Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Sister Conferences Best Papers Track

GS^{2}P: A Generative Pre-trained Learning to Rank Model with Over-parameterization for Web-Scale Search (Extended Abstract)*

Yuchen Li[1,2] , **Haoyi Xiong**[2] , **Linghe Kong**[1] , **Jiang Bian**[2] , **Shuaiqiang Wang**[2] , **Guihai Chen**[1] and **Dawei Yin**[2]
[1]Shanghai Jiao Tong University, China [2]Baidu Inc., China
{yuchenli, linghe.kong, gchen}@sjtu.edu.cn, haoyi.xiong.fr@ieee.org,
{jiangbian03, shqiang.wang}@gmail.com, yindawei@acm.org

Abstract

While *learning to rank* (LTR) is widely employed in web searches to prioritize pertinent webpages from the retrieved contents based on input queries, traditional LTR models stumble over two principal stumbling blocks leading to subpar performance: 1) the lack of well-annotated query-webpage pairs with ranking scores to cover search queries of various popularity, debilitating their coverage of search queries across the popularity spectrum, and 2) ill-trained models that are incapable of inducing generalized representations for LTR, culminating in overfitting. To tackle above challenges, we proposed a *Generative Semi-Supervised Pre-trained* (GS^{2}P) LTR model. We conduct extensive offline experiments on a publicly available dataset and a real-world dataset collected from a large-scale search engine. We also deploy GS^{2}P at a large-scale web search engine with realistic traffic, where we can observe significant improvement in real-world applications.

1 Introduction

The booming increase of internet users and web content surges the demands on web search. In the current digital epoch, large-scale search engines manage an impressive archive of trillions of webpages, providing service to hundreds of millions of active users daily while handling billions of queries. The search procedure commences with a user query, often a text string, necessitating keyword or phrase extraction to comprehend user attempting [Zhao *et al.*, 2010; Li *et al.*, 2023d]. Post identification of keywords, search engines evaluate the relation between the query and webpages, subsequently retrieving highly relevant ones from their vast databases [Liu *et al.*, 2021]. These webpages are then sorted based on content attributes and click-through rates, positioning the most relevant ones on top of the result [Li *et al.*, 2023a].

The optimization of the user experience, achieved by catering to information needs, largely depends on the effective sorting of retrieved content. In this realm, Learning to Rank (LTR) becomes instrumental, requiring a considerable amount of query-webpage pairings with relevancy scores for effective supervised LTR [Li *et al.*, 2023b; Qin and Liu, 2013; Li *et al.*, 2023c]. Nevertheless, the commonplace scarcity of well-described, query-webpage pairings often compels semi-supervised LTR, harnessing both labeled and unlabeled samples for the process [Szummer and Yilmaz, 2011; Zhang *et al.*, 2016]. Recent years have seen the integration of deep models in LTR, aimed at end-to-end ranking loss minimization [Li *et al.*, 2020; Wang *et al.*, 2021; Li *et al.*, 2022; Yang and Ying, 2023]. However, these models occasionally falter in learning generalizable representations from structural data due to limited or noisy supervision, sometimes resulting in performance that is weaker compared to statistical learners [Bruch *et al.*, 2019]. Further discussion on this subject can be found in a comprehensive review available in a recent scholarly work [Werner, 2022].

In order to tackle the above issues, we propose Generative Semi-Supervised Pre-trained LTR (GS^{2}P) model. The proposed GS^{2}P first generates high-quality pseudo labels for every unlabeled query-webpage pair through co-training of multiple/diverse LTR models based on various ranking losses, then learns generalizable representations with a self-attentive network using both generative loss and discriminative loss. Finally, given the generalizable representations of query-webpage pairs, by incorporating an MLP-based ranker with Random Fourier Features (RFF), GS^{2}P pushes LTR models into so-called interpolating regime [Belkin, 2021] and obtains superb performance improvement. To demonstrate the effectiveness of GS^{2}P, we conduct comprehensive experiments on a publicly available LTR dataset [Qin and Liu, 2013] and a real-world dataset collected from a large-scale search engine. We also deploy GS^{2}P at the search engine and evaluate the proposed model using online A/B tests in comparison with the online legacy system.

