TRACE: Transformer-based user Representations from Attributed Clickstream Event sequences

William Black[1], Alexander Manlove[1], Jack Pennington[1], Andrea Marchini[1], Ercument Ilhan[1] and Vilda Markeviciute[1]

[1]*Expedia Group, 407 St John St, London EC1V 4EX*

Abstract

For users navigating travel e-commerce websites, the process of researching products and making a purchase often results in intricate browsing patterns that span numerous sessions over an extended period of time. The resulting clickstream data chronicle these user journeys and present valuable opportunities to derive insights that can significantly enhance personalized recommendations. We introduce TRACE, a novel transformer-based approach tailored to generate rich user embeddings from live multi-session clickstreams for real-time recommendation applications. Prior works largely focus on single-session product sequences, whereas TRACE leverages site-wide page view sequences spanning multiple user sessions to model long-term engagement. Employing a multi-task learning framework, TRACE captures comprehensive user preferences and intents distilled into low-dimensional representations. We demonstrate TRACE's superior performance over vanilla transformer and LLM-style architectures through extensive experiments on a large-scale travel e-commerce dataset of real user journeys, where the challenges of long page-histories and sparse targets are particularly prevalent. Visualizations of the learned embeddings reveal meaningful clusters corresponding to latent user states and behaviors, highlighting TRACE's potential to enhance recommendation systems by capturing nuanced user interactions and preferences.

Keywords

transformers, user embeddings, clickstream data, multi-task

1. Introduction

On tourism e-commerce websites users often exhibit complex navigation patterns whilst they browse travel and accommodation options before making a purchase. A typical user could land on the homepage, search for a flight then bounce, only to return a few days later to browse hotels and then purchase a package holiday. The resulting clickstream data captures these intricate journeys and offers valuable insights into users' behaviour and intentions. By harnessing this data and better understanding users' latent psychological states and preferences, we can significantly enhance personalized experiences by matching them with more relevant content [1, 2, 3, 4, 5] and adapting the experience to better suit their context [4]. For instance, users earlier in their search can be presented with more exploratory content, as compared to users nearer the end of the purchase funnel.

However, achieving this level of personalization can be challenging as user journeys often span multiple sessions over an extended period of time, and specific goals, such as completing a purchase, occur infrequently within this window. This is a particularly pertinent challenge within the tourism industry as users will often only make one booking a year, which can take weeks of searching and planning before purchasing it months in advance.

In this work, we present TRACE (Transformer-based Representations of Attributed Clickstream Event sequences), a novel approach for generating rich user embeddings from live multi-session clickstream data with sparse targets. TRACE employs a multi-task learning (MTL) framework, where a lightweight transformer encoder is trained to predict multiple user engagement targets based on sequences of

Workshop on Recommenders in Tourism (RecTour 2024), October 18th, 2024, co-located with the 18th ACM Conference on Recommender Systems, Bari, Italy

wblack@expediagroup.com (W. Black); amanlove@expediagroup.com (A. Manlove); jpennington@expediagroup.com (J. Pennington); amarchini@expediagroup.com (A. Marchini); eilhan@expediagroup.com (E. Ilhan); vmarkeviciute@expediagroup.com (V. Markeviciute)

attributed clickstream events. By jointly predicting a diverse set of user future engagement signals, the model is encouraged to learn robust versatile representations. We demonstrate its effectiveness using a real-world travel e-commerce dataset.

Numerous works have explored the use of statistical and machine learning techniques on clickstreams to mine patterns [6, 7, 8] or cluster user behaviors [9, 10, 11, 12] for analytical insights or motivating recommendations. Comparable works have also investigated neural and MTL approaches to user modeling, but typically focus on product-level interactions or single session sequences [13, 14, 15, 16]. TRACE instead ingests live clickstream data and addresses more general sequences of site-wide page views spanning multiple sessions in order to obtain rich user journey representations for real-time downstream applications. PinnerFormer [17] notably uses a transformer, but relies on previously learned embeddings and abundant pin-based interactions. TRACE learns directly and exclusively from the sequence of attributed page views and employs a MTL approach to overcome sparse engagement signals. Zhuang et al. [18] studied attributes at the sequence level, whereas TRACE is more granular and addresses attributes at the event-level. Where Rahmani et al. [19] incorporated temporal signals in sequential recommendations, TRACE instead adopts learnable positional encodings which capture both event and session positions.

