TransformerRank: Enhancing Listwise Ranking with Advanced Attention Mechanisms

Anonymous ACL submission

Abstract

In the realm of recommendation systems and search engine optimization, the comprehensive understanding of listwise item dependencies has emerged as a pivotal challenge. Traditional ranking methods, predominantly pointwise or pairwise, have been limited in capturing the intricate dynamics within item lists. In this study, we developed TransformerRank, a novel approach specifically tailored for the complexities of listwise ranking. This method innovatively employs a custom transformer model within a sliding window technique, extending beyond the capabilities of conventional ranking algorithms.

Our extensive experiments conducted on diverse datasets, including TripClick, Yahoo!, and ORCAS, demonstrated TransformerRank's superiority. It consistently outperformed established methods across key metrics such as NDCG@10 and MAP. Additionally, an ablation study was executed to determine the balance between accuracy and computational efficiency, underscoring the practicality of our approach.

TransformerRank provides a significant advancement in the field of listwise ranking. It not only enhances the accuracy and efficiency of ranking systems but also offers a deeper insight into the dynamics of item interdependencies. This research expands the potential applications in data science and natural language processing, setting a new benchmark for future explorations in leveraging listwise dependencies in sequence data.

1 Introduction

Item ranking, a vital component in domains such as e-commerce, web search, and personalized content delivery, represents a significant area of research within machine learning and information retrieval. Traditional item ranking methodologies primarily involve pointwise methods like RankNet (Burges et al., 2005), and listwise approaches including ListNet (Cao et al., 2007a) and SoftRank (Taylor et al., 2009). Despite their foundational impact, these conventional approaches often struggle to capture the dynamic interactions and complex dependencies that are characteristic of modern, data-intensive item ranking scenarios (Li et al., 2014; Wang et al., 2020; Zhao et al., 2021).

In the current landscape, most commonly used ranking systems are based on pointwise approaches or simplistic interpretations of listwise ranking. Methods such as TPrank or Uni-retrieval (Qiao et al., 2019; Zhang, 2022) typically focus on single-item relevance, falling short in learning from result-oriented data such as click logs. This limitation becomes evident in the ever-changing world of e-commerce and web search, where user preferences and item relevances continually evolve, presenting a demand for models that are both precise and adaptable (Wang et al., 2020).

Consider the scenario where a user searches for "Healthy Snack Options." The rank logs depicted in Table 1 offer insightful observations into how user interaction patterns shift with varying item list compositions. Traditional pointwise models, or those prioritizing item relevancy in isolation, might mirror the "Alone" situation in the table. They often predict certain items as relevant based on individual assessments. However, when these items are presented together, as seen in the "With Others" column, their interactions and user clicks change significantly. This discrepancy underscores the need for a list-to-list learning approach that is result-oriented, focusing on the actual outcomes of item combinations rather than isolated relevancy predictions. Such an approach acknowledges the importance of contextual relevance and inter-item dependencies, essential aspects often overlooked in pointwise ranking models but critical in real-world ranking scenarios.

The emergence of transformer architectures

Snack Option	Alone	With Others
Almonds	Yes	No
Fruit Salad	Yes	Yes
Granola Bars	No	No
Greek Yogurt	Yes	No
Dark Chocolate	No	Yes

Table 1: Rank logs for the query "Healthy Snack Options" showing user clicks when items are presented alone and when presented with others, illustrating the dependency effect in listwise ranking.

(Vaswani et al., 2017a), particularly their success in natural language processing, offers a new perspective for addressing item ranking challenges. Transformers, known for their self-attention mechanism capable of capturing long-range dependencies within sequences (Bai et al., 2019), provide a groundbreaking framework for item ranking. However, adapting these models, originally designed for language tasks, to the specifics of item ranking entails addressing unique challenges inherent to this field.

In item ranking, models encounter more complex data structures, such as extended document embeddings or intricate item features, demanding a refined approach capable of handling high-dimensional data and understanding the nuanced dependencies at an item level (Burges et al., 2005). Moreover, the core of listwise ranking is the comprehension of the broader context and interrelationships among items in a list, requiring an attention mechanism designed to effectively capture these extended dependencies (Taylor et al., 2009).