2 Methodology

2.1 Preliminaries

Given a set of search queries $\mathcal{Q} = \{q_1,\ q_2,\ \ldots\}$ and all archived webpages $\mathcal{W} = \{w_1, w_2, \ldots\}$, for each query $q_i \in \mathcal{Q}$, the search engine retrieves a set of relevant webpages denoted as $W_i = \{w_1^i, w_2^i, \ldots\} \subset \mathcal{W}$. After annotating,

*This work was initially presented at the 10th IEEE International Conference on Data Science and Advanced Analytics (DSAA) in 2023 and Machine Learning (MLJ) in 2024.

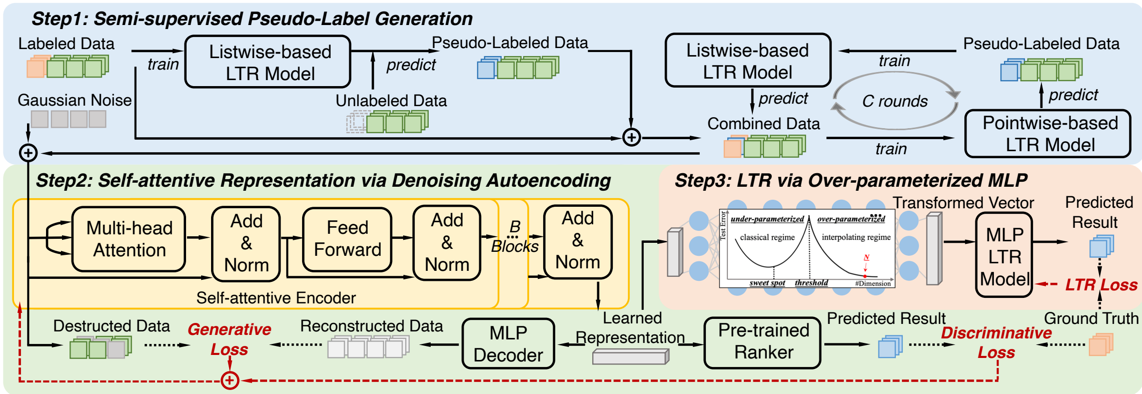

Figure 1: The framework of GS^{2}P.

each query q_i is assigned with a set of relevance scores $\boldsymbol{y}_i = \{y_1^i, y_2^i, \dots\}$. In this work, we follow the settings in [Qin and Liu, 2013; Li *et al.*, 2023e] and scale the relevance score from 0 to 4 to represent levels of relevance, which represents whether the webpage w.r.t. the query is bad (0), fair (1), good (2), excellent (3) or perfect (4). We denote a set of query-webpage pairs with relevance score annotations as $\mathcal{T}^L = \{(q_1, W_1, \boldsymbol{y}_1),\ (q_2, W_2, \boldsymbol{y}_2), \dots\}$. The core problem of semi-supervised LTR is to leverage unlabeled pairs, i.e., $\mathcal{T}^U = \{(q_1', W_1'),\ (q_2', W_2'), \dots\} \subset \mathcal{Q}$ and $|\mathcal{T}^U| \gg |\mathcal{T}^L|$, in the training process.

2.2 Semi-supervised Pseudo-Label Generation

Given the overall set of queries $\mathcal{Q}$ and the set of all webpages $\mathcal{W}$, GS^{2}P first obtains every possible query-webpage pair from both datasets, denoted as (q_i, w_i^j) for $\forall q_i \in \mathcal{Q}$ and $\forall w_i^j \in W_i \subset \mathcal{W}$, i.e., the j^{th} webpage retrieved for the i^{th} query. For each query-webpage pair (q_i, w_i^j), GS^{2}P further extracts an m-dimensional feature vector $\boldsymbol{x}_{i,j}$ representing the features of the j^{th} webpage under the i^{th} query. Then, the labeled and unlabeled sets of feature vectors can be presented as $\mathcal{D}^L = \{(\boldsymbol{x}_{i,j}, \boldsymbol{y}_j^i) | \forall (q_i, W_i, \boldsymbol{y}) \in \mathcal{T}^L$ and $\forall w_i^j \in W_i\}$ and $\mathcal{D}^U = \{\boldsymbol{x}_{i,j} | \forall (q_i, W_i) \in \mathcal{T}^U\}$. Inspired by [Li *et al.*, 2023e], GS^{2}P leverages a semi-supervised learning LTR manner to generate high-quality pseudo labels for unlabeled samples.

