www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(2), May-Aug 2024, 95-123. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2024.v15i2.465

A Transformer-Based Multi-Domain Recommender System for E-commerce

Victor Giovanni Morales-Murillo[1], David Pinto[1], Fernando Perez-Tellez[2], Franco Rojas-Lopez[3]

[1]Language & Knowledge Engineering (LKE), Benemérita Universidad Autónoma de Puebla, Puebla, México.
[2]School of Enterprise Computing and Digital Transformation, Technological University Dublin, Dublin, Ireland.
[3]Universidad Politécnica Metropolitana de Puebla, Puebla, México.
E-mails:vg055@hotmail.com, dpinto@cs.buap.mx, Fernando.PerezTellez@tudublin.ie, franco.rojas@metropoli.edu.mx

Abstract. Recommender systems are one of the most critical applications of AI, data science, and advanced analytics techniques because it has become integrated into our daily lives. Additionally, it serves as a powerful tool for making informed, effective, and efficient decisions and choices across a wide range of items. However, traditional techniques such as content-based and collaborative filtering often fail to consider the dynamic and short-term preferences of users when generating recommendations. To address this limitation, this research focuses on a session-based recommendation task using an XLNet transformer with various training strategies based on language modeling. Moreover, a dataset containing 102 million reviews of Amazon products was pre-processed to create two new datasets, one for a single domain and another for multi-domain data. A comparison between a GRU and the training strategies of XLNet reveals that the best training strategy achieves a 136.23% improvement in NDCG@20 and a 95.69% increase in Recall@20 for multi-domain data. In a single domain, it achieves a 168.81% improvement in NDCG@20 and a 25% increase in Recall@10.

Keywords: Recommender System, Session-Based recommendation, Transformer, NLP, E-commerce.

Article Info
Received May 23, 2024
Accepted Jun 1, 2024

1 Introduction

Recommender system (RS) stands as one of the most vital applications of artificial intelligence (AI), data science, and advanced analytics techniques. Its integration into our daily lives spans across work, business operations, study, entertainment, and socialization (Wang, Pasi, Hu, & Cao, 2020). Moreover, the recommendation system serves as a potent tool for making well-informed, effective, and efficient decisions across a vast array of items, including commercial products, movies, playlists, points of interest, hotels, restaurants, friends, travels, careers, and more (Wang et al., 2021). Major international companies such as Amazon, Spotify, and Netflix have integrated these systems into their core services. By addressing information overload, enhancing user experiences, and reducing churn rates, RSs significantly impact company profits.

Traditionally, techniques in recommender systems have revolved around content filtering, which computes similar items based on their metadata, collaborative filtering, which identifies similar users using a set of rankings, and hybrid approaches that combine both techniques to optimize RS performance. State-of-the-art research has identified main issues such as the cold start problem, scalability, and lack of novelty in recommendations, all of which are addressed by hybrid approaches (Çano & Morisio, 2017). However, these techniques have typically relied on all historical user-item interactions, including clicks, purchases, views, etc., to learn long-term and static user preferences on items, making the underlying assumption that all historical interactions are equally important to the user's current preferences (Wang et al., 2021). This assumption overlooks two key points: firstly, the importance of short-term recent preferences and the time-sensitive context in user choices, and secondly, the dynamic nature of user preferences, which evolve over time. Moreover, it's noted that only a small number of historical interactions represent the most recent interactions of users, which contain the short-term recent preferences of users (Jannach, Ludewig, & Lerche, 2017).

In recent years, a new paradigm of recommender systems, known as session-based recommender systems (SBRS), has emerged to address the short-term and dynamic preferences of users, aiming to generate more timely and accurate recommendations (Wang et al., 2019). This paradigm is a subarea of sequential recommender systems (SRS), making SBRS closely related to SRS in terms

of input, output, and recommendation mechanism. The primary objective of SBRS is to learn the dependencies embedded in sequences or sessions to infer users' dynamic preferences. In an SBRS, user preferences are learned from sessions, each comprising multiple user-item interactions occurring within a continuous period of time. By considering each session as the fundamental input unit, SBRS can capture both a user's short-term preferences from their recent sessions and preference dynamics, reflecting changes in preferences from one session to another, thereby enabling more precise and timely recommendations.

Recent advancements in sequential and session-based recommendation systems have been driven by improvements in model architecture and pretraining techniques originating from the field of Natural Language Processing (NLP), particularly Transformer architectures. These architectures have supplanted convolutional and recurrent neural networks in language modeling tasks due to their efficient parallel training, scalability with training data and model size, and effectiveness in modeling long-range sequences. Moreover, Transformers have facilitated the development of higher-capacity models and introduced data augmentation and training techniques that significantly enhance the efficacy of sequential recommendation systems. The sequential processing of user interactions in sequential and session recommendations bears resemblance to the language modeling (LM) task, leading to the adaptation of many Transformer architectures from NLP, such as the Transformers4Rec library (de Souza Pereira Moreira, Rabhi, Lee, Ak, & Oldridge, 2021). This open-source library, built upon HuggingFace's Transformers library, shares the goal of extending the advances of NLP-based Transformers to the recommender system community, making these advancements readily accessible for sequential and session-based recommendation tasks.