This paper introduces TransformerRank, a transformative adaptation of the transformer model, meticulously tailored for the nuanced challenges of listwise item ranking. Deviating from traditional token-based transformer applications, TransformerRank is innovatively designed to process item-level data. It efficiently handles broad lists of candidate items, initially ranked by a transformer-based model, and then refines these lists using a sophisticated sliding window mechanism.

TransformerRank leverages a specialized attention mechanism that is sensitive to the global context within item lists. This advanced feature considerably enhances the accuracy and contextual relevance of the ranking system, allowing for direct list-level learning and optimization. Furthermore, the inclusion of a sliding window approach in TransformerRank is pivotal for fine-tuning the order of candidate items, ensuring an efficient and effective realization of listwise ranking results.

In summary, our contributions to the field of item ranking are as follows:

- **Proposing TransformerRank: A New Item Ranking Approach:** We introduce TransformerRank, an innovative model specifically designed for item ranking. This model is result-oriented, capable of directly learning from item lists and predicting outcomes, such as the number of clicks for a given list.
- **Efficient Sliding Window Optimization:** TransformerRank incorporates an efficient sliding window approach to fine-tune item rankings. While initial listwise ranking provides relevant results, the integration of sliding window optimization with TransformerRank specifically addresses efficiency in order adjustment.
- **State-of-the-Art Performance on Three Public Datasets:** Our approach marks a fundamental shift from traditional item ranking methods. TransformerRank has demonstrated state-of-the-art performance across three public datasets, showcasing its potential as a groundbreaking method in the field. This model not only addresses key limitations of existing ranking systems but also paves the way for further research and development in item ranking methodologies.

2 Related Work

2.1 Pointwise and Listwise Ranking Approaches

Early research in item ranking predominantly focused on pointwise and listwise methods. Pointwise approaches, treating item ranking as a classification or regression problem (Li et al., 2010), have been foundational in this field. However, they often fall short in capturing complex item dependencies and dynamic relationships. Listwise approaches, on the other hand, consider entire lists of items for ranking (Cao et al., 2007a). While offering a more holistic view, these methods sometimes struggle with scalability and intricacy in large datasets. Our work extends these traditional frameworks by introducing a transformative approach to pointwise ranking that integrates the depth of listwise analysis, enabling the modeling of intricate dependencies more effectively.

2.2 Transformer Models in Sequence Processing

The advent of transformer models, introduced by Vaswani et al. (Vaswani et al., 2017a), has reshaped the landscape of sequence processing. Their attention-based architecture excels at capturing complex sequential relationships, leading to significant advancements in tasks like natural language processing. Despite their success, the application of transformers in item ranking has not been fully explored. Our research fills this gap by adapting transformer models specifically for item ranking, leveraging their sophisticated attention mechanisms to provide an innovative solution for optimizing item sequences.

2.3 Sliding Window Techniques

Sliding window techniques have been instrumental in various optimization contexts, exemplified by their use in large-scale graph optimization by Cho et al. (Cho, 2016). These techniques offer precise, localized modeling, yielding context-aware solutions. We build upon this concept by integrating the sliding window technique with transformer models. This novel amalgamation allows for localized optimization while simultaneously capturing nuanced dependencies between items within each window. Our approach represents a methodological innovation, combining the strengths of sliding window techniques with the advanced capabilities of transformers for item ranking.

3 Methodology

In this section, we introduce a novel methodology for listwise item ranking that seamlessly integrates multiple stages of analysis and optimization. Initially, a transformer-based model conducts pointwise ranking, establishing an initial order of items based on individual relevance. Building upon this, our central innovation, *TransformerRank*, applies list-to-list learning to evaluate and enhance the collective arrangement of items. The pivotal aspect of our approach is the utilization of a sliding window technique, akin to beam search in sequence generation, which iteratively refines the order of items derived from the point-wise ranking (Wiseman and Rush, 2016). This technique enables dynamic reordering within the list, optimizing the overall ranking sequence to more accurately reflect inter-item dependencies and collective coherence.