2.3 Self-attentive Representation Learning via Denoising Autoencoding

Denoised Self-attentive Autoencoder. Given an m-dimensional feature vector $\tilde{\boldsymbol{x}}_{i,j}$ of a query-webpage pair $(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{y}_j^i)$ in combined data, GS^{2}P aims to utilize a self-attentive encoder to learn a generalizable representation $\boldsymbol{z}_{i,j}$. Specifically, given a vector $\tilde{\boldsymbol{x}}_{i,j}$ generated from *Semi-supervised Pseudo-Label Generation*, GS^{2}P (1) passes it through a fully-connected layer and produces a hidden representation. Then, GS^{2}P (2) feeds the hidden representation into a self-attentive autoencoder, which consists of B encoder blocks of Transformer [Vaswani *et al.*, 2017]. In particular, each encoder block incorporates a multi-head attention layer and a feed-forward layer, both followed by layer normalization. Eventually, GS^{2}P (3) generates the learned representation $\boldsymbol{z}_{i,j}$ from the last encoder block. For each original feature vector $\tilde{\boldsymbol{x}}_{i,j}$, the whole training process can be formulated as $\boldsymbol{z}_{i,j} = f_{\tilde{\theta}}(\tilde{\boldsymbol{x}}_{i,j})$, where $\tilde{\theta}$ is the set of parameters of the self-attentive encoder.

Given the learned representation $\boldsymbol{z}_{i,j}$, GS^{2}P leverages an MLP-based decoder for the reconstruction task. Specifically, for each representation $\boldsymbol{z}_{i,j}$ produced from the self-attentive autoencoder, GS^{2}P uses the MLP-based decoder to map $\boldsymbol{z}_{i,j}$ to a generalizable representation $\boldsymbol{z}_{i,j}'$, which has the same dimension with the original feature vector. The whole training process can be formulated as $\boldsymbol{z}_{i,j}' = g_{\theta'}(\boldsymbol{z}_{i,j})$, where the θ' is the set of parameters of the MLP-based decoder. Finally, GS^{2}P jointly optimizes the parameter sets $\tilde{\theta}$ and θ' to minimize the generative loss as $\mathcal{L}_{\mathrm{G}} = \frac{1}{|\mathcal{Q}|}\frac{1}{|W_i|}\sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|W_i|} \ell_{\mathrm{G}}\left(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{z}_{i,j}'\right)$, where ℓ_{G} is the squared error, which could be presented as $\ell_{\mathrm{G}}\left(\tilde{\boldsymbol{x}}_{i,j}, \boldsymbol{z}_{i,j}'\right) = \|\tilde{\boldsymbol{x}}_{i,j} - \boldsymbol{z}_{i,j}'\|^2$.

Pre-trained Ranker. Given the learned vector $\boldsymbol{z}_{i,j}$ generated from *Denoised Self-attentive Autoencoder*, GS^{2}P leverages a fully-connected layer to obtain predicted scores $\boldsymbol{r}_{i,j}$ as $\boldsymbol{r}_{i,j} = k_\theta(\boldsymbol{z}_{i,j})$, where θ is the set of discriminative parameters of *Pre-trained Ranker*. Against the ground truth, GS^{2}P utilizes the discriminative loss function $\mathcal{L}_{\mathrm{D}}$ to compute the loss of ranking prediction as $\mathcal{L}_{\mathrm{D}} = \frac{1}{|\mathcal{Q}|}\frac{1}{|W_i|}\sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|W_i|} \ell_{\mathrm{D}}\left(\boldsymbol{y}_j^i, \boldsymbol{r}_{i,j}\right)$, where ℓ_{D} is denoted as the standard LTR loss function. Then, GS^{2}P jointly optimizes the discriminative loss $\mathcal{L}_{\mathrm{D}}$ and the generative loss $\mathcal{L}_{\mathrm{G}}$ to accomplish both discriminative (LTR) and generative (denoising autoencoding for reconstruction) tasks simultaneously as $\mathcal{L}_{\mathrm{Final}} = \alpha\mathcal{L}_{\mathrm{D}} + \beta\mathcal{L}_{\mathrm{G}}$, where $\alpha, \beta \in [0, 1]$ are weight coefficients to balance two terms.