Therefore, this study constructs a Transformer-based multi-domain recommender system for e-commerce utilizing the latest NLP advancements such as Transformers4Rec, facilitating an empirical analysis of session-based recommendation on domain-specific and multi-domain data sourced from the Amazon review dataset (Ni, Li, & McAuley, 2019). Various training techniques, including Causal LM (CLM), Masked LM (MLM), Permutation LM (PLM), and Replacement Token Detection (RTD) by a Transformer architecture known as XLNet (Yang et al., 2020), are compared within a Gated Recurrent Unit (GRU). This document is structured as follows: Section 2 introduces the background on session-based recommender systems and Transformers. Section 3 discusses related work. Section 4 elaborates on the proposed methodology. Section 5 presents the results, and Section 6 concludes the study.

2 Background

The background section provides an overview of session-based recommender systems and language modeling based on Transformers, highlighting their fundamentals and significance in analyzing various training techniques within these architectures.

2.1 Session-based recommender system

Session-based recommender systems focus on understanding and predicting user preferences based on sequential interactions or sessions. These systems aim to capture the dynamic nature of user behavior by considering the sequence of actions taken within a session. By leveraging this sequential data, session-based recommender systems can offer personalized recommendations that align with users' immediate preferences and interests.

The difference between session data and sequence data lies in their structure and organization. A session refers to a finite list of interactions, which can be either ordered or unordered. When interactions within a session are arranged chronologically, it is termed as an ordered session. Conversely, if the interactions are not arranged chronologically, it is referred to as an unordered session. Multiple sessions can occur at different times, collectively forming a user's session data. These sessions are delineated by various boundaries, with potentially non-identical time intervals between them. For instance, consider Figure 1, which illustrates an example of user session data. In this scenario, three sessions are depicted, each separated by boundaries occurring at intervals of 2 weeks and 4 weeks over time. Session 1 comprises three user-item interactions, while sessions 2 and 3 each contain two user-item interactions. Despite variations in the number of interactions, all sessions contribute to the general user's session data, providing insights into their behavior and preferences over time.

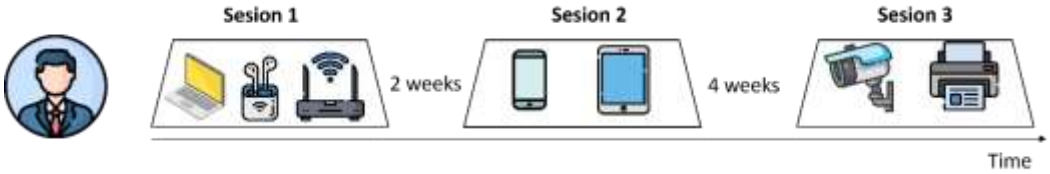

Figure 1. This is an example of user's session data which contains three sessions divided by boundaries of 2 weeks and 3 weeks over times.

A sequence refers to a list of historical items arranged in a specific order, such as a sequence of item IDs. In the context of user data, a sequence typically represents a series of interactions or events, with each item in the sequence indicating a specific action taken by the user over time. Unlike sessions, sequences do not have multiple boundaries and are characterized by a single continuous sequence of events. For example, consider Figure 2, which illustrates an example of user sequence data. In this depiction, the user's sequence data consists of a single sequence comprising five movies watched in chronological order. Each movie in the sequence represents a distinct event or interaction, forming a clear and ordered representation of the user's viewing history. Unlike sessions, which may contain multiple sessions separated by boundaries, a sequence encapsulates all interactions within a single continuous sequence, providing a comprehensive overview of the user's behavior.

Figure 2. This is an example of user's sequence data which contains one sequence of five movies watched in chronological order by the user.

Indeed, systems that utilize unordered session data typically rely on dependencies based on co-occurrence. In this context, co-occurrence refers to the occurrence of items together within the same session, irrespective of their order. These systems analyze patterns of item co-occurrence to identify associations and make recommendations based on items frequently observed together in sessions.

On the other hand, systems that utilize ordered session or sequential data leverage sequential dependencies. Here, the order of interactions within sessions is crucial, as it reflects the temporal sequence of user actions. By considering the sequential order of interactions, these systems capture dependencies between items based on the sequence in which they were encountered. This approach allows for the modeling of user preferences and behavior over time, enabling the generation of personalized and contextually relevant recommendations.

The goal of a session-based recommender system is to predict the unknown part of a session, which could be an item or a batch of items, or even the future session, such as the next-basket. This task is defined by five key entities:

- Users, along with their properties.
- Items, along with their properties.
- Actions, which encompass users' interactions with items, such as clicks, views, and purchases.
- Interactions, represented as triplets of [user, action, item], capturing the relationship between users, actions, and items.
- Sessions, each characterized by properties such as session duration, internal order of interactions, action types, user information, and data structure.