3.1 Problem Formulation

Given a set of candidate items C and a query q, the task is to select and sequence a subset π that optimally satisfies the query. This task must account for the complex dependencies among items. The objective can be formulated as:

$$\pi^* = \underset{\pi \subseteq C, |\pi| = k}{\arg\max}\ Q(q, \pi) \tag{1}$$

In this equation, π^* is the optimal ordered subset of size k from candidates C, and $Q(q, \pi)$ quantifies the utility or relevance of subset π considering the query q and the dependencies among items in π.

Objective: Our goal is to create a ranking mechanism that not only identifies relevant items but also orders them in a way that maximizes overall utility, taking into account the interdependencies and effects of item combinations in the list.

4 Pointwise Scoring with Transformer-Based Models

In the first stage of our ranking process, we utilize a deep transformer-based model for pointwise scoring and item embedding generation (Vaswani et al., 2017b; Radford et al., 2018). The strength of such models lies in their ability to capture intricate relationships and contextual nuances, making them ideal for evaluating the relevance of items in relation to specific queries.

To adapt this model for our scoring task, we fine-tune it on a dataset comprising query-item pairs, each annotated with a relevance score. The fine-tuning process adjusts the model's weights to minimize the difference between the predicted scores and the actual annotated relevance scores. This objective is captured by the following loss function:

$$\mathcal{L} = \sum_{i=1}^{N} (s_i - \hat{s}_i)^2 \tag{2}$$

where $\mathcal{L}$ represents the loss function, N is the total number of query-item pairs in the training set, and s_i and $\hat{s}_i$ denote the true and predicted relevance scores for the i^{th} pair, respectively.

After fine-tuning, for a given query q and item d, the model generates embeddings, represented as e_q and e_d. The pointwise relevance score is then computed as:

$$s = \sigma(e_q^T W e_d + b) \tag{3}$$

In this equation, σ is the sigmoid activation function, which confines scores within the [0, 1] range. W is a weight matrix optimized during training, and b is a bias term.

This approach of utilizing transformer-based models for pointwise scoring and embedding generation is a crucial step in our ranking process, providing a robust foundation for the subsequent listwise ranking and optimization stages.

5 TransformerRank

TransformerRank leverages the state-of-the-art transformer architecture, renowned for its effectiveness in sequence modeling, to tackle the complexities of listwise ranking in information retrieval.

5.1 Model Formulation

Given a query q and a sequence of items $\pi = \{d_1, d_2, ..., d_N\}$, TransformerRank's objective is to optimize a listwise ranking score $Q(q, \pi)$. Unlike traditional methods that primarily focus on identifying the most relevant items independently, TransformerRank adopts a goal-oriented learning approach. It directly learns from the outcomes of how items are presented and interacted with in a list. For instance, it can discern user engagement, such as clicks or impressions, when a list is presented, without the need for calculating click-through rates or estimating relevance. This approach distinguishes TransformerRank from methods that decompose search logs into pointwise learning, fundamentally altering how it harnesses the original dataset.

Embeddings and Positional Encoding: Each item d_j in the sequence is converted into a high-dimensional representation via embeddings. These embeddings are derived from the transformer-based model fine-tuned in the initial stage of ranking. To capture the sequential nature of the item list, each embedding is augmented with a positional encoding:

$$e_{d_j} = \text{Embed}(d_j) + \text{Pos}(j)$$

This fusion of content and positional information equips TransformerRank with the capability to discern both the individual importance and the relative placement of items in the sequence. The positional encoding used here is based on sinusoidal functions (Vaswani et al., 2017b), defined as:

$$\text{Pos}(j)_{(2i)} = \sin\left(\frac{j}{10000^{2i/d}}\right)$$

$$\text{Pos}(j)_{(2i+1)} = \cos\left(\frac{j}{10000^{2i/d}}\right)$$

where j is the position and i is the dimension. This sinusoidal encoding facilitates the model's understanding of the position and distance between items in the list.

Longformer-Based Attention in TransformerRank: TransformerRank incorporates Longformer's sliding window attention mechanism, designed for efficient processing of longer sequences (Beltagy et al., 2020). This approach scales linearly with sequence length, making it well-suited for the extended item embeddings in listwise ranking. The attention computation for each position j in the item sequence is represented as:

$$\text{A}(e_{d_j}) = \text{Softmax}\left(\frac{e_{d_j} W_Q (S_{j,w} W_K)^T}{\sqrt{w}}\right) S_{j,w} W_V$$

Here, $S_{j,w}$ denotes the embeddings within a window of size w centered around j, and W_Q, W_K, W_V are the query, key, and value weight matrices, respectively. This localized attention allows TransformerRank to focus on a subset of items at a time, enhancing computational efficiency.

