Improving Product Search with Season-Aware Query-Product Semantic Similarity

Haoming Chen
haomingchen0320@gmail.com
Harvard University
Boston, MA, USA

Yetian Chen, Jingjing Meng, Yang Jiao, Yikai Ni,
Yan Gao, Michinari Momma, Yi Sun
{yetichen,ajmeng,jaoyan,yika,yanngao,michi,yisun}@amazon.com
Amazon
Seattle, WA, USA

ABSTRACT

Product search for online shopping should be season-aware, i.e., presenting seasonally relevant products to customers. In this paper, we propose a simple yet effective solution to improve seasonal relevance in product search by incorporating seasonality into language models for semantic matching. We first identify seasonal queries and products by analyzing implicit seasonal contexts through time-series analysis over the past year. Then we introduce explicit seasonal contexts by enhancing the query representation with a season token according to when the query is issued. A new season-enhanced BERT model (SE-BERT) is also proposed to learn the semantic similarity between the resulting seasonal queries and products. SE-BERT utilizes Multi-modal Adaption Gate (MAG) to augment the season-enhanced semantic embedding with other contextual information such as product price and review counts for robust relevance prediction. To better align with the ranking objective, a listwise loss function (neural NDCG) is used to regularize learning. Experimental results validate the effectiveness of the proposed method, which outperforms existing solutions for query-product relevance prediction in terms of NDCG and Price Weighted Purchases (PWP).

KEYWORDS

seasonal relevance, product search, language model

ACM Reference Format:
Haoming Chen and Yetian Chen, Jingjing Meng, Yang Jiao, Yikai Ni, Yan Gao, Michinari Momma, Yi Sun. 2023. Improving Product Search with Season-Aware Query-Product Semantic Similarity. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543873.3587625

1 INTRODUCTION

Seasonality is an important dimension for relevance matching in online shopping product search. Given the same query, a user's intent may vary depending on the season when the query is issued. Consequently, the change of user intent would directly affect query-product relevance matching for product search. As shown in Figure 1, the query "jacket for men" has two different sets of relevant products in winter and in summer: it leads to more sales of the product "Wind Cheater Jacket" in summer/fall season, while more sales of the product "Bomber Jacket" in winter. Such queries are called *seasonal* queries. According to [16], 39% queries are highly seasonally relevant to the time of search and would benefit from handling seasonality in ranking.

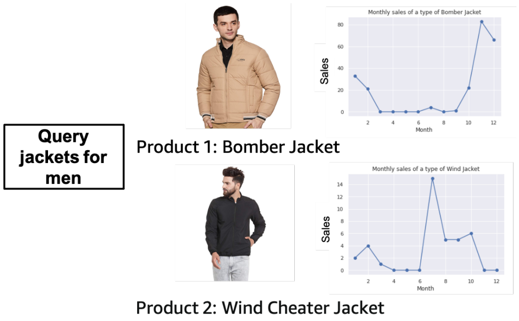

Figure 1: Monthly Sales of two types of jackets resulting from the query "jackets for men". In this example, "jackets for men" is an implicit seasonal query, as customers have different purchase preferences during different seasons, even though the query does not explicitly convey the seasonal intent such as "winter" or "summer". The y-axis has been re-scaled to omit absolute numbers.

It is worth noting that the majority of seasonal queries are implicitly seasonal, which do not contain explicit seasonal keywords. For example, "jackets for men" in the previous example is an implicit seasonal query. On the contrary, "winter jacket" is an explicit seasonal query, whose seasonal intent is explicitly conveyed by including "winter" in the query keywords. Implicitly seasonal queries pose great challenges for product search. First, unlike explicit seasonal queries, because of the absence of seasonal keywords in implicit queries, it is difficult to infer seasonal relevance between the implicit queries and products purely based on lexical or semantic matching between query keywords and product titles. Second, existing works on detecting seasonal queries [8, 14] mainly rely on analyzing the temporal dynamics of query frequency, which may fall short when it comes to implicit seasonal queries.