Various leading approaches to session-based recommender systems exist, including conventional methods, latent representation techniques, and deep neural network approaches. These approaches are discussed in detail in the literature, providing insights into their strengths, weaknesses, and applications in addressing the challenges of session-based recommendation (Wang et al., 2021) (Wang et al., 2019).

2.1.1 Conventional approaches

Data mining or machine learning techniques are employed to capture the dependencies inherent in session data for session-based recommendations. The following techniques are commonly utilized for this purpose:

- Pattern/rule mining.
- K nearest neighbour.
- Markov chain.
- Generative probabilistic model.

2.1.2 Latent representation based approaches

Low-dimensional latent representations are generated for each interaction within sessions using shallow models. These learned representations encode informative dependencies between interactions, facilitating the generation of subsequent session-based recommendations. The following methods are commonly employed for this purpose:

- Latent factor model.
- Distributed representation.

2.1.3 Deep neural network based approaches

These approaches leverage the capability to model complex intra- and inter-session dependencies for recommendations. The classification for these techniques is divided into: (1) Basic deep neural networks, where a fundamental neural network architecture such as Recurrent Neural Networks (RNN) is utilized. (2) Advanced models, which employ sophisticated mechanisms or models such as attention models.

- Basic deep neural networks:
    - Recurrent neural networks.
    - Multilayer perceptron networks.
    - Convolutional neural networks.
    - Graph neural networks.
- Advanced models:
    - Attention model.
    - Memory networks.
    - Mixture model.
    - Generative model.
    - Reinforcement learning.

2.2 Transformers

In this study, our focus is on advanced models, specifically attention models known as Transformers, which have gained significant popularity in the Natural Language Processing (NLP) community. Proposed by Vaswani et al. in 2017 (Vaswani et al., 2023), this neural architecture has demonstrated remarkable performance across various NLP tasks. At the core of Transformers lies a self-attention mechanism, which emphasizes the contextual relationships among words or tokens in a sequence. One of the key advantages of Transformers is their ability to handle long-term dependencies effectively. This is achieved through the self-attention mechanism, which enables the model to assess the importance of different words or tokens in the sequence while processing the entire sequence simultaneously. This mechanism, also referred to as scaled dot-product attention, allows the model to compute attention weights for each word/token based on its relationships with other words/tokens in the input sequence. By leveraging attention, the model can focus on relevant parts of the input during both the encoding and decoding stages, enhancing its ability to capture intricate dependencies and produce accurate predictions.

These architectures have been categorized based on their pre-training tasks, which are essential for learning universal language representations. There are three main types of learning approaches utilized.

- Supervised learning**:** In this approach, the model learns from labeled data, where each input is associated with a corresponding output label. Supervised learning is commonly used in tasks where the ground truth is available during training, allowing the model to learn to predict the correct output.
- Unsupervised learning: Unsupervised learning involves training the model on unlabeled data, where the objective is to learn patterns and structure from the data without explicit guidance or labels. This approach is often used to discover underlying patterns or representations in the data without the need for labeled examples.
- Self-supervised learning: Self-supervised learning falls under the broader category of unsupervised learning but involves creating supervised-like tasks from the input data itself. Instead of relying on external labels, the model generates its own supervision signals from the input data. This approach has gained popularity due to its ability to leverage large amounts of unlabeled data effectively.

Pre-training tasks often use the self-supervised learning on the transformers, and these tasks are introduced below.

2.2.1 Masked Language Modeling (MLM)

Multiple NLP tasks have achieved remarkable success by pre-training text encoders to learn from bidirectional contexts. One of the most prominent pre-training approaches is Masked Language Modeling, known for its conceptual simplicity and empirical effectiveness (Meng et al., 2023). This approach involves masking a portion of tokens from input sentences and then training the model to predict the masked tokens based on the surrounding context (Qiu et al., 2020).

2.2.2 Causal Language Modeling (CLM)

Causal Language Modeling is designed to predict the next element in a sequence in an autoregressive manner, making it one of the fundamental applications of the Transformer model (Wu & Varshney, 2023). This pre-training approach is commonly employed in text generation tasks, where the model considers only the past context and not the future context when generating predictions, as demonstrated by models like GPT-2.

2.2.3 Permuted Language Modeling (PLM)

In Permutation Language Modeling, a permutation is randomly sampled from all possible permutations of the sequence. Subsequently, certain tokens within the permuted sequence are selected as targets, and the model is trained to predict these targets based on the remaining tokens and their natural positions within the sequence. It's important to note that this permutation does not alter the natural positions of the tokens in the sequence; rather, it solely determines the order in which token predictions are made (Qiu et al., 2020).

2.2.4 Replaced Token Detection (RTD)

Replacement Token Detection is a discriminative pre-training technique that involves training a generator to produce replaced tokens and a discriminator to differentiate between real and replaced tokens. This method enhances pre-training efficiency by decreasing the computational overhead in the head compared to previous masked language modeling approaches (Lu et al., 2023).