Incorporating Global Attention: Alongside the sliding window attention, TransformerRank integrates Longformer's global attention mechanism. This feature enables the model to attend to critical positions or items that have a broad contextual impact on the entire list. Global attention is particularly beneficial for identifying key items in listwise ranking that influence the perception of other items in the list:

$$\text{GlobalAttn}(e_{d_j}) = \text{Softmax}\left(e_{d_j} W_G E^T\right) E$$

In this formula, W_G is a weight matrix dedicated to global attention, and E represents the complete sequence of embeddings. Global attention provides a means to factor in the significance of specific items over the entire ranking.

Handling Extended Embeddings: By adapting Longformer's attention mechanisms, TransformerRank efficiently manages longer item embeddings. The combined use of local windowed attention and global attention offers a scalable solution to processing extensive sequences in listwise ranking. This integration ensures that TransformerRank maintains a balance between local item relationships and global list context, crucial for effective ranking in information retrieval tasks.

Ranking Score Prediction: TransformerRank, utilizing the Longformer architecture, processes the sequence of item embeddings to generate a comprehensive representation for ranking. This involves passing the Longformer-processed embeddings through position-wise feed-forward networks (FFN):

$$Q(q, \pi) = \sigma(\text{FFN}(\text{LongformerOutput}(\pi)))$$

The FFN, composed of layers with non-linear activation functions such as ReLU, is designed to distill complex interactions between items into a coherent ranking score. σ represents the activation function of the Score Predictor, which is a sigmoid function, transforming the output into a final ranking score. This adaptation ensures that TransformerRank is capable of handling the extended sequences typical of listwise ranking tasks.

Efficiency of Longformer's Attention Mechanism: TransformerRank effectively employs Longformer's attention mechanisms, combining local windowed attention with global attention, to efficiently process long sequences in listwise ranking. This dual attention approach enables the model to simultaneously focus on local item interactions and the broader global context, providing a comprehensive understanding of the entire item list. The integration of Longformer's scalable mechanisms is particularly advantageous for handling extended item embeddings, as it prevents a significant increase in computational complexity. This synergy between Longformer's nuanced attention focus and TransformerRank's sophisticated position-wise networks positions the model as a robust and advanced solution in the field of information retrieval, adept at managing the complexities of ranking extensive lists of items.

6 Enhanced Sliding Window Approach for Ranking

The Enhanced Sliding Window Approach presents an advanced technique for item ranking, which is illustrated in Figure 1. Commencing with an initial ranking determined by pointwise methods, the method utilizes a sliding window that begins its scan from the end of the list, effectively performing a backward scan. This backward movement targets the identification of items that have been potentially ranked lower but could have a significantly increased relevance when paired with specific surrounding items. This optimization is guided by the listwise function Q.

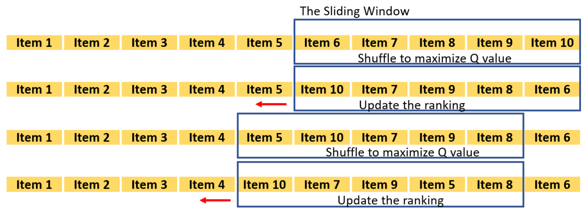

Figure 1: An example of our sliding window ranking workflow. Note that item 10 was mistakenly ranked lower initially but is ranked higher through the process.

During this backward traversal, each item within the window is assessed for its potential to enhance the localized ranking. If certain items are deemed more relevant in the context of their neighbors, an exhaustive search within the window determines the optimal order for those items. If an item doesn't fit this criterion, it's retained at its current position at the end of the window, which upholds the integrity of the previously determined high-relevance sequence and speeds up the scanning process. This iterative and detailed optimization across the entire list assures an improved and globally refined ranking.

Consider a scoring function $Q(q, \pi)$ quantifying the ranking quality within a window, where x represents the item permutation. As the window moves, permutations are appraised to identify the optimal permutation π^* maximizing $Q(q, \pi)$. Thus, for each permutation π in the window, the optimal solution adheres to:

$$Q(q, \pi^*) \geq Q(q, \pi), \quad \forall \pi \in I \tag{4}$$

Where I encapsulates all possible permutations in the window.

