\boldsymbol{E}_{\mathrm{pr}}^{neg}). \tag{21}$$

To distinguish different query patterns and thus trigger the homologous experts, we use the one-hot mapping result of the original query as the input of each gating network. Specifically, we sequentially set four different word patterns: product word, attribute word, brand word and type word. When any pattern exists in the query, the corresponding position's element is triggered as 1, otherwise is 0. In the in-house statistical result, these four word patterns take up for about 98.7% of all user queries.

We design three types of gating network including implicit, semi-explicit and explicit gating network (see Figure 3). Their difference is the balance degree between automatic parameter learning and frozen parameter setting. Implicit gating network has fully automatic parameter learning, while explicit gating network has fully frozen parameter setting and semi-explicit version is in the middle. The final HMMoE model uses semi-explicit gating network setting. We further compare the above three settings in the experiments.

**Tower networks.** Two tower networks are used to absorb the outputs of the gating networks and expert network, and then transform these outputs into the final logit value of. So the logit values of pointwise-task ($s_i$), pairwise-task ($\widetilde{s_{i_1}}$ and $\widetilde{s_{i_2}}$) on the $i$-th example are defined as:

$$\widetilde{s_i} = Tower_{\mathrm{pt}}(\boldsymbol{H}_{\mathrm{pt}}), \widetilde{s_{i_1}} = Tower_{\mathrm{pr}}(\boldsymbol{H}_{\mathrm{pr}}^{pos}), \widetilde{s_{i_2}} = Tower_{\mathrm{pr}}(\boldsymbol{H}_{\mathrm{pr}}^{neg}). \tag{22}$$

The whole model framework can be seen in Figure 2(b).

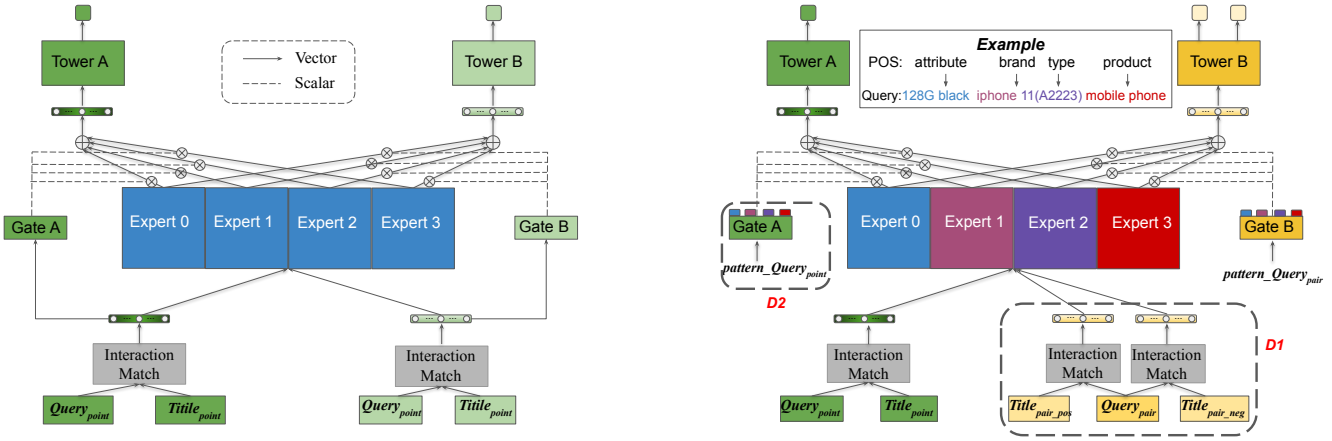
## 3.3 Model learning

Here we highlight the design on the heterogeneous task learning, especially the pairwise-data task. The loss of the pointwise task is defined as:

$$L_{\mathrm{pt}} := -\sum_{i=1}^{N} \left[ s_i \log(\frac{1}{1+e^{-\widetilde{s_i}}}) + (1-s_i)\log(1 - \frac{1}{1+e^{-\widetilde{s_i}}}) \right] \tag{23}$$