2.4 LTR via Over-parameterized MLP

Given the learned representation $\boldsymbol{z}_{i,j} \in \mathcal{R}^n$ generated from *Self-attentive Representation Learning via Denoising Autoencoding*, GS^{2}P converts this representation vector into an N-dimensional version, represented as $\boldsymbol{h}_{i,j} = \mathbf{h}(\boldsymbol{z}_{i,j})$. This step is implemented using the feature transformation $\mathbf{h}(z)$. In this procedure, GS^{2}P utilizes a transformation rooted in random Fourier features to execute $\mathbf{h}(z)$ [Rahimi and Recht, 2007], thereby mapping the original features of LTR into a higher dimensional feature space. An important point to con-

Methods	5%		10%		15%		20%	
	@4	@10	@4	@10	@4	@10	@4	@10
XGBoost	31.76	34.10	36.72	39.12	39.93	41.01	42.60	45.84
LightGBM	35.72	39.32	39.89	42.05	43.90	45.67	46.56	48.52
RMSE	34.82	38.02	38.75	41.95	42.97	45.65	45.75	48.86
RankNet	34.06	37.43	38.12	41.32	42.24	45.08	45.01	47.89
LambdaRank	35.28	38.50	39.32	42.47	43.40	46.23	46.26	49.56
ListNet	34.36	37.94	38.31	41.76	42.51	45.40	45.32	48.42
ListMLE	33.47	36.95	37.52	40.84	41.53	44.43	44.39	47.26
ApproxNDCG	33.98	37.20	37.94	41.01	42.09	44.70	44.94	47.50
NeuralNDCG	35.15	38.26	39.07	42.10	43.32	45.97	46.08	49.20
$\mathrm{CR}_{\mathrm{RMSE}}$	36.04	38.54	39.52	42.48	43.67	46.25	46.86	49.75
$\mathrm{CR}_{\mathrm{RankNet}}$	35.90	38.42	39.44	42.37	43.45	45.98	46.70	49.61
$\mathrm{CR}_{\mathrm{LambdaRank}}$	36.45	38.93	40.03	43.10	44.36	46.88	47.57	50.47
$\mathrm{CR}_{\mathrm{ListNet}}$	37.53	40.08	41.28	44.21	45.17	47.73	48.35	51.24
$\mathrm{CR}_{\mathrm{ListMLE}}$	35.67	38.16	39.40	42.35	43.28	45.86	46.62	49.48
$\mathrm{CR}_{\mathrm{ApproxNDCG}}$	37.93	40.41	41.47	44.32	45.53	48.03	48.81	51.69
$\mathrm{CR}_{\mathrm{NeuralNDCG}}$	37.26	40.65	40.76	43.69	44.85	47.52	48.16	51.13
$\mathrm{GS^2P}_{\mathrm{RMSE}}$	39.02	40.88	41.80	44.72	45.72	48.22	48.72	51.40
$\mathrm{GS^2P}_{\mathrm{RankNet}}$	38.15	40.42	40.03	44.21	44.93	47.85	47.85	50.98
$\mathrm{GS^2P}_{\mathrm{LambdaRank}}$	39.47	41.43	42.17	45.20	46.07	48.89	49.15	51.97
$\mathrm{GS^2P}_{\mathrm{ListNet}}$	39.53	41.62	42.28	45.42	46.15	49.16	49.18	52.20
$\mathrm{GS^2P}_{\mathrm{ListMLE}}$	37.66	39.87	39.80	43.70	44.52	47.28	47.41	50.24
$\mathrm{GS^2P}_{\mathrm{ApproxNDCG}}$	39.57	41.76	42.39	45.65	46.31	49.31	49.25	52.25
$\mathrm{GS^2P}_{\mathrm{NeuralNDCG}}$	**39.72**	**41.97**	**42.56**	**45.83**	**46.38**	**49.53**	**49.36**	**52.47**