This iterative method guarantees local optima in each window, refining the global ranking by identifying intricate item dependencies.

This algorithm iteratively applies the window across the ranked items and finds the optimal permutation within each window. It integrates the scoring function Q to evaluate the permutations and gradually improves the overall ranking.

Our proposed algorithm also demonstrates notable computational advantages, aptly fitting real-world scenarios. When tasked with ranking G items and selecting k for display, traditional exhaustive permutation methods would involve an evaluation overhead of $C(G, k)$, with C representing the combinatorial function. In stark contrast, our strategy requires only $(G-m+1)C(m, m)$ assessments. Given the typical constraint where m is much smaller than both G and k (i.e., $m \ll G$ and

$m \ll k$), our method's computational footprint can be efficiently managed to meet deployment specifications, thus assuring prompt and accurate ranking.

7 Experiments

In this section, we evaluate the efficacy of TransformRank, complemented by our enhanced sliding window technique. We specifically assess the model's prowess in click prediction and relevancy ranking.

We delve into the foundational details supporting our approach, shedding light on the challenges addressed. Our ablation study further illustrates the efficiency and effectiveness of the proposed method.

For a thorough understanding, details such as feature selection for Learning to rank baselines (informed by LETOR 4.0 (Qin and Liu, 2013)), hyperparameter choices, model parameters, and reproducibility code are furnished in the supplementary materials.

The experiments were conducted on a Large Google Cloud Service cluster, equipped with four Tesla V100 GPUs.

7.1 Datasets

1. **TripClick:** This dataset is derived from the Trip medical database, a comprehensive source for clinical research publications. TripClick aims to provide a clearer understanding of real-world user behavior, especially concerning click-through. Featuring over 500,000 queries, relevance scores between queries and items are predefined, making it an ideal controlled environment for experimental testing in the domain of click prediction. The controlled nature of this dataset ensures reliable evaluations, diminishing the noise often encountered in real-world datasets (Team, 2020).

2. **Yahoo! Front Page Today Module User Click Log Dataset (Yahoo):** Encompassing millions of user interactions, this dataset is invaluable for the learning-to-rank community. The real-world nature of this dataset introduces challenges of noise and user biases, making it a rigorous testbed for any ranking model. The dataset is enriched with real-world query-item relevance labels derived from user interactions, making it indispensable for assessing real-world performance of learning-to-rank methodologies (Research, 2020b).

3. **ORCAS:** Produced by Microsoft Research, ORCAS is a benchmark dataset tailored for search and ranking challenges. It boasts over 500,000 queries and associated rankings, each enriched with extensive features and relevance judgments. The dataset's richness in terms of features and judgments positions it as a comprehensive platform for in-depth evaluations, facilitating robust comparisons and ensuring reproducibility in results (Research, 2020a).

7.2 Baselines

SVMrank (Joachims, 2006): A widely-used pairwise learning-to-rank method based on support vector machines.
RankNet (Burges, 2005): A pairwise neural network-based ranking model.
ListNet (Cao et al., 2007b): A listwise ranking approach using neural networks.
LambdaMART (Burges, 2010): An ensemble method that combines RankNet and MART, utilizing gradient boosting with decision trees for ranking.
BM25 (Robertson et al., 2009): A classical probabilistic-based ranking function in information retrieval.
TPRank (Qiao et al., 2019): A pointwise ranking approach leveraging the BERT transformer for direct document ranking.
coCondenser (Gao and Callan, 2021): Introduces a novel method for dense passage retrieval using co-attention and contrastive learning.
Uni-Retriever (Zhang et al., 2022): A model that utilizes pre-training techniques for enhancing document ranking performance.
TransformerRank with enhanced sliding window (TSrank): Our proposed method for listwise item ranking with transformers and sliding window optimization.

7.3 Evaluation Metrics

NDCG@10: A metric assessing ranking quality using graded relevance (Järvelin and Kekäläinen, 2002).
MAP: Represents the mean of the average precision scores across queries (Manning et al., 2008).
Precision@10: Denotes the proportion of top-k recommendations that are relevant (Manning et al.,

2008).