To explicitly address the seasonality problem, Yang et al.[16] propose to model the seasonality of a product as the probability of being purchased in each of the twelve months estimated by the proportion of a product's annual sales concentrated in each month (called MSC score) based on historical sales. A neural model is

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9419-2/23/04.
https://doi.org/10.1145/3543873.3587625

trained, which takes the product title as input, to predict the MSC score for products without historical sales data. However, the MSC score is a product-level score that fails to capture the purchase intent conveyed by query keywords. Incorporating temporal information into search ranking has been studied in [3, 7, 16]. Dakka et al. [3] automatically identify important time intervals for time-sensitive queries over a news archive, and use these intervals to adjust the document relevance scores by boosting the scores of documents published within the important intervals. Kanhabua and Nørvåg [7] employ both entity-based and temporal features derived from annotation data to learn a learning to rank model. The entity-based features aim to capture the semantic similarity between a query and a document, whereas the temporal features measure the temporal similarity. Recently, language models such as BERT have become popular for semantic matching in product search [1, 4, 6, 12, 13]. To study the temporal effects in language models, Dhingra et al [4] propose to jointly modeling text with its timestamp to improve memorization of seen facts and calibration on predictions about unseen facts from future time periods. Agarwal and Nenkova [1] systematically study the temporal effects on downstream language tasks, including temporal model deterioration and the benefit from retraining models on more recent data to improve model performance. Rosin, Guy and Radinsky [12] propose TempoBERT, which uses time as an additional context of texts and perform time masking in pertraining to facilitate the acquisition of temporal knowledge. Later, Rosin and Radinsky [13] extend multi-head attention to include an additional temporal attention.

Motivated by the above, we explore the use of language models for semantic relevance prediction, enhanced by the additional time context. However, our investigation focuses on the seasonality, and how it can be utilized to better measure the relevance between a seasonal query and products. We start by defining seasonality at the query level and obtain corresponding seasonal products. Then we propose to explicitly introduce the seasonal context by directly concatenating the time token at the beginning of query texts. Built upon popular BERT-based language models, we propose a season-enhanced BERT model (SE-BERT) to learn the semantic similarity between the query texts and product texts (product title, material, etc.). The semantic embedding is augmented by other contextual information such as product price and review counts for robust relevance prediction through a Multi-modal Adaption Gate (MAG) [11]. We study the effects of different loss functions on the learning outcome and demonstrate the effectiveness of the proposed model in predicting query-product relevance by incorporating seasonality. Comprehensive experiments reveal the importance of incorporating the seasonal signals into semantic matching and highlight the necessity of season-aware language models for product search.

2 PROBLEM FORMULATION AND MODELING

Let $\mathbb{Q}$ be the set of all possible queries and $\mathbb{A}$ be the set of all products. For a given query $q \in \mathbb{Q}$, let $A^q = \{a_i\}_{i=1}^{n_q} \subset \mathbb{A}$ be the subset of n_q matched products. Let a query-product pair (q, a_i) be represented by a p-dimensional feature vector $\mathbf{x}_i^q \in \mathbb{R}^p$. Our goal is to learn a scoring function $f_\theta : \mathbb{R}^p \to \mathbb{R}$ that can assign a score $s_i^q(t)$ to each (q, a_i) pair from its corresponding vector representation, conditioning on a specific season t, i.e., $(\mathbf{x}_i^q|t) \mapsto s_i^q(t)$. Specifically, t is taken as an input to f_θ. The items can then be ranked in descending order of scores.

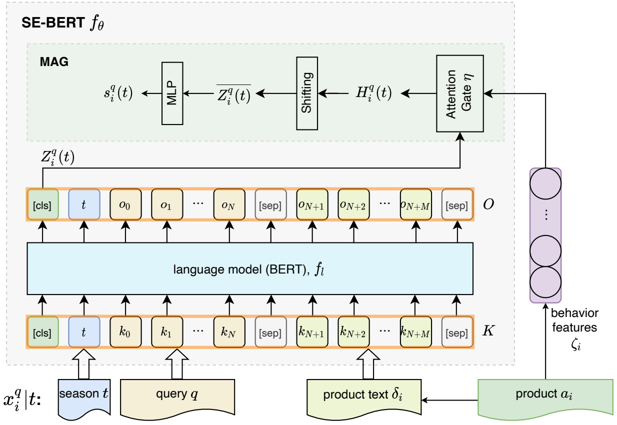

Figure 2: The overview of SE-BERT.