Table 1: Results for Web30K on NDCG across diverse labeled data percentages.

sider is that increasing the number of dimensions (N) leads to over-parameterization of the LTR model via the addition of more input features. This scenario brings about a feature-wise 'double descent' phenomenon in predicting generalization errors [Belkin *et al.*, 2019; Belkin, 2021]. GS^{2}P sets the optimal value for N, stemming from cross-validation performed on the labeled dataset to ensure the best generalization performance. Therefore, incorporating $\boldsymbol{h}_{i,j}$ for every pair of query-webpage paves the path for an over-parameterized LTR model. This advanced model operates in the interpolating regime and is projected to exhibit excellent generalization performance [Belkin, 2021]. In this way, GS^{2}P transforms $\boldsymbol{z}_{i,j}$ into a high-dimensional vector $\boldsymbol{h}_{i,j}$ and constructs a Ranker (i.e., MLP-based LTR model) for the LTR task with several popular ranking loss functions.

3 Experiments

3.1 Experimental Setup

Datasets. We carry out the offline experiments on a standard and publicly available dataset Web30K [Qin and Liu, 2013] and a real-world dataset *commerical dataset* collected from Baidu search engine. Specifically, the commercial Dataset contains 50,000 queries. The dataset is annotated by a group of professionals on the crowdsourcing platform, who assign a score between 0 and 4 to each query-document pair.

Metrics. To assess the performance of various ranking systems comprehensively, we leverage the following metrics. Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2017] is a standard listwise accuracy metric, which has been commonly used in research and industrial community. For our online evaluation, we utilize the Good vs. Same vs. Bad (GSB) [Zhao *et al.*, 2011], which is an online pairwise-based evaluation methodology evaluated by annotators. Considering the confidentiality of commercial information, we only report the difference between the results of GS^{2}P and the online *legacy system* [Zou *et al.*, 2021].

Loss Functions and Competitor Systems In this work, we leverage the following advanced ranking loss functions to evaluate the proposed model comprehensively, such as RMSE, RankNet [Burges *et al.*, 2005], LambdaRank [Burges *et al.*, 2006], ListNet [Cao *et al.*, 2007], ListMLE [Xia *et al.*, 2008], ApproxNDCG [Qin *et al.*, 2010], and NeuralNDCG [Pobrotyn and Białobrzeski, 2021]. As for the ranking model, we choose the following state-of-the-art ranking models as the competitor for GS^{2}P, such as MLP, Context-aware Ranker (CR) [Pobrotyn *et al.*, 2020], XGBoost [Chen and Guestrin, 2016] and LightGBM [Ke *et al.*, 2017].

3.2 Offline Experimental Results

Overall Results. Table 1 and 2 present the average results for offline evaluation, where GS^{2}P is compared with competitors on Web30K and the commercial dataset. Intuitively, we could observe GS^{2}P outperforms all competitors with different losses under various ratios of labeled data on two datasets. More specifically, GS^{2}P with NeuralNDCG gets 3.60% and nearly 3.57% higher NDCG@4 and NDCG@10 on Web30K dataset, compared with the pointwise-based self-trained MLP model with NeuralNDCG. On Commercial Dataset, GS^{2}P on average obtains nearly 2.84% and 3.14% improvement on NDCG@4 and NDCG@10, when compared with NeuralNDCG. GS^{2}P+NeuralNDCG could gain the most improvement under the less ratio of labeled data on both metrics on two datasets, which demonstrates the effectiveness of GS^{2}P under low-resource situations.