3 Related works

The study titled *BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer* (Sun et al., 2019) introduced a sequential recommendation model leveraging deep bidirectional self-attention to model user behavior sequences. The authors employed the Cloze objective, commonly used in language understanding tasks, to sequential recommendation. This objective involves predicting randomly masked items in the sequence by considering both their left and right context jointly. Furthermore, the authors conducted experiments on four datasets and demonstrated that their model outperformed other sequential models in terms of recommendation performance.

The research titled *Exploiting deep transformer models in textual review-based recommender systems* (Gheewala, Xu, Yeom, & Maqsood, 2024) underscores that textual reviews contain pertinent information that can effectively infer user preferences over items. The study highlights that deep learning models better capture user-item interactions from textual reviews compared to traditional recommendation approaches, thereby enhancing predictive performance. The authors employed and analyzed deep transformer models for review-based recommender systems, noting that deep transformer models can extract interpretable and relevant user and item representations more effectively than traditional deep learning networks. The results demonstrate that the best-performing deep transformer model achieved a maximum relative improvement on RMSE of 4.6% and a MAE of 7.4% with Amazon electronics compared to the best outcome from traditional deep learning networks.

The study titled *Transformers4Rec: Bridging the Gap between NLP and Sequential/Session-Based Recommendation* (de Souza Pereira Moreira et al., 2021) addressed the growing disparity between Natural Language Processing (NLP) and sequential/session-based recommendation fields. The authors developed Transformers4Rec, an open-source library aimed at bridging this gap by providing various transformer architectures tailored for sequential and session-based recommendations. They achieved promising results, particularly in next-click prediction for user sessions, despite the sequence lengths being much shorter than those commonly encountered in NLP tasks. Additionally, their experiments demonstrated that the superior architectures yielded improved performance across two e-commerce datasets, while maintaining similar performance to baseline models on two news datasets.

4 Methods

This section outlines the methodology employed to conduct an analysis on a Transformer-Based Multi-Domain Recommender System for E-commerce, encompassing two experiments. The first experiment utilized a dataset comprising products from a single category on Amazon, while the second experiment employed a dataset featuring products from 15 different categories on Amazon.

4.1 Data Analysis

We utilized a dataset of Amazon review data sourced from (Ni et al., 2019), comprising 233.1 million reviews collected from May 1996 to October 2018. This dataset offers various subsets, including complete review data, rating-only data, 5-score data, per-category data, and smaller subsets designed for experimentation such as k-scores and ratings only. From the per-category data subset of complete review data, we selected 15 subsets for our analysis. Table 1 displays the details of these subsets, which collectively contain 102,395,764 reviews and 5,993,235 metadata entries for products.

Table 1. Per-category data on Amazon products.

No	Category	Reviews	Products
1	All_Beauty	371,345	32,992
2	AMAZON_FASHION	883,636	186,637
3	Musical_Instruments	1,512,530	120,400
4	Industrial_and_Scientific	1,758,333	167,524
5	Video_Games	2,565,349	84,893
6	Grocery_and_Gourmet_Food	5,074,160	287,209
7	Patio_Lawn_and_Garden	5,236,058	279,697
8	Office_Products	5,581,313	315,644
9	Pet_Supplies	6,542,483	206,141
10	Toys_and_Games	8,201,231	634,414
11	Movies_and_TV	8,765,568	203,970
12	Cell_Phones_and_Accessories	10,063,255	590,269
13	Sports_and_Outdoors	12,980,837	962,876
14	Electronics	20,994,353	786,868
15	Home_and_Kitchen	21,928,568	1,301,225
	Total	102,395,764	5,993,235

Review data contains the next fields:

- reviewerID: ID of the reviewer.
- asin: ID of the product.
- reviewerName: name of the reviewer.
- vote: helpful votes of the review.
- verify: if review is verified.
- style: a dictionary of the product metadata.
- reviewText: text of the review.
- overall: rating of the product.
- summary: summary of the review.
- unixReviewTime: time of the review (unix time).
- reviewTime: time of the review (raw).
- image: images that users post after they have received the product.

Metadata of products contains the next fields:

- asin: ID of the product.
- asin: title: name of the product.
- feature: bullet-point format features of the product.
- description: description of the product.
- price: price in US dollars (at time of crawl).
- imageURL: url of the product image.
- related: related products (also bought, also viewed, bought together, buy after viewing)
- salesRank: sales rank information.
- brand: brand name.
- categories: list of categories the product belongs to.
- tech1: the first technical detail table of the product.
- tech2: the second technical detail table of the product.
- similar: similar product table.

Data preprocessing is crucial to adapt the dataset for session-based recommendation tasks, given that it contains information related to content-based filtering, collaborative filtering, and hybrid approaches, which are traditional recommendation techniques. The objective is to incorporate additional characteristics that complement the product IDs in the sessions, such as overall rating, verification status, price, category, brand, etc. Additionally, it is necessary to group reviews by dates to create user sessions.