Recall@10: Gauges the fraction of the total relevant items found within the top-k recommendations (Manning et al., 2008).

MRR: Computes the average reciprocal rank of the initial correct result (Voorhees, 1999).

7.4 Experimental Procedure

To maintain consistency across our experiments, all datasets were standardized, ensuring uniform feature scaling. We divided each dataset into training, validation, and test sets, following an 80-10-10 split. This partitioning was strategically chosen to optimize the performance of our models while ensuring robust validation and testing.

The training set played a crucial role in fine-tuning models that rely on pre-trained architectures, such as BERT. This process was vital for adapting these models to our specific ranking tasks. The validation set was instrumental in optimizing hyperparameters, a step essential for achieving the best model performance.

For model assessment, we utilized the test set, applying a range of predefined metrics to evaluate the models' effectiveness. An initial retrieval phase was conducted using Solr's TF-IDF algorithm. In this phase, we aimed to retrieve the top 400 results, a number slightly above the average items retrievable across the three datasets. This approach was designed to ensure comprehensive coverage of potential items while maintaining a manageable dataset size for analysis.

Following this initial retrieval, we conducted an in-depth comparative analysis, contrasting TransformerRank's performance against established baseline methods. This comparison was crucial for demonstrating the efficacy and advantages of our approach.

Further details, including an in-depth description of TransformerRank's model settings, links to the datasets used, and additional relevant information, are comprehensively presented in the supplementary material.

7.5 Experimental Results

In the evaluation of various ranking methods across three prominent datasets, we observed distinctive performance differences in Table 2. For the TripClick dataset, the proposed TSrank method showcased a marked improvement, scoring highest in all metrics, notably with an NDCG@10 of 0.69, a MAP score of 0.65, Precision@10 at 0.70, Recall@10 at 0.72, and an MRR of 0.76. All these scores were found to be statistically significant, surpassing the performance of traditional methods like SVMrank, RankNet, and even more contemporary approaches like Uni-Retriever.

Similarly, in the Yahoo dataset, TSrank outperformed the rest with significant leads in all metrics: NDCG@10 (0.67), MAP (0.66), Precision@10 (0.69), Recall@10 (0.70), and MRR (0.75). This trend continued in the ORCAS dataset, with TSrank registering the top scores, with NDCG@10 at 0.68, MAP at 0.66, Precision@10 and Recall@10 at 0.68 and 0.70, respectively, and MRR at 0.73. While several methods such as Uni-Retriever and coCondenser produced commendable results, none achieved the consistently superior performance of TSrank across all datasets and metrics. On average, the proposed TSrank approach exhibited a 5%-12% improvement over the previous state-of-the-art methods. These findings underscore the effectiveness of the TSrank method, positioning it as a prime choice for addressing listwise item ranking challenges.

7.6 Ablation Studies

Our ablation study focuses on exploring the concept of locality in listwise ranking, specifically investigating the impact of sliding window size using the Yahoo! click dataset. This analysis helps us understand how varying window sizes influence ranking performance. Additionally, we extend our examination to a relevance-based dataset to determine how TransformerRank performs when documents are individually annotated by humans based on relevance, providing further insights into the model's versatility in different ranking scenarios.

From Figure 2, we identified that an increase in window size enhances performance but also incurs higher computational costs. Notably, the performance gain plateaus when the window size reaches 6, suggesting that this is an optimal balance between effectiveness and efficiency.

The study also underscores the significance of item dependencies and their localized nature within ranking datasets. This insight is particularly relevant when considering human attention patterns and their impact on item connections.