2.1 Learning seasonal semantic relevance

To address the challenge of capturing the seasonal semantic relevance between the text of query and products, we propose Season-enhanced BERT (SE-BERT) model. As shown in Figure 2, SE-BERT involves a pretrained BERT as a language model to capture the season-related semantic relevance between the semantic context of a query and a product. As behavior signals are important for product search and reflect the short-term and long-term relevance between a query and products, we incorporate behavior information such as product price and review counts as additional contexts to improve relevance prediction through a Multi-modal Adaption Gate (MAG) [11]. Compared with other feature fusing approaches such as concatenation, MAG highlights the relevant information from non-text data conditioned on the current embedding vector. The architecture of our proposed SE-BERT is shown in Fig. 2.

Seasonal query identification. Constructing the seasonal query set is one of the keys to learn seasonal semantic relevance. Given a query q, the bestselling product set in month m is defined as A_m^q, where $1 \le m \le 12$. If q is a seasonal query, customers tend to purchase different sets of products depending on the season/occassion, resulting in a varying A_m^q as m changes. Therefore, to capture such implicit seasonal contexts, we use the Jaccard Index to discriminate season queries. For query q, the Jaccard Index matrix J_{m_1,m_2}^q between $A_{m_1}^q$ and $A_{m_2}^q$, where $1 \le m_1, m_2 \le 12$ and $m_1 \ne m_2$, is defined as

$$J_{m_1,m_2}^q = |A_{m_1} \cap A_{m_2}| / |A_{m_1} \cup A_{m_2}|. \tag{1}$$

The mean Jaccard index of query q over 12 months is

$$\overline{J^q} = \frac{1}{12 \times 12} \sum_{m_1=1,m_2=1}^{12,12} J_{m_1,m_2}^q. \tag{2}$$

We then define the seasonal query set as $\{Q, t, \gamma\} = \{q | \overline{J^q} \le \gamma\}$, where γ is the seasonality threshold.

Season-enhanced semantic relevance. As a common approach to capture the relevance between two sentences, a language model takes the tokenized sentences with a separator [SEP] in between.

However, it is challenging to capture seasonality when the seasonal time token is not explicitly present in the tokenized sentences. To better learn the seasonal semantic relevance, we propose to explicitly prepend the seasonal time token at the beginning of the query texts.

Formally, given a query and season $(q_i, t) \in \{Q, t, \gamma\}$, and a product $a_i, a_i \in A^q$, the product text of a_i is δ_i. Denote the seasonal time token as t, the tokenized input to a language model is

$$K = [[cls]; t; q_i; [sep]; \delta_i; [sep]]. \tag{3}$$

The token embedding vector is

$$O = f_l(K), \tag{4}$$

where f_l is the language model. $Z_i^q(t)$ is the semantic embedding of the $[cls]$ token, where $Z_i^q(t) \in O$. We hypothesize that prepending the seasonal time token to a seasonal query will help the self-attentions of language models like BERT to attend on the seasonal signal during training.

Enhancing seasonal semantic relevance with behavior features. Given tabular behavior features ζ_i, MAG concatenates the semantic embedding $Z_i^q(t)$ with ζ_i to produce a gating vector η, where W_η is the weight matrix, b_i is a scalar bias, and R is a non-linear activation function. The gating vector η measures the effect of additional modality conditional on the current context.

$$\eta = \mathrm{R}\left(W_\eta\left[Z_i^q(t); \zeta_i\right] + b_i\right). \tag{5}$$

Subsequently, a non-text displacement vector $H_i^q(t)$ is created from the tabular behavior features ζ_i with its gating vectors.