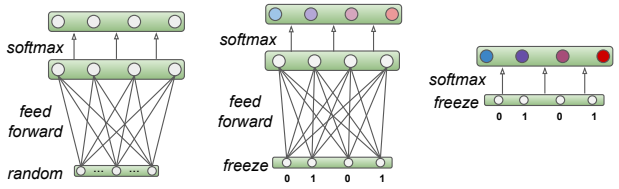
where $N$ is the size of each batch and $s_i$, $\widetilde{s_i}$ is the groundtruth, prediction value of the $i$-th example. The loss of the pairwise task is defined as:

$$L_{\mathrm{pr}} := -\sum_{i=1}^{N} \left[ s_{<i_1,i_2>} \log \widetilde{s}_{<\widetilde{i_1},\widetilde{i_2}>} + (1 - s_{<i_1,i_2>}) \log(1 - \widetilde{s}_{<\widetilde{i_1},\widetilde{i_2}>}) \right] \tag{24}$$




(a) MMoE (b) HMMoE

**Figure 2: The comparison between the original MMoE and our proposed HMMoE. HMMoE has two main designs: heterogeneous tasks learning (D1) and strong interpretability (D2).**




(a) Implicit gating network (b) Semi-explicit gating network (c) Explicit gating network

**Figure 3: Different types of gating network setting**

where

$$s_{<i_1,i_2>} = \begin{cases} 1, & s_{i_1} > s_{i_2}; \\ 0, & s_{i_1} \leqslant s_{i_2}. \end{cases} \tag{25}$$

To avoid the value overflow likely caused by pairwise data, we set

$$\widetilde{s}_{<\widetilde{i_1},\widetilde{i_2}>} = min(max(\frac{1}{1+e^{\widetilde{s_{i_2}}-\widetilde{s_{i_1}}}}, \varepsilon), 1-\varepsilon) \tag{26}$$

where $\varepsilon$ is a very small value.

## 4 EXPERIMENTS

We demonstrate the effectiveness of HMMOE on both public Amazon Review dataset and in-house dataset. We will first describe the metrics, experiment settings and how we generate the dataset. And then we will analyze the contribution of different component/setup in our method, to demonstrate the effectiveness of the model design. Finally, we will have a comprehensive experiment comparison with other neural network ranking models, to illustrate the model performance.

### 4.1 Experiment Setup

*4.1.1 Baseline Method.* Besides different setups of HMMOE models such as single task MOE, explicit/implicit Part-of-Speech pattern, we compare our method with state-of-the-art deep neural network models for text pair semantic match problem, such as DSSM[19], ARC-I [2] and ARC-II[2].

*4.1.2 Evaluation Metrics and Parameter Settings.* We use four evaluation methods to evaluate model in our experiment, including ACC, AUC, BML-AUC and SUM(F1). We evaluated our model on both in-house relevance and feedback data as well as a public data set on Amazon Product Review data.

In our experiments, both MMoE models have 8 experts (i.e., $n = 8$) with a single layer network, in which the size of the hidden layers is 16. Similarly, the tower networks for each task are also 16 hidden units. The vocabulary set is about 40k, it is generated from the unigrams and bigrams, the size of dimension is 64. The batch-size is 1024 and learning rate is 1e-3. For each query and item's title, we set a limited sequence length, the query is 10 and item's title is 65.