The efficacy of TransformerRank is further highlighted in its performance on the MS-MARCO document ranking dataset, as detailed in Table 3. This dataset shares similar characteristics with ORCAS, offering a relevant context for evaluating relevance-

	TripClick					Yahoo					ORCAS				
Method	NDCG@10	MAP	Precision@10	Recall@10	MRR	NDCG@10	MAP	Precision@10	Recall@10	MRR	NDCG@10	MAP	Precision@10	Recall@10	MRR
SVMrank	0.52	0.47	0.50	0.53	0.58	0.48	0.45	0.47	0.49	0.55	0.45	0.42	0.44	0.46	0.53
RankNet	0.53	0.49	0.52	0.55	0.60	0.50	0.47	0.49	0.51	0.57	0.47	0.44	0.46	0.48	0.55
ListNet	0.54	0.50	0.53	0.56	0.61	0.51	0.48	0.50	0.53	0.58	0.49	0.46	0.48	0.50	0.56
LambdaMART	0.57	0.54	0.56	0.59	0.64	0.54	0.52	0.53	0.56	0.61	0.53	0.50	0.52	0.54	0.59
BM25	0.55	0.52	0.54	0.57	0.62	0.53	0.50	0.51	0.54	0.59	0.51	0.48	0.50	0.52	0.57
TPRank	0.58	0.55	0.57	0.60	0.65	0.56	0.54	0.55	0.58	0.63	0.55	0.52	0.54	0.56	0.61
coCondenser	0.60	0.58	0.61	0.63	0.67	0.59	0.58	0.60	0.62	0.66	0.59	0.56	0.58	0.60	0.65
Uni-Retriever	0.62	0.59	0.63	0.65	0.68	0.61	0.60	0.62	0.63	0.68	0.61	0.59	0.61	0.63	0.66
TSrank	**0.69***	**0.65***	**0.70***	**0.72***	**0.76***	**0.67***	**0.66***	**0.69***	**0.70***	**0.75***	**0.68***	**0.66***	**0.68***	**0.70***	**0.73***

Table 2: Performance of baselines and proposed methods across different datasets. An asterisk (*) denotes statistically significant improvement, and bold values highlight the superior performance of TSrank.

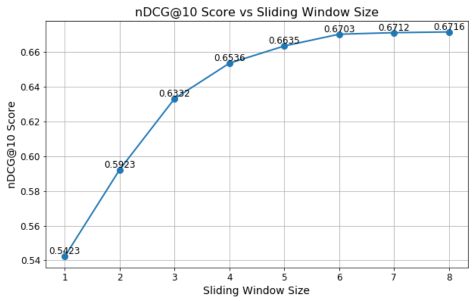

Figure 2: Variation in ranking performance (nDCG@10) as the sliding window size increases. Optimal performance is observed at $m = 6$.

based ranking scenarios (Nguyen et al., 2016). The results on MS-MARCO provide valuable insights into TransformerRank's adaptability across different annotation methods and corpora. Although its performance advantage is slightly less pronounced on MS-MARCO compared to ORCAS, TransformerRank consistently surpasses other methods. This performance underlines its proficiency in identifying document dependencies and relevance. Particularly in scenarios where items bear contextual interconnections, TransformerRank's performance is exemplary, showcasing its wide applicability and flexibility in various ranking environments.

Method	NDCG@10	MRR
SVMrank	0.558	0.388
RankNet	0.564	0.373
ListNet	0.562	0.367
LambdaMART	0.558	0.353
BM25	0.546	0.314
TPRank	0.601	0.446
coCondenser	0.609	0.448
Uni-Retriever	0.606	0.439
TSRank	**0.629 (+3.3%)**	**0.471 (+5.1%)**

Table 3: Performance of various baseline methods and TransformerRank on the MS-MARCO dataset. Metrics include NDCG@10 and MRR.

8 Conclusion

In this paper, we introduced TransformerRank, an innovative model that redefines the approach to listwise item ranking. Engineered to leverage the strengths of transformer architectures, TransformerRank is uniquely capable of direct list-level learning, adeptly capturing complex dependencies among items. Our empirical evaluations have demonstrated its significant edge over traditional ranking models. TransformerRank excels in interpreting the nuanced interactions and contextual information within item lists, a capability that sets a new standard in the field.

The essence of TransformerRank lies in its innovative approach to listwise ranking, facilitating a deeper understanding of item relationships beyond the conventional item-by-item analysis. This is complemented by the integration of sliding window optimization, which enhances the model's ability to process lengthy sequences of items efficiently while maintaining the integrity of listwise analysis. Additionally, the adaptability of TransformerRank ensures its applicability across various dynamic ranking scenarios, making it a versatile tool for modern ranking challenges.

Looking forward, the potential applications of TransformerRank extend beyond item ranking, offering promising opportunities in other complex data processing tasks. Future research could focus on enhancing the model's scalability and computational efficiency, further broadening its utility in handling more extensive and complex datasets. TransformerRank's success in our studies underscores the transformative potential of advanced transformer models in the realm of listwise item ranking, paving the way for continued innovation in this field.

9 Ethical Considerations

In adherence to the Responsible NLP Research Checklist, this study upholds the highest ethical standards. We have ensured meticulous citation of all utilized artifacts, maintaining respect for intellectual property rights. Transparency and reproducibility are key pillars of our research, and we provide comprehensive details for result replication. The rights to use this paper are granted for academic purposes, facilitating scholarly discourse and progression in the field.

Furthermore, the development and application of the TransformerRank model comply with ethical norms, including the principles of honesty, fairness, and respect for the rights of others as outlined in the ACM Code of Ethics. Our commitment to ethical research extends to responsible data and AI technology use, ensuring positive contributions to machine learning and information retrieval fields.

The code and related materials for this study will be available on GitHub and the Hugging Face platform, promoting collaborative and responsible scientific practice.

9.1 Potential Risks

This research, while advancing the field of item ranking with TransformerRank, presents potential risks that must be acknowledged. The primary risk involves the misuse of advanced ranking algorithms, potentially leading to biased or unfair outcomes if the underlying data or implementation is not handled with care. Additionally, there is a risk of over-reliance on automated ranking systems, which might overlook subtle nuances that require human judgment. We emphasize the importance of using TransformerRank responsibly, ensuring that its application in various domains is guided by ethical considerations and fairness. To mitigate these risks, we advocate for continuous monitoring and evaluation of the model's impact on diverse datasets and real-world scenarios.

10 Limitations

This study's primary limitation is the reliance on existing datasets, which may not fully represent the diverse and evolving nature of real-world item ranking scenarios. TransformerRank, while adept at handling large datasets, may encounter performance variability due to data quality and inherent biases. Additionally, the computational efficiency of the model, particularly when processing extremely large datasets, is an area for future optimization. The model's generalizability across various languages and cultural contexts also remains to be rigorously tested, as these factors can significantly influence ranking dynamics. Finally, while TransformerRank demonstrates promising results, its application in real-world systems may require further tuning to align with specific domain requirements and user behaviors.

References

S. Bai et al. 2019. Transformers for time-series data. *Neural Computing and Applications*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Christopher JC Burges. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM.

Christopher JC Burges. 2010. From ranknet to lambdamart: An overview. In *Learning*, volume 11, pages 23–581.

C.J.C. Burges et al. 2005. Learning to rank using gradient descent. In *ICML*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007a. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007b. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Minock Cho. 2016. A sliding window technique for incremental large-scale graph optimization. In *ICRA*, page ...

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.

Ping Li, Christopher J.C. Burges, and Qiang Wu. 2010. A boosted tree method for pointwise learning-to-rank models. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*.

X. Li et al. 2014. Scalability of listwise learning-to-rank methods. *Information Retrieval Journal*.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*, 1(8).

Microsoft Research. 2020a. Orcas. https://microsoft.github.io/msmarco/ORCAS.html.

Yahoo Research. 2020b. Yahoo! front page today module user click log dataset. https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=49.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

M. Taylor et al. 2009. Softrank: Optimizing non-smooth rank metrics. *WSDM*.

Trip Database Team. 2020. Tripclick. https://tripdatabase.github.io/tripclick/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. *Advances in neural information processing systems*, 30.

Ellen M Voorhees. 1999. The trec-8 question answering track report. In *TREC*, volume 99, pages 77–82.

Y. Wang et al. 2020. Adaptive learning-to-rank models for dynamic environments. *TOIS*.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*.

Hao Zhang. 2022. Language model decomposition: Quantifying the dependency and correlation of language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2508–2517, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianjin Zhang, Zheng Liu, Weihao Han, Shitao Xiao, Ruicheng Zheng, Yingxia Shao, Hao Sun, Hanqing Zhu, Premkumar Srinivasan, Weiwei Deng, et al. 2022. Uni-retriever: Towards learning the unified embedding based retriever in bing sponsored search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4493–4501.

L. Zhao et al. 2021. Real-time search result ranking with deep learning. In *SIGIR*.