$$H_i^q(t) = \eta \cdot \left(W_h Z_i^q(t)\right) + b_h, \tag{6}$$

where W_h is a weight metrix and b_h is a scalar bias. Finally, we take a weighted summation of the semantic embedding $Z_i^q(t)$ with its associated non-text displacement.

$$\overline{Z}_i^q(t) = Z_i^q(t) + \alpha H_i^q(t), \tag{7}$$

where $\overline{Z}_i^q(t)$ is the enhanced semantic embedding with behavior features. The weight of the displacement vector is controlled through a scaling factor α to ensure that it remains in a reasonable range.

$$\alpha = \min(\left(\frac{\|Z_i^q(t)\|_2}{\|H_i^q(t)\|_2}\right) * \beta, 1), \tag{8}$$

where $\|\cdot\|$ is the L_2 norm and β is a hyperparameter. β=1.0 in our experiments.

2.2 Loss functions

While not ideal, pointwise losses such as mean-square error (MSE) [5] are still used for its simplicity and scalability in product search systems. MSE measures the error between the predicted relevance score $s_i^q(t)$ of a given query-product pair (q, a_i) and their ground truth relevance. Purchase conversion rate is a commonly used surrogate for the ground truth (q, a_i) relevance, defined as $PCR(q, a_i, t) = \frac{Purchase(q, a_i, t)}{Impressions(q, a_i, t)}$.

As pointwise loss functions treat the relevance prediction as a simple regression of the ground truth relevance for each individual query-product pair (q, a_i), they ignore possible interactions between products. Moreover, pairwise loss functions are only loosely related to the evaluation metrics for product ranking such as Normalised Discounted Cumulative Gain (NDCG), causing a mismatch between the optimization objective and the evaluation criterion. Therefore, we propose to use listwise loss functions that better align with the evaluation metric in our model [10, 15]. However, NDCG cannot be directly used as the loss function, as it is non-differentiable. Therefore, we use a surrogate loss function, NeuralNDCG [9], to approximate the NDCG score. Below we will first define NDCG, then summarize NeuralNDCG.

Let r_j denote the relevance of the product ranked at the j-th position, $g(\cdot)$ denote a gain function and $d(\cdot)$ denote a discount function. The Discounted Cumulative Gain (DCG) at the k-th position is defined as $\mathrm{DCG@}k = \sum_{j=1}^{k} g(r_j)\, d(j)$, and NDCG at the k-th position is defined as

$$\mathrm{NDCG@}k = \frac{1}{\mathrm{maxDCG\,@}k}\mathrm{DCG@}k \tag{9}$$

where maxDCG @k represents the maximum possible value of DCG@k. Since the sorting operator is the source of discontinuity in NDCG (and other IR metrics), by substituting it with a differentiable approximation we obtain a smooth variant of the metric. The NeuralNDCG method aims to approximate the gain function $g(\cdot)$.

$$\mathrm{NeuralNDCG}_k(\tau)(s, y) = N_k^{-1} \sum_{j=1}^{k} (\mathrm{scale}(\widehat{P}) \cdot g(y))_j \cdot d(j) \tag{10}$$

where $\tau = 1.0$ is a temperature parameter controlling the accuracy of approximation. N_k^{-1} is the maxDCG at k-th rank, $\widehat{P}$ is the permutation matrix to approximate the sorting operator, y represents the ground truth labels, $scale(\cdot)$ is Sinkhorn scaling and $g(\cdot)$ and $d(\cdot)$ are their gain and discount functions.

3 EXPERIMENTAL SETTINGS

Dataset To train the model, we sampled seasonal queries from a year of customer search record from an online shopping platform. All queries in the candidate pool is required to have at least one purchase. Additionally, we require a *(query, product, season)* triplets with at least 20 impressions to ensure reliable PCR estimation. As a result, the final train set consists of 2,800 queries together with an average of 120 associated impressed products. To ensure a fair evaluation, we created our test set by sampling queries queries from January to June of the following year. Specially, the test set contains about 40% new products not seen in the training set.