**Table 1: Results on Amazon dataset of HMMoE. 'Sent' is single pointwise task learning, 'Rate' is single pairwise task learning and 'Sent+Rate' is multi-task learning.**

| Metrics | Sent | Rate | Sent + Rate |
|---|---|---|---|
| $\text{Sent}_{\text{ACC}}$ | 0.7687 | 0.2328 | **0.7690** |
| $\text{Sent}_{\text{AUC}}$ | 0.6229 | 0.5004 | **0.6326** |
| $\text{Rate}_{\text{ACC}}$ | 0.4607 | 0.5950 | **0.6172** |
| $\text{Rate}_{\text{AUC}}$ | 0.4447 | 0.6345 | **0.6630** |
| BML-AUC | 0.5259 | 0.5390 | **0.6288** |
| SUM(F1) | 0.5089 | 0.5296 | **0.6239** |

### 4.2 Amazon Review Data Experiment

*4.2.1 Data Generation.* Although the task with heterogeneous data is common in real world applications, it's hard to find an available public data. Here we construct a heterogeneous dataset from Amazon Review Dataset, with task of review sentiment analysis and

pairwise rating. For Amazon Product Review Data, it mainly include product id, text of the review, name of the product, and rating of the product. To construct dataset containing tasks of both pointwise and pairwise data, we construct pairwise data under the same reviewer, by gathering reviewed products of different ratings. We construct pointwise data by feeding the review text into a BERT fine-tuned sentiment analysis model. The fine-tuning is performed on a holdout shard (according reviewer id), so that the rest of the experiments does not use this slice of data. Eventually we generate about 30 million training data. For pointwise data, the task is to predict the sentimental polarity or ratings from review, while for pairwise data the task is to point out which item the reviewer preferred.

*4.2.2 Experiment Results.* Table 1 presents the ACC and AUC on each task. According to the results, tasks with a single target (sentiment or rating score) receive the good result on metrics from related task labels, but cannot perform well on the metrics of the other target. However by using multiple tasks labels, as shown in task of sentiment analysis and rating together, metrics on both task exceed the single task results.

## 4.3 In-house E-commerce Data Experiment

*4.3.1 Data Generation.* To demonstrate the efficacy of our method in a real industrial setting, we experiment on an in-house dataset. We utilize historical 6-month users' search log, constructed two types of heterogeneous search data with the same data size.

**Relevance Distillation Data.** Since raw relevance labels are much more labor-intensive to collect, we take a distillation approach, relying on the state of the art NLP models such as BERT to produce high quality relevance predictions as labels. We collect 170m (query, item) pairs, filtered by the criterion of being clicked at least once or skipped at least 5 times. The evaluation data uses human labeled (query, item) pairs with a relevance label.

**User Feedback Data.** According to the user's order feedback data, we constructed the pairwise data of different ordered items under the same query. In order to avoid cross-leakage, the pairwise data is randomly shuffled by query divided into 90% training and 10% test set.

- It contains about 170m instances of the 5-tuple's ($query$, $item_a$, $item_b$, $order\_cnt_a$, $order\_cnt_b$) pairs, where one of the item's $order\_cnt$ is 0, and $|order\_cnt_a - order\_cnt_b| \geq 5$.
- Similarly, we generate 170m ($query$, $item$, $is\_ordered$) 3-tuples, where the item's $is\_ordered$ is 1 or 0.

*4.3.2 Experiment Setups.* To test whether valid this work's motivation is, we design two parts of experiment settings. First, we want to observe **whether partially ordered user feedback is better than point-to-point user feedback**. Second, we want to observe **the effects of different types of gating networks**.

*4.3.3 Experiment Results.* Table 2 shows the results of different multi-expert models on the in-house data. Under the new metric of BML-AUC and SUM(F1), HMMoE model with semi-explicit gating network achieves the best performance compared with the other models. MoE and MMoE model have very close performance with HMMoE model under the relevance metric, but MoE cannot solve preference learning task and the result of MMoE is far less than HMMoE's under the metric of preference learning.