In the first step of preprocessing, the subsets are processed to remove duplicate instances and drop columns not relevant to reviews and products data, including *reviewerName*, *reviewText*, *summary*, *vote*, *style*, *image*, *title*, *feature*, *date*, *imageURL*, *imageURLHighRes*, *description*, *also_view*, *also_buy*, *fit*, *details*, *similar_item*, *tech1*, and *tech2*. Furthermore, we concatenate the *reviewerID* and *unixReviewTime* to form the session ID, indicating that a session comprises all purchases made by a user on the same date.

During this initial step, we compute and save the number of sessions by date and the number of categories associated with each date. We then select a week with the highest number of sessions, encompassing all 15 categories. This process incurs a high computational cost due to the large number of reviews. The table below presents the results for three weeks, with week 3 containing the largest number of reviews, totaling 246,272 instances. To mitigate computational costs, we will utilize instances only from the period between 02/01/2017 and 08/01/2027. Besides, Week 3 stands out with the highest number of instances compared to the other weeks. This suggests that there may be significant user activity or product interactions during this period, making it an important time frame to analyze for our session-based recommendation task. And this approach offers the advantage of reducing processing time while still capturing relevant data.

Table 2. Analysis of instances by week.

Week 1		Week 2		Week3	
Date	**Instances**	**Date**	**Instances**	**Date**	**Instances**
26/12/2014	28,799	19/01/2016	35,690	02/01/2017	36,811
27/12/2014	27,031	20/01/2016	44,785	03/01/2017	42,204
28/12/2014	29,662	21/01/2016	35,769	04/01/2017	36,687
29/12/2014	40,000	22/01/2016	35,346	05/01/2017	37,025
30/12/2014	30,742	23/01/2016	29,830	06/01/2017	31,271
31/12/2014	29,930	24/01/2016	29,493	07/01/2017	31,385
01/12/2014	33,587	25/01/2016	34,529	08/01/2017	30,889
Total	219,751	-	245,442	-	246,272

In the second step of preprocessing, we processed the sets of products by merging them with the corresponding set of reviews related to the same category. Additionally, we extracted the category information from the "rank" field using regular expressions. Sessions with only one interaction were removed to ensure the dataset's quality. Each preprocessed set was saved as a JSON file.

Finally, all sets were combined to create a unified dataset encompassing all categories and preprocessed sessions. This consolidated dataset will serve as the foundation for further analysis and model development in our session-based recommendation task.

4.2 Experiments

In this section, we outlined the utilization of Transformers4Rec (de Souza Pereira Moreira et al., 2021) to devise training strategies. Transformers4Rec serves as an end-to-end recommendation system framework, encompassing data preprocessing, model training, and evaluation stages. Developed in Python, the framework leverages PyTorch and Hugging Face Transformers to facilitate efficient implementation and experimentation with transformer-based models for recommendation tasks.

The NVIDIA NVTabular library, closely related to Transformers4Rec, offers GPU-accelerated capabilities for preprocessing tasks. This tool supports feature engineering techniques tailored for session-based recommendation, including operations for grouping time-sorted interactions by user or session, as well as truncating sequences to retain the first or last N interactions. Additionally, NVTabular enables the saving of preprocessed data in a structured and queryable Parquet format. One of the key advantages of NVTabular is its ability to expedite training and evaluation processes by loading data directly onto the GPU. Furthermore, the library provides a configuration file that allows users to specify which features should be treated as continuous or categorical characteristics, enhancing flexibility and customization for model training.

An incremental training and evaluation approach is implemented, wherein a sliding window with a single time unit, such as a day or hour, is utilized in temporal order to train the model incrementally. This involves fine-tuning the parameters of a model that has already been trained using past data. The sessions are divided into time windows, denoted as T, with each window having a length of one day. Evaluation is conducted for each subsequent time window Ti + 1, where i to n-1, using sessions from past time windows for training [T1, ..., Ti]. Furthermore, the sessions within each time window are split into a 50:50 ratio between validation and test sets. The validation sets from each time window are utilized for hyperparameter tuning, while the test sets are employed for reporting metrics. Finally, the reported metrics are the averages across all time windows.

Evaluation in session-based recommendation is conducted using traditional Top-N ranking metrics such as NDCG@N and Recall@N. Normalized Discounted Cumulative Gain (NDCG) measures the effectiveness of a ranking system by considering the position of relevant items in the ranked list. It takes into account the notion that items higher in the ranking should receive more credit than items lower in the ranking. Recall@N, on the other hand, assesses the proportion of correctly identified relevant items in the top N recommendations, relative to the total number of relevant items in the dataset. In simpler terms, it indicates how many relevant items were successfully identified among the top N recommendations. The formulas for these metrics are as follows:

$$NDCG@N = \frac{DCG@N}{IDCG@N} = \frac{\sum_{i=1}^{k\ (actual\ order)} \frac{Gains}{log_2(i+1)}}{\sum_{i=1}^{k\ (ideal\ order)} \frac{Gains}{log_2(i+1)}} \quad (1)$$

$$Recall@N = \frac{No.of\ recommended\ items\ @N\ that\ are\ relevant}{Total\ No.of\ relevant\ items} \quad (2)$$

In our experiments, we utilized the XLNet transformer architecture, which offers support for both auto-regressive language modeling and auto-encoding, along with the PLM training strategy. Additionally, XLNet can also employ other training strategies such as MLM, CLM, and RTD. We conducted two experiments using different datasets. The first dataset focused on a specific domain of Amazon products known as Amazon Fashion. The second dataset encompassed 15 diverse domains as described in Table 1, which was outlined in the previous section.

In the first experiment, we focused on a single domain of Amazon Fashion, containing 186,637 instances. We began by preprocessing the dataset, which involved removing unused columns from both the reviews and product metadata. Additionally, duplicate instances were eliminated, and a new column called "event_type" was added with a value of "purchase" to indicate the interaction type. The product and review sets were then merged together. Session IDs were assigned by concatenating the reviewerID and unixReviewTime, and sessions with only one interaction were removed. This preprocessing approach was similar to that used for the multi-domain dataset. Categorical features such as user_id, event_type, brand, category, verified, price, and overall were defined. Temporal features were extracted based on unixReviewTime to create cyclical features (sine and cosine) that could be represented in a continuous space. Additionally, continuous features like "overall" were normalized. However, overall was found to be more useful as a categorical feature. The maximum session length was set to 20, and the model was trained using a sliding window approach. Specifically, the model was trained with data from four days out of seven days, with validation data from the following day. This process continued iteratively, with the training set shifting forward one day at a time. Four different training strategies were applied: MLM, PLM, RTD, and CLM. Furthermore, a sub-experiment was conducted to explore the use of side information for next item prediction, incorporating categorical features.

In the second experiment, we followed a similar approach to the first experiment, but this time, we utilized a multi-domain dataset encompassing products from 15 different categories on Amazon. The preprocessing steps were similar to those in the first experiment. Categorical features such as user_id, event_type, brand, category, verified, price, and overall were defined, and temporal features were extracted from unixReviewTime. Continuous features were normalized, with overall being primarily treated as a categorical feature. The maximum session length was set to 20, and the model was trained using a sliding window approach similar to the first experiment. However, in this experiment, we focused solely on the MLM training strategy. Additionally, a sub-experiment was conducted to explore the integration of side information for next item prediction, incorporating categorical features. A notable challenge encountered in this experiment was the computational cost of data preprocessing, owing to the large number of reviews and products associated with the 15 categories of Amazon products. This challenge required careful management of computational resources and optimization of preprocessing pipelines to ensure efficient execution.

Finally, a GRU trained with the CLM strategy was employed as the baseline model. The choice of GRU, a type of Recurrent Neural Network (RNN), was made to facilitate a comparison between the training strategies based on transformers and those based on RNNs. This comparison aims to evaluate the effectiveness of transformer-based strategies in session-based recommendation tasks relative to traditional RNN-based approaches.

5 Results

The results of the first experiment are depicted in Figure 3. It's evident that the training strategies of MLM, CLM, PLM, and RTD based on the XLNet architecture outperform the GRU algorithm. Additionally, when incorporating side information for next item prediction using categorical features, the CLM strategy achieves the best results. Specifically, this strategy achieves a remarkable improvement of +168.81% in both NDCG@10 and NDCG@20 relative to the baseline. Furthermore, it achieves a notable improvement of +25% in both Recall@10 and Recall@20 compared to the baseline. These results underscore the effectiveness of transformer-based strategies, particularly CLM with side information, in enhancing the performance of session-based recommendation systems.

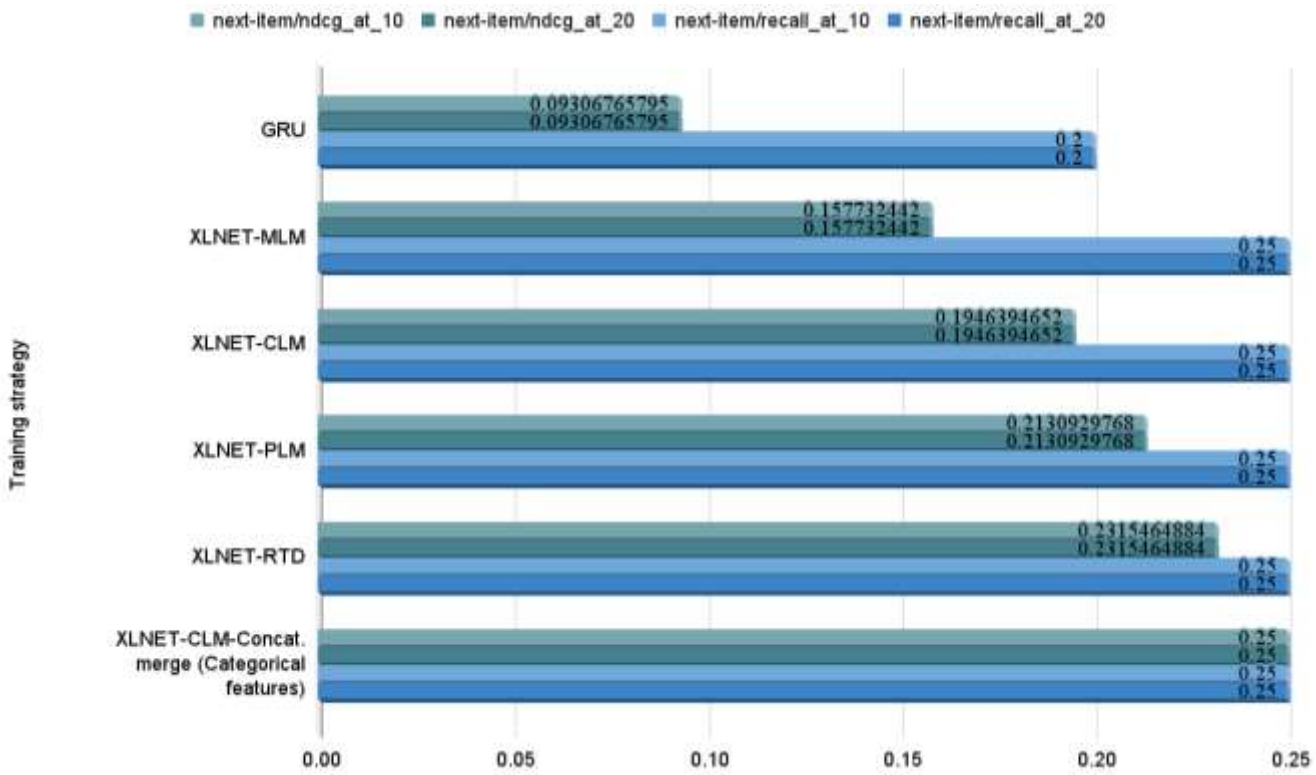

Figure 3. Results of first experiment with a domain data.

The results of the second experiment are presented in Figure 4. It is evident that the training strategy of MLM based on the XLNet architecture outperforms the GRU algorithm. Additionally, when incorporating side information for next item prediction using both categorical and continuous features, the MLM strategy achieves the best results. Specifically, this strategy achieves a significant improvement of +135.71% in NDCG@10 and +136.23% in NDCG@20 relative to the baseline. Furthermore, it achieves a notable improvement of +86.39% in Recall@10 and +95.69% in Recall@20 compared to the baseline. These results highlight the superiority of transformer-based strategies, particularly MLM with side information, in enhancing the performance of session-based recommendation systems.

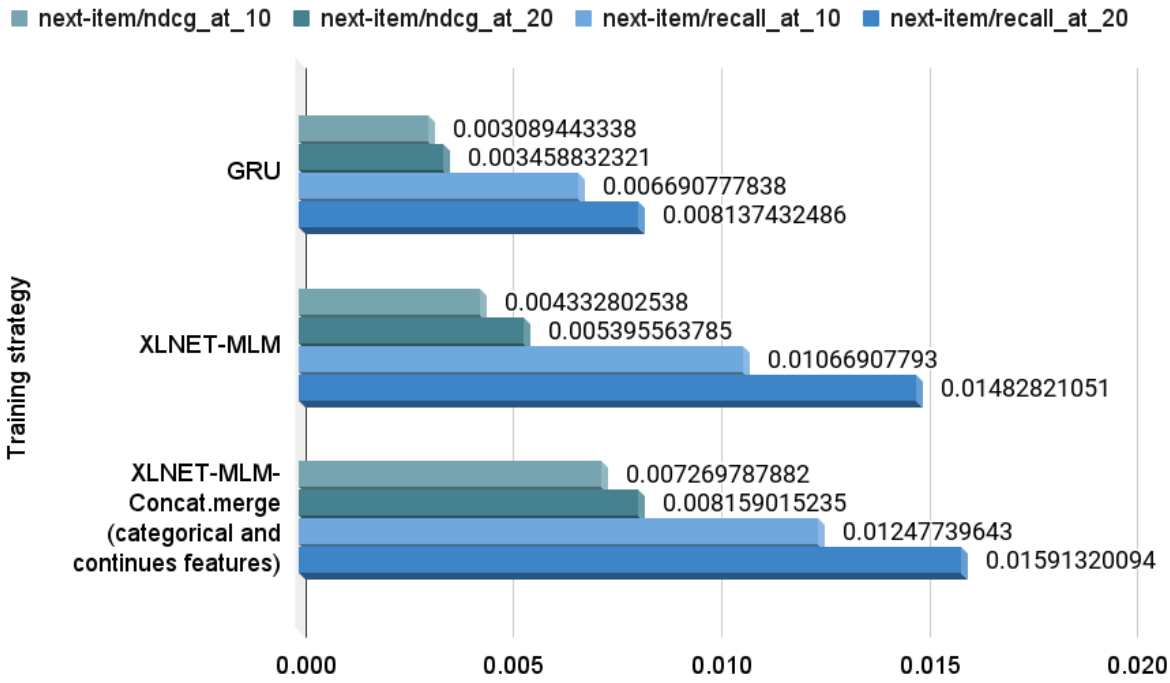

Figure 4. Results of second experiment with multi-domain data.