For training with the listwise loss function, the 120 products associated with a given query is randomly sampled into groups of 16 products with replacement. The listwise loss function minimizes the loss on these groups. Our model was trained with Adamw optimizer with a learning rate of 0.00003 and a dropout rate of 0.1.

Evaluation Metrics We compare the performance of all models using both relevance and business metrics. We report ranking performance by measuring NDCG at rank cutoffs at 8 and 22, and examine the business impact with Price Weighted Purchases (PWP) with the same cutoffs (i.e., PWP@K for K=8, 22)[1].We use NDCG as the primary metric for model selection and evaluation as it is the standard objective for relevance model training. PWP@K serves as secondary metric as it indicates business impact.

[1] To compute the PWP@K, we first sort the query-product in each query group in the evaluation data in descending order of the output score of the model. We then compute the average of PWP in the top K results

Table 1: Summary of methods performances without or with seasonal signal (With MSE as loss function)

Without seasonal signal					With seasonal signal				
Method	NDCG@8	NDCG@22	PWP@8	PWP@22	Method	NDCG@8	NDCG@22	PWP@8	PWP@22
Semantic relevance									
BERT	0.2502	0.3661	1.0	1.0	SE-BERT	0.2530 (+1.12%)	0.3687 (+0.71%)	1.009 (+0.89%)	1.007 (+0.70%)
+ Behavior features									
BERT+MAG	0.2770	0.3834	1.090	1.076	SE-BERT+MAG	0.2835 (+2.35%)	0.3887 (+1.38%)	1.102 (+1.04%)	1.077 (+0.10%)
BERT+Con	-	-	-	-	SE-BERT+Con	0.2830	0.3882	1.109	1.091
BERT+Add	-	-	-	-	SE-BERT+Add	0.2827	0.3880	1.108	1.088

PWP@K (K=8 or 22) are normalized by PWP@K for the baseline method (BERT without seasonal signal and without behavior feature) to omit absolute numbers. $+x\%$ in parentheses indicate the percentage gain of corresponding metrics against the models trained with the similar method but without using seasonal signal. These highlight the effect of adding seasonal signal for relevance modeling. Con: concatenation, Add: addition.

4 EXPERIMENTAL RESULTS

We first studied the impact of adding seasonal signal as additional input into the language model to learn the query-product semantic similarity. We trained a series of models by varying the input to our model. For input to the language model, we use product title as well as metadata such as style, color, materials. In addition to textual information, behavior features such as product price, product sale velocity, user review rating, number of user reviews, product ages can provide additional signal to enhance the seasonal relevance prediction. Thus, we combine the output from the language model with these behavior features as input to the MLP layers for final relevance prediction.

We first studied the effect of different time granularity for defining seasons. We tried two definitions: 1) each month is a "season"; 2) six seasons defined as: Spring (March and April), Summer (May and Jun), Moonsoon (July and August), Autumn (September and October) and Pre-winter (November, December) and Winter (January and February)[2]. Granularity at month-level has a well-defined boundary across months and is independent of the geo-location, but it suffers from the sparsity issue for a given (query, product, time) triplet as there are around 70% of triplets with no purchase records in purchase data. Our results show that the 6-season definition outperforms the month-based season classification. Therefore, we use the 6-season definition for all our following experiments.

4.1 Effectiveness of Seasonal Signal

Table 1 presents the summary of model performance for with and without seasonal signals. Our first observations is that with only the textual input, SE-BERT outperforms BERT. Using BERT as the baseline, SE-BERT improves NDCG@8 and NDCG@22 by 1.12% and 0.71%, respectively. It also improves PWP@8 and PWP@22 by 0.89% and 0.70%, respectively. It demonstrates the effectiveness of enhancing the seasonality in semantic relevance matching. Secondly, we observe that augmenting the semantic embedding with behavior features improves the model performance. SE-BERT+MAG exceeds SE-BERT with a large margin on all metrics. Specifically, compared with BERT+MAG, which is not seasonal-enhanced, SE-BERT+MAG achieves 2.35% NDCG@8 and 1.38% NDCG@22 gains. We hypothesize that the increases in the gain (e.g. the gain for NDCG@8 is increased to 2.35% from 1.12%) is attributed to the benefit that MAG brings to SE-BERT in learning seasonal semantic relevance.