**Table 2: Comparison of multi-expert models on the in-house data. '$\text{Rel}_{\text{ACC}}$' and '$\text{Rel}_{\text{AUC}}$' are the accuracy metric and AUC metric under the relevance learning task using relevance annotation data, '$\text{Ord}_{\text{ACC}}$' and '$\text{Ord}_{\text{AUC}}$' are the accuracy metric and AUC metric under the preference learning task using user order data. HMMoE$_i$, HMMoE$_e$ and HMMoE applies implicit, explicit and semi-explicit gating network setting respectively.**

| Types | $\text{Rel}_{\text{ACC}}$ | $\text{Rel}_{\text{AUC}}$ | $\text{Ord}_{\text{ACC}}$ | $\text{Ord}_{\text{AUC}}$ | BML − AUC | SUM(F1) |
|---|---|---|---|---|---|---|
| MoE | 0.8360 | 0.8538 | - | - | - | - |
| MMoE | 0.8316 | 0.8502 | 0.8147 | 0.8973 | - | - |
| HMMoE$_i$ | 0.8328 | 0.8517 | 0.8874 | 0.9575 | 0.8502 | 0.6940 |
| HMMoE$_e$ | 0.8300 | 0.8415 | 0.8755 | 0.9488 | 0.8435 | 0.7000 |
| HMMoE | **0.8379** | **0.8550** | **0.8899** | **0.9584** | **0.8530** | **0.7092** |

## 4.4 Comparison with Deep Text Match Models

Here we compare HMMoE along with semi-explicit gating network setting with some deep text match models on Amazon and in-house datasets. We select three representation-based matching models (DSSM, MVLSTM and ARC-I) and three interaction-based matching models (ARC-II, KNRM and MatchPyramid) from a high-quality codebase named as MatchZoo [30]. These results are shown in Table 3. In general, HMMoE outperforms these baseline models apart from the relevance accuracy metric on Amazon dataset. The main reason is that multi-task learning can reduce some negative effects of the noise in each single task.

**Table 3: Comparison with deep text match model under Amazon dataset and in-house dataset.**

| Models | Amazon dataset | | In-house dataset | |
|---|---|---|---|---|
| | $\text{Sent}_{\text{ACC}}$ | $\text{Sent}_{\text{AUC}}$ | $\text{Rel}_{\text{ACC}}$ | $\text{Rel}_{\text{AUC}}$ |
| DSSM | 0.6329 | 0.5030 | 0.7686 | 0.8219 |
| MVLSTM | **0.7703** | 0.6217 | 0.8052 | 0.7877 |
| ARC-I | 0.7688 | 0.6215 | 0.8294 | 0.8312 |
| ARC-II | 0.7682 | 0.6216 | 0.8165 | 0.8071 |
| KNRM | 0.7673 | 0.5895 | 0.8002 | 0.7768 |
| MatchPyramid | 0.7676 | 0.5970 | 0.8052 | 0.8093 |
| HMMoE | 0.7687 | **0.6229** | **0.8379** | **0.8550** |

## 4.5 Online A/B Testing

To demonstrate the effectiveness of the HMMoE model in the real-world e-commerce environment, we push HMMoE online and observe continuous 14 days A/B testing results (under default sort) on three different online channels of JD.com, including $\text{JD}_{\text{APP}}$, $\text{JX}_{\text{APP}}$ and $\text{JS}_{\text{APP}}$. Because online environment uses point-to-point relevance estimation, we retain its pointwise-part network and discard its pairwise-part network when applying HMMoE online.

The online statistical results are shown in Table 4. UV (unique visitor)-value and UCVR (unique visitor click conversion rate) are two chosen metrics, which measures the gross merchandise value

generated per user session and the rate of conversion per user session respectively. We can conclude that HMMoE achieves significant or nearly significant gain compared to the online baseline DNN model on all channels. The reason behind such result is HMMoE's fine-grained query patterns processing mechanism and mutual gain from relevance learning and preference learning.