Methods	**5%**		**10%**		**15%**		**20%**	
	@4	@10	@4	@10	@4	@10	@4	@10
XGBoost	48.39	52.12	52.83	56.45	56.14	60.03	58.03	62.61
LightGBM	50.48	53.50	54.13	59.04	57.00	62.14	60.47	65.82
RMSE	49.73	53.42	54.13	57.86	57.43	61.34	59.42	64.76
RankNet	49.32	53.07	53.76	57.37	57.08	60.92	59.17	64.25
LambdaRank	50.82	54.24	55.07	58.62	58.16	62.05	61.12	65.28
ListNet	50.26	53.61	54.52	58.04	57.81	61.47	59.74	64.82
ListMLE	48.73	52.46	53.08	56.70	56.32	60.25	58.42	63.68
ApproxNDCG	49.08	52.75	53.44	57.02	56.79	60.61	58.84	64.01
NeuralNDCG	50.68	53.89	54.88	58.31	58.02	61.82	61.03	64.97
$\mathrm{CR}_{\mathrm{RMSE}}$	50.43	53.63	54.52	58.70	56.90	61.74	60.42	65.22
$\mathrm{CR}_{\mathrm{RankNet}}$	50.86	54.06	54.98	58.26	57.32	61.82	60.83	65.61
$\mathrm{CR}_{\mathrm{LambdaRank}}$	52.47	55.67	56.13	59.84	58.90	63.79	61.87	66.59
$\mathrm{CR}_{\mathrm{ListNet}}$	52.45	55.64	56.08	59.82	58.74	63.24	62.28	67.09
$\mathrm{CR}_{\mathrm{ListMLE}}$	51.05	54.30	54.76	58.46	57.53	62.01	61.04	65.83
$\mathrm{CR}_{\mathrm{ApproxNDCG}}$	51.92	55.08	55.68	59.40	58.42	62.87	62.00	66.75
$\mathrm{CR}_{\mathrm{NeuralNDCG}}$	52.06	55.31	55.87	59.61	58.67	63.20	62.18	66.84
$\mathrm{GS}^2\mathrm{P}_{\mathrm{RMSE}}$	52.72	55.48	55.89	59.60	58.82	63.13	61.92	66.24
$\mathrm{GS}^2\mathrm{P}_{\mathrm{RankNet}}$	53.13	55.93	56.20	59.92	58.94	63.41	62.28	66.67
$\mathrm{GS}^2\mathrm{P}_{\mathrm{LambdaRank}}$	53.67	56.72	56.90	60.76	59.58	64.19	62.95	67.65
$\mathrm{GS}^2\mathrm{P}_{\mathrm{ListNet}}$	54.00	57.18	57.28	61.04	59.93	64.50	63.38	67.96
$\mathrm{GS}^2\mathrm{P}_{\mathrm{ListMLE}}$	53.41	56.24	56.51	56.51	59.20	63.72	62.50	66.88
$\mathrm{GS}^2\mathrm{P}_{\mathrm{ApproxNDCG}}$	54.23	57.32	57.44	61.12	60.12	64.62	63.58	68.05
$\mathrm{GS}^2\mathrm{P}_{\mathrm{NeuralNDCG}}$	**54.36**	**57.43**	**57.62**	**61.25**	**60.28**	**64.76**	**63.72**	**68.12**

Table 2: Results for Commercial Dataset on NDCG across diverse labeled data percentages.

	$\mathbf{GS^2P}_{\mathrm{ApproxNDCG}}$		$\mathbf{GS^2P}_{\mathrm{NeuralNDCG}}$	
	Random	Long-Tail	Random	Long-Tail
ΔGSB	**+3.00%**	**+4.00%**	**+5.50%**	**+6.50%**

Table 3: Performance improvements of $\mathrm{GS}^2\mathrm{P}$ with ApproxNDCG loss and $\mathrm{GS}^2\mathrm{P}$ with NeuralNDCG loss for the online evaluation.