Additionally, we compare two other feature fusion methods with MAG: concatenation (SE-BERT+Con) and addition (SE-BERT+Add). For concatenation, we concatenate non-text vector with the semantic embedding of SE-BERT. For addition, we project the semantic embedding and the behavior feature vector into the same dimension, then add them element-wise. Compared with SE-BERT+Con and SE-BERT+Add, SE-BERT+MAG achieves the best performance on the primary evaluation metrics, NDCG@8 and NDCG@22.

4.2 Effectiveness of Listwise Loss Function

We study the impact of using list-wise loss (NeuralNDCG) over point-wise loss (MSE). Specifically, we compare two models, both with seasonal tokens and behavior features, but trained with two different loss functions: MSE and NeuralNDCG. As shown in Table 2, training model with NeuralNDCG outperforms MSE consistently across all metrics.

Table 2: Ranking performance of using different loss functions (Both models are trained with SE-BERT with MAG to fuse behavior features with BERT output)

Loss function	NDCG@8	NDCG@22	PWP@8	PWP@22
MSE	0.2835	0.3887	1.102	1.077
NeuralNDCG	**0.2920**	**0.3945**	**1.124**	**1.096**

5 CONCLUSIONS

In this paper, we propose a season-enhanced language model to predict seasonal relevance of query-product pairs for product search. We demonstrate that including the season signal can improve the ranking performance of neural ranking model. We also show the effectiveness of incorporating multiple modalities (behavior signals) into existing language model through a MAG adaption. Fine-tuning the season-enhanced BERT model with listwise loss function (neuralNDCG) further boosts the model performance by bridging the gap between the loss function and ranking metric. Our future work includes incorporating other modalities such as product image and geolocation for seasonal relevance prediction, as we have demonstrated that additional contextual information can benefit relevance matching, and product images may contain extra seasonal signals, especially for fashion products.

[2] This definition of 6 seasons is commonly accepted season classification in India[2].

REFERENCES

[1] Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics* 10 (2022), 904–921.

[2] Wikipedia contributors. 2023. https://en.wikipedia.org/wiki/Ritu_(Indian_season)

[3] Wisam Dakka, Luis Gravano, and Panagiotis G Ipeirotis. 2008. Answering general time sensitive queries. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 1437–1438.

[4] Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics* 10 (2022), 257–273.

[5] Muhammad Ibrahim and Mark Carman. 2016. Comparing pointwise and listwise objective functions for random-forest-based learning-to-rank. *ACM Transactions on Information Systems (TOIS)* 34, 4 (2016), 1–38.

[6] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2021. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv preprint arXiv:2110.08534* (2021).

[7] Nattiya Kanhabua and Kjetil Nørvåg. 2012. Learning to rank search results for time-sensitive queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2463–2466.

[8] Anagha Kulkarni, Jaime Teevan, Krysta M Svore, and Susan T Dumais. 2011. Understanding temporal query dynamics. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 167–176.

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

[10] Razieh Rahimi, Ali Montazeralghaem, and James Allan. 2019. Listwise neural ranking models. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 101–104.

[11] Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, Amir Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. Integrating multimodal information in large pretrained transformers. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, Vol. 2020. NIH Public Access, 2359.

[12] Guy D Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 833–841.

[13] Guy D Rosin and Kira Radinsky. 2022. Temporal Attention for Language Models. *arXiv preprint arXiv:2202.02093* (2022).

[14] Milad Shokouhi. 2011. Detecting seasonal queries by time-series analysis. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 1171–1172.

[15] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*. 1192–1199.

[16] Haode Yang, Parth Gupta, Roberto Fernández Galán, Dan Bu, and Dongmei Jia. 2021. Seasonal Relevance in E-Commerce Search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4293–4301.