**Table 4: Online A/B testing experiment results.**

| Metrics | $JD_{APP}$ | $JX_{APP}$ | $JS_{APP}$ |
|---|---|---|---|
| UV-value | +0.93% | +3.26% | +2.71% |
| P-value | 1.25e-2 | 7.30e-2 | 2.12e-1 |
| UCVR | +0.3384% | +1.45% | +1.40% |
| P-value | 2.56e-2 | 2.19e-2 | 5.04e-3 |

## 5 CONCLUSION

In this paper, we study an important e-commerce search problem, that is, the general mismatch between the semantic relevance and user preference. To tackle this problem, we redesign a novel and reasonable evaluation metric by theoretical analysis, in order to ensure the balance between the relevance learning and preference learning. The Mixture-gate Mixture of Experts (MMoE) framework is suitable to such a metric, but still preserves its inherent limitations: cannot deal with heterogeneous tasks and weak interpretability. So we reform MMoE into heterogeneous MMoE (HMMoE). In a specific application scene, HMMoE can assign the stationary experts to participate in the following calculation according to explicit or semi-explicit gating networks. Finally, we verify HMMoE's effectiveness compared to base MMoE and other text match models in the offline experiments, and then show HMMoE's positive benefits in the online e-commerce system.

## REFERENCES


[1] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.* AAAI Press, 2016, pp. 2793–2799.

[2] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2042–2050.

[3] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017.* ACM, 2017, pp. 1291–1299.

[4] Y. Jiang, Y. Shang, Z. Liu, H. Shen, Y. Xiao, W. Xiong, S. Xu, W. Yan, and D. Jin, "Bert2dnn: Bert distillation with massive unlabeled data for online e-commerce search," *Proceedings of the 20th IEEE International Conference on Data Mining, ICDM 2020*, 2020.

[5] W. Zhang, W. Bao, X. Liu, K. Yang, Q. Lin, H. Wen, and R. Ramezani, "Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.* ACM / IW3C2, 2020, pp. 2775–2781.

[6] H. Zhu, J. Jin, C. Tan, F. Pan, Y. Zeng, H. Li, and K. Gai, "Optimized cost per click in taobao display advertising," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017.* ACM, 2017, pp. 2191–2200.

[7] J. Mao, Y. Liu, K. Zhou, J. Nie, J. Song, M. Zhang, S. Ma, J. Sun, and H. Luo, "When does relevance mean usefulness and user satisfaction in web search?" in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016.* ACM, 2016, pp. 463–472.

[8] A. Moffat, P. Thomas, and F. Scholer, "Users versus models: what observation tells us about effectiveness metrics," in *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013.* ACM, 2013, pp. 659–668.

[9] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009.* ACM, 2009, pp. 621–630.

[10] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[11] D. Das, H. Massa, A. Kulkarni, and T. Rekatsinas, "An empirical analysis of the impact of data augmentation on knowledge distillation," *arXiv preprint arXiv:2006.03810*, 2020.

[12] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[13] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018.* ACM, 2018, pp. 1930–1939.

[14] P. Sirotkin, "On search engine evaluation metrics," *arXiv preprint arXiv:1302.2318*, 2013.

[15] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[16] S. J. Mason and N. E. Graham, "Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation," *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, vol. 128, no. 584, pp. 2145–2166, 2002.

[17] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[18] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks," *Journal of Artificial Intelligence Research*, vol. 42, pp. 689–718, 2011.

[19] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2333–2338.

[20] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.* AAAI Press, 2016, pp. 2835–2841.

[21] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* ACM, 2016, pp. 55–64.

[22] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017.* ACM, 2017, pp. 55–64.

[23] Z. Dai, C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for soft-matching n-grams in ad-hoc search," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.* ACM, 2018, pp. 126–134.

[24] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers.* Association for Computational Linguistics, 2017, pp. 1657–1668.

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling BERT for natural language understanding," *CoRR*, vol. abs/1909.10351, 2019.

[26] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019.

[27] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.

[28] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019, pp. 13–23.

[29] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[30] J. Guo, Y. Fan, X. Ji, and X. Cheng, "Matchzoo: A learning, practicing, and developing system for neural text matching," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.* ACM, 2019, pp. 1297–1300.