3.3 Online Evaluation

To comprehensively evaluate our proposed model, we conduct a manual comparison experiment. Intuitively, manual comparison results are presented in Table 3. In particular, we observe that our proposed model outperforms the online legacy system by a large margin for random and long-tail (i.e., the search frequency of the query is lower than 10 per week) queries. Specifically, $\mathrm{GS}^2\mathrm{P}$ with NeuralNDCG loss achieves the largest improvement compared with the legacy system with 5.50% and 6.50% for random and long-tail queries, respectively. Moreover, $\mathrm{GS}^2\mathrm{P}$ with ApproxNDCG loss also improves the performance for random and long-tail queries.

Figure 2 illustrates the relative performance between $\mathrm{GS}^2\mathrm{P}$ and the base model, expressed via $\Delta NCDG@4$. Logically, $\mathrm{GS}^2\mathrm{P}$ shows marked enhancement in performance across all days when compared to the base system, evidencing its practical capability in upgrading the efficacy of a large-scale search engine. Even more impressively, $\mathrm{GS}^2\mathrm{P}$ has shown substantial growth on this large-scale platform. A prominent highlight is $\mathrm{GS}^2\mathrm{P}$ outperforming the online base model by a significant margin of 0.61% relative improvement on $\Delta NCDG@4$, a feat achieved by the NeuralNDCG loss-trained model using a nominal 5% labeled data ratio. $\mathrm{GS}^2\mathrm{P}$ has showcased consistent performance across both online and offline platforms.

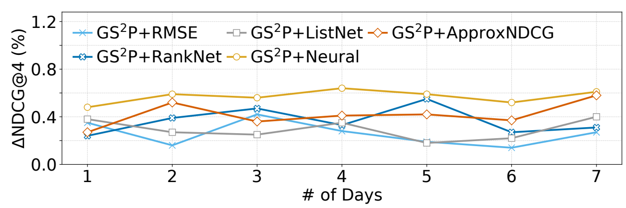

Figure 2: A/B test results of $\mathrm{GS}^2\mathrm{P}$ and the legacy system for 7 days (t-test with $p < 0.05$ over the baseline).

4 Conclusion

In this work, we design, implement and deploy a generative semi-supervised pre-trained model $\mathrm{GS}^2\mathrm{P}$ on a real-world large-scale search engine to address the problems of LTR under semi-supervised settings. We substantiate the effectiveness of $\mathrm{GS}^2\mathrm{P}$ through comprehensive offline and online analyses, juxtaposed against an extensive lineup of rivals. The offline trials denote a considerable leap in $\mathrm{GS}^2\mathrm{P}$'s performance relative to other baselines. Furthermore, $\mathrm{GS}^2\mathrm{P}$ significantly enhances the online ranking efficacy in practical applications, mirroring the positive outcomes observed in the offline experiments.

Acknowledgments

This work was supported in part by NSFC grant 62141220, 61972253, U1908212, 62172276, 61972254, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Shanghai Science and Technology Development Funds 23YF1420500, Open Research Projects of Zhejiang Lab No. 2022NL0AB01.

References

[Belkin *et al.*, 2019] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[Belkin, 2021] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

[Bruch *et al.*, 2019] Sebastian Bruch, Masrour Zoghi, Michael Bendersky, and Marc Najork. Revisiting approximate metric optimization in the age of deep neural networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1241–1244, 2019.

[Burges *et al.*, 2005] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Machine Learning, Proceedings of the Twenty-Second International Conference, ICML*, pages 89–96, 2005.

[Burges *et al.*, 2006] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 193–200, 2006.

[Cao *et al.*, 2007] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference*, pages 129–136, 2007.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[Järvelin and Kekäläinen, 2017] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. *SIGIR Forum*, 51(2):243–250, 2017.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 3146–3154, 2017.

[Li *et al.*, 2020] Minghan Li, Xialei Liu, Joost van de Weijer, and Bogdan C. Raducanu. Learning to rank for active learning: A listwise approach. In *25th International Conference on Pattern Recognition*, pages 5587–5594, 2020.