Overall, the key distinction of TRACE is the use of a transformer-based MTL framework with event-session position encoding to generate versatile user embeddings from enriched multi-session clickstream sequences with event-level attributes, which has not been explored in depth by previous research nor applied to travel e-commerce.

2. Methodology

2.1. Problem Formulation

Each time a user visits a new page it is logged in a clickstream as a page view event $P \in \mathcal{P}$, characterized by a small set of contextual features including the page name and timestamp t_P. These events collectively form user sessions $\mathcal{S}$, representing ordered sequences of the pages visited within defined time intervals. Formally, a session $\mathcal{S} = \{P_0, P_1, ..., P_N\}$, where P_j denotes the jth page the user visited in this session, subject to the condition that

$$t_{P_j} - t_{P_{j-1}} \leq T, \quad \forall j \in [1, N]. \tag{1}$$

Here T is a fixed constant, often in the order of magnitude of a few hours. If the difference in timestamps of two sequential page view events is greater than T, the latter is considered to be in a new session.

Then for each user, we define their journey J as the chronological sequence of their sessions, where $J = \{\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_k\}$, with $\mathcal{S}_i$ representing their ith session. In this way a journey J is the sequence of pages a user has visited across multiple sessions. We use a corpus of user journeys $\mathcal{J} = \{J_0, J_1, ...\}$ captured on a large-scale travel e-commerce site over a few months, where $|\mathcal{J}| > 50\text{M}$, and the vocabulary exceeds 1000 page names.

Our objective is to predict future engagement of users using their past navigation patterns on the website. Formally, we want to learn a model $f : \mathcal{J} \rightarrow \mathbb{R}^d$ for some positive integer d, which summarises these journeys in rich low dimensional representations that can then be used for downstream machine learning applications, such as content personalisation and product recommendations. As such the model f must satisfy three main requirements:

1. Effectively capture the intricate page navigation patterns in users' journeys which span multiple sessions.
2. Meaningfully distill user journeys into embeddings that can predict engagement across diverse tasks and contexts.
3. Scale efficiently to accommodate high-traffic real-time production environments.

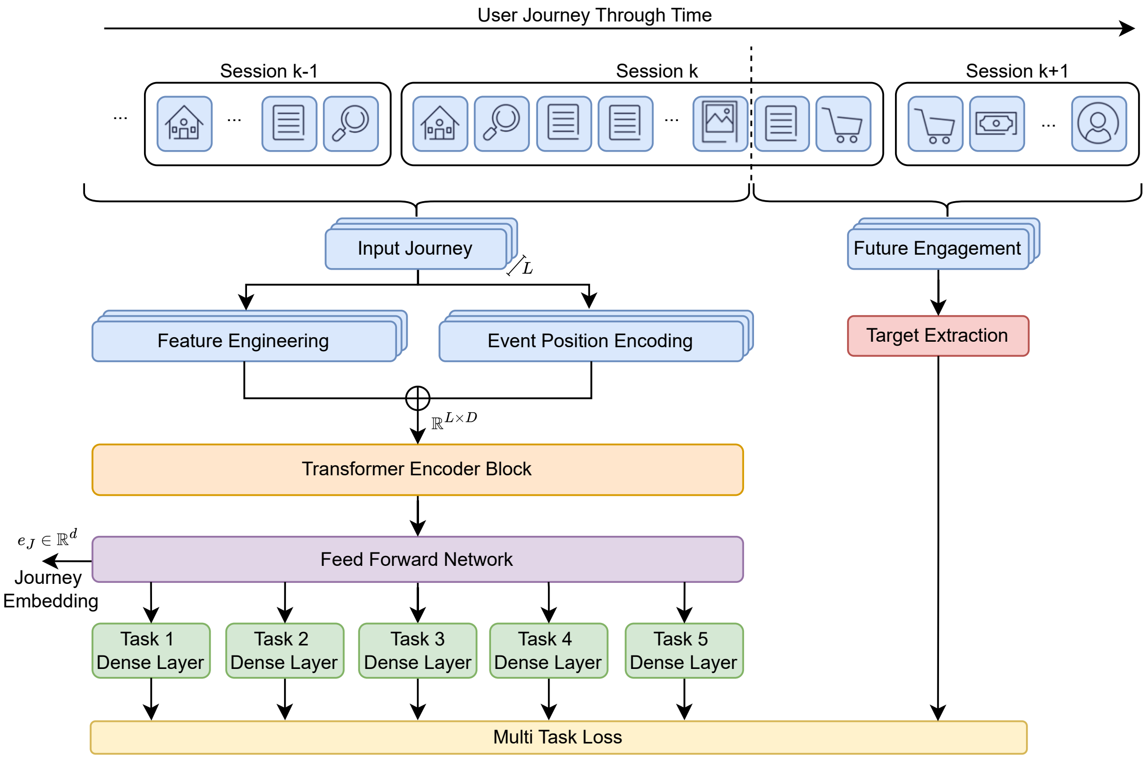

Figure 1: Overview of the TRACE multi-task transformer architecture.

To generate our datasets, we split each journey at a random point and designate the pages before as the input journey, and those after to be used for target generation.

In our proposed approach, TRACE, we train a multi-task transformer. This model takes as input some sequence of pages in the form of some journey J, and predicts a cohort of future user engagement targets. We extract the output of the final layer of the shared backbone of the model as the journey embedding $e_J \in \mathbb{R}^d$. We hypothesise that if the embeddings are predictive across a cohort of diverse user engagement tasks, they will capture a generalised understanding of a user's diverse intents. Figure 1 illustrates the components of TRACE. We address each in more detail below.

2.2. Feature Engineering and Position Encoding

We first crop each input journey, taking up to the L most recent page view events, where L is chosen in a way to capture most users' entire recent page view history.

Each page view event P has a set of categorical attributes, such as a page name and the user's device type, which are passed through their own learnable embedding layer to produce a dense representation in $\mathbb{R}^{32}$. We engineer two features from the event timestamp; the time interval between consecutive events and the time elapsed until the most recent event, both logged and standard scaled. Additionally we encode session ID where events in the nth most recent session are given value n. These time-based features aim to capture planning phases and session gaps common in extended travel user journeys. All features are standard scaled and concatenated s.t. each P is now represented by a vector in $\mathbb{R}^D$, where D is approximately a few hundred.

We also enumerate the event position, where the mth most recent event in the entire journey is given value m. Then both the event and session position indexes are independently embedded in $\mathbb{R}^D$ via their own learnable layers, and added onto the final feature vector, acting as an event-session position encoding. This was designed to allow the model to learn representations specific to session and position combinations, enabling it to capture dynamics both within and across multiple sessions more effectively.

For input journeys of length less than L we pad the D features with value 0. As such each journey J can now be encoded as some matrix $M_J \in \mathbb{R}^{L \times D}$.

2.3. Model Architecture

In TRACE, we use a transformer encoder architecture to process the input sequences of pages, and train it in a multi-task regime across five different targets, representing a variety of future user engagement signals.

An encoded journey $M_J \in \mathbb{R}^{L \times D}$ is passed through a single transformer encoder block, constisting of a multi-head self-attention layer with 8 heads followed by a position-wise fully connected feed-forward network (FFN) with an intermediate dimension of 128. We employ dropout and a residual connection around each of the two sub-layers, followed by layer normalization. Global max pooling is applied to the output of this encoder block, before being forward passed through a FFN. For an input journey J the output of this shared backbone is some $e_J \in \mathbb{R}^d$.

This tensor e_J is then passed through five separate task-specific dense layers, each compressing down to a scalar value so the final output of the model is some five logits $\hat{\mathbf{y}} \in \mathbb{R}^5$. After training we then remove the five task-specific heads, and take the output of the shared backbone e_J as the journey embedding. We deliberately restrict the heads to be simple logistic regression layers. This approach encourages the shared backbone to capture most of the nuance, ensuring the embeddings are information-rich and generalizable, as opposed to relying too heavily on the task-specific layers.

Throughout the architecture we use ReLU activations, except for the final shared dense layer where sigmoid is used for its desirable bounding property. This ensures normalization of the output embedding, with our experiments demonstrating no performance loss. We set dimension $d = 32$ for the embedding, which is well suited for downstream applications.

2.4. Multi Task Training Regime and Objective

The motivation behind the MTL approach is that by jointly predicting a diverse set of user engagement signals, the model is encouraged to learn comprehensive and generalizable representations that can be effectively utilized across a variety of downstream applications, extending beyond just the tasks during training. Furthermore, by mixing the infrequent targets such as purchases, with more common events like product searches, the model learns from a stronger signal and as our results show perform better on those sparse tasks. This is especially advantageous in the travel domain for events such as bookings, as demonstrated in our experiments.

The model is trained on five binary classification tasks which represent potential future actions of a user: (PW2) Make any purchase within two weeks; (BN5) Bounce within next five pages, and the following which relate to actions within rest of session; (SRP) Make a search for a product; (PDP) View a product details page; and (VUO) View an upcoming order. For more details on the metrics used in training see Table 1. Each task head has its own class-weighted binary cross-entropy loss function. The overall objective is expressed as a linear combination of these task-specific losses. For a journey J with model prediction $\hat{\mathbf{y}}$ and true labels $\mathbf{y}$, the loss is defined as:

$$\mathcal{L}(J, \mathbf{y}) = -\sum_{k=1}^{5} \left[w_k \cdot y_k \cdot \log(\hat{y}_k) + (1 - y_k) \cdot \log(1 - \hat{y}_k) \right]. \tag{2}$$

Class weights w_k are computed as the reciprocal of the proportion of positive samples for each task k, in order to account for task-specific class imbalance. We weights tasks equally to encourage the model to develop features which generalize across each task.

3. Experimental Results

3.1. Downstream Embedding Evaluation

Supervised probing techniques have previously been developed to assess linguistic embeddings [20, 21, 22], although are not directly suited to this scenario. We instead propose a downstream strategy for evaluating the richness of information contained within a set of embeddings. After training, we compute ground truth targets on an unseen test set of historical user journeys. These targets seek to encapsulate users' latent psychological states and future intentions. For this, we use the same five tasks from the TRACE objective in eqn. 2, and introduce three more evaluation tasks that were not previously seen. These include: (PWS) whether a user converts in the current session; (HOM) returns to the homepage in the current session; and (RE7) whether they return to the site within seven days. For more details see Table 1. This captures a broad scope of user outcomes, allowing us to characterize how well the embeddings generalize.

We pass the unseen test journeys through the model to obtain a corresponding set of embeddings. Next, we train XGBoost models [23] on these test set embeddings. We fit one XGBoost model independently to each unique evaluation task and optimize hyperparameters, such as *max_depth* and *learning_rate*, using K-fold cross validation. The trained XGBoosts then undergo evaluation and we compute performance metrics on the model predictions. These metrics serve as proxies for assessing the richness of embeddings and exemplify downstream model performance across various use cases. Throughout this section, we evaluate each upstream embedding model by using the same procedure on the same unseen test set.

Table 1
Explanation of targets and how they are used in training and downstream evaluation.

Target Name	Description of future action of user	Used in training	Used in Evaluation
PW2	Make a purchase within two weeks.	Yes	Yes
BN5	Bounce within next five pages.	Yes	Yes
SRP	Make a search for a hotel or flight within session.	Yes	Yes
PDP	View a hotel or flight details page within session.	Yes	Yes
VUO	View an upcoming order within session.	Yes	Yes
PWS	Make a purchase within session.	No	Yes
HOM	Return to homepage within session.	No	Yes
RE7	Return to site within 7 days.	No	Yes

3.2. Comparable Models vs TRACE

We evaluate the quality of TRACE embeddings against several comparable approaches. We express our comparisons as the mean uplift taken over all evaluation tasks. Results are shown in Table 2.

Table 2
Mean % uplift in XGBoost metrics from myopic baseline across eval. tasks for TRACE and comparable models.

Model	AUROC	AUPRC	F1	Acc
TRACE	**+7.23**	**+13.58**	**+2.73**	**+2.15**
ST Cohort	+6.38	+10.75	+2.72	+2.06
ST Aggregated	+6.34	+10.62	+2.18	+1.73
MT LSTM	+1.91	−3.29	−0.29	+0.27
Mini-GPT	+1.86	−2.40	−1.13	−0.60

Myopic Baseline. Our baseline predicts targets using explicit attributes from only the most recent event. We report all results as percentage uplifts from this. TRACE significantly outperforms this, highlighting the benefits of mining a user's full navigation history.

Single Task Cohort. To demonstrate the effectiveness of TRACE's MTL approach, we trained a dedicated single-task transformer for each of the evaluation tasks. These models each produce an embedding. In Table 3, the TRACE score on a given task is compared to the corresponding dedicated ST model embedding's score. Overall results show that the TRACE embeddings outperform every task-specific equivalent on the 5 tasks TRACE was trained on, and even wins on all but one of the unseen targets, demonstrating the advantages of the MTL approach.

Single Task Aggregated. Here we combine the task-specific models' embeddings into a single embedding of the same length by taking the mean along each dimension.

Multi-Task LSTM. We note the demonstrated efficacy of LSTMs in related works [14, 18, 24, 25]. We train a comparable LSTM minimizing the same multi-task objective function shown in (2).

Mini-GPT. We train a small GPT-style model [26] on the page name sequences, with a single transformer block and causal masking in the attention layer for next event prediction. Embeddings are computed from the mean of the transformer block outputs.

Table 3
Mean % uplift in XGBoost AUROC on the eight eval. tasks for TRACE vs dedicated single-task models.

Model	PW2	BN5	SRP	PDP	VUO	HOM	PWS	RE7
TRACE	**+11.8**	**+9.32**	**+6.73**	**+7.29**	**+3.99**	**+7.70**	**+4.35**	+6.52
ST	+11.2	+8.14	+5.08	+6.25	+3.56	+5.59	+2.99	**+8.26**

3.3. Ablation Experiments

In Table 4 we list the results of our ablation studies.

3.3.1. Position Encodings

In section 2.2, we discussed our approach to position encoding which is designed to tackle event sequences over multiple sessions. Static trigonometric position encodings are also widely popular [13, 27]. We trained a variant including this additional encoding, but found better performance without it.

3.3.2. Number of Encoders

Here, we vary the number of transformer encoders, h. Our results suggest that $h = 1$ encoder is sufficient for capturing the structure of the data, likely due to our sequences being of relatively shorter length with small vocabulary compared to typical NLP applications [27]. We measured the time taken for the forward pass in each variant. The experiments were conducted on a system equipped with an Nvidia T4 Tensor Core GPU (16 GiB VRAM) and an Intel Xeon processor (32 vCPUs, 128 GiB RAM, 2.5 GHz clock speed). We repeat the model call 10,000 times and measure the mean and standard deviation for various encoder configurations. The results are as follows:

- $h = 1$ encoder: 27.5 ms ± 0.1 ms
- $h = 2$ encoders: 40.8 ms ± 0.1 