6 Conclusions

We conducted an analysis on the Transformer-Based Multi-Domain Recommender System for E-commerce using a dataset comprising 102 million reviews and 6 million products' metadata. This dataset underwent preprocessing to generate two new datasets, one for domain-specific data and the other for multi-domain data. In each dataset, we selected one week with the highest number of instances for experimentation.

Using NVTabular, we engineered features and selected principal categorical, continuous, and cyclical features such as overall rating, verification status, category, and temporal characteristics. Various training strategies, including Masked Language Modeling, Causal Language Modeling, Permuted Language Modeling, and Replaced Token Detection, were applied to the XLNet transformer architecture. These models were incrementally trained and evaluated using sliding windows by day.

The results of these experiments were compared with those obtained from a Recurrent Neural Network (RNN) model, specifically the GRU. The findings demonstrated that the XLNet transformer architecture, coupled with diverse training strategies, outperformed the GRU in both domain-specific and multi-domain data settings. This underscores the efficiency of transformer-based approaches in session-based recommender systems for multi-domain data. However, it is worth noting that while notable improvements were observed in the multi-domain data experiments, the results were slightly better for the domain-specific data. This suggests that the task of session-based recommender systems for multi-domain data remains challenging and warrants further investigation.

Acknowledgements

The authors express their gratitude to the Language & Knowledge Engineering Lab (LKE) and Facultad de Ciencias de la Computación (FCC) of Benemérita Universidad Autónoma de Puebla (BUAP), the School of Enterprise Computing and Digital Transformation of Technological University Dublin (TU Dublin), and the Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) for their support in providing computing resources and financial assistance.

References

de Souza Pereira Moreira, G., Rabhi, S., Lee, J.M., Ak, R., Oldridge, E. (2021). Transformers4rec: Bridging the gap between nlp and sequential / session-based recommendation. Proceedings of the 15th acm conference on recommender systems (p. 143–153). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3460231.3474255

Gheewala, S., Xu, S., Yeom, S., Maqsood, S. (2024). Exploiting deep transformer models in textual review based recommender systems. Expert Systems with Applications, 235, 121120, https:// doi.org/https://doi.org/10.1016/j.eswa.2023.121120 Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417423016226

Jannach, D., Ludewig, M., Lerche, L. (2017, dec). Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Modeling and User-Adapted Interaction, 27 (3–5), 351–392, https://doi.org/10.1007/s11257-017-9194-1 Retrieved from https://doi.org/10.1007/s11257-017-9194-1

Lu, K., Potash, P., Lin, X., Sun, Y., Qian, Z., Yuan, Z., Naumann, T., Cai, T., Lu, J. (2023, July). Prompt discriminative language models for domain adaptation. T. Naumann, A. Ben Abacha, S. Bethard, K. Roberts, & A. Rumshisky (Eds.), Proceedings of the 5th clinical natural language processing workshop (pp. 247–258). Toronto, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.clinicalnlp-1.30

Meng, Y., Krishnan, J., Wang, S., Wang, Q., Mao, Y., Fang, H., Ghazvininejad, M., Han, J., Zettlemoyer, L. (2023). Representation deficiency in masked language modeling.

Ni, J., Li, J., McAuley, J. (2019, November). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp) (pp. 188–197). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1018

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X. (2020, September). Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 63 (10), 1872–1897, https://doi.org/10.1007/s11431-020-1647-3 Retrieved from http://dx.doi.org/10.1007/s11431-020-1647-3

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P. (2019). Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2023). Attention is all you need.

Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M., Lian, D. (2021). A survey on session-based recommender systems.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q.Z., Orgun, M. (2019, August). Sequential recommender systems: Challenges, progress and prospects. Proceedings of the twenty-eighth international joint conference on artificial intelligence. International Joint Conferences on Artificial Intelligence Organization. Retrieved from http://dx.doi.org/10.24963/ijcai.2019/883

Wang, S., Pasi, G., Hu, L., Cao, L. (2020). The era of intelligent recommendation: Editorial on intelligent recommendation with advanced ai and learning. IEEE Intelligent Systems, 35 (5), 3-6, https://doi.org/10.1109/MIS.2020.3026430

Wu, X., & Varshney, L.R. (2023). A meta-learning perspective on transformers for causal language modeling.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V. (2020). Xlnet: Generalized autoregressive pretraining for language understanding.

Çano, E., & Morisio, M. (2017, November). Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21 (6), 1487–1524, https://doi.org/10.3233/ida-163209 Retrieved from http://dx.doi.org/10.3233/IDA-163209