[Li *et al.*, 2022] Yuchen Li, Haoyi Xiong, Linghe Kong, Rui Zhang, Dejing Dou, and Guihai Chen. Meta hierarchical reinforced learning to rank for recommendation: A comprehensive study in moocs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 302–317, 2022.

[Li *et al.*, 2023a] Yuchen Li, Haoyi Xiong, Linghe Kong, Zeyi Sun, Hongyang Chen, Shuaiqiang Wang, and Dawei Yin. Mpgraf: a modular and pre-trained graphformer for learning to rank at web-scale. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 339–348. IEEE, 2023.

[Li *et al.*, 2023b] Yuchen Li, Haoyi Xiong, Linghe Kong, Qingzhong Wang, Shuaiqiang Wang, Guihai Chen, and Dawei Yin. S2phere: Semi-supervised pre-training for web search over heterogeneous learning to rank data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4437–4448, 2023.

[Li *et al.*, 2023c] Yuchen Li, Haoyi Xiong, Linghe Kong, Shuaiqiang Wang, Zeyi Sun, Hongyang Chen, Guihai Chen, and Dawei Yin. Ltrgcn: Large-scale graph convolutional networks-based learning to rank for web search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 635–651. Springer, 2023.

[Li *et al.*, 2023d] Yuchen Li, Haoyi Xiong, Linghe Kong, Rui Zhang, Fanqin Xu, Guihai Chen, and Minglu Li. Mhrr: Moocs recommender service with meta hierarchical reinforced ranking. *IEEE Transactions on Services Computing*, 2023.

[Li *et al.*, 2023e] Yuchen Li, Haoyi Xiong, Qingzhong Wang, Linghe Kong, Hao Liu, Haifang Li, Jiang Bian, Shuaiqiang Wang, Guihai Chen, Dejing Dou, et al. Coltr: Semi-supervised learning to rank with co-training and over-parameterization for web search. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Liu *et al.*, 2021] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. Pre-trained language model for web-scale retrieval in baidu search. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3365–3375, 2021.

[Pobrotyn and Białobrzeski, 2021] Przemysław Pobrotyn and Radosław Białobrzeski. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. *arXiv preprint arXiv:2102.07831*, 2021.

[Pobrotyn *et al.*, 2020] Przemysław Pobrotyn, Tomasz Bartczak, Mikołaj Synowiec, Radosław Białobrzeski, and Jarosław Bojar. Context-aware learning to rank with self-attention. *arXiv preprint arXiv:2005.10084*, 2020.

[Qin and Liu, 2013] Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

[Qin *et al.*, 2010] Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for direct optimization of information retrieval measures. *Inf. Retr.*, 13(4):375–397, 2010.

[Rahimi and Recht, 2007] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pages 1177–1184, 2007.

[Szummer and Yilmaz, 2011] Martin Szummer and Emine Yilmaz. Semi-supervised learning to rank with preference regularization. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, pages 269–278, 2011.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2021] Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H. Chi. DCN V2: improved deep & cross network and practical lessons for web-scale learning to rank systems. In *WWW '21: The Web Conference*, pages 1785–1797, 2021.

[Werner, 2022] Tino Werner. A review on instance ranking problems in statistical learning. *Mach. Learn.*, 111(2):415–463, 2022.

[Xia *et al.*, 2008] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference*, pages 1192–1199, 2008.

[Yang and Ying, 2023] Tianbao Yang and Yiming Ying. AUC maximization in the era of big data and AI: A survey. *ACM Comput. Surv.*, 55(8):172:1–172:37, 2023.

[Zhang *et al.*, 2016] Xin Zhang, Ben He, and Tiejian Luo. Training query filtering for semi-supervised learning to rank with pseudo labels. *World Wide Web*, 19(5):833–864, 2016.

[Zhao *et al.*, 2010] Shiqi Zhao, Haifeng Wang, and Ting Liu. Paraphrasing with search engine query logs. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference*, pages 1317–1325, 2010.

[Zhao *et al.*, 2011] Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. Automatically generating questions from queries for community-based question answering. In *Proceedings of 5th international joint conference on natural language processing*, pages 929–937, 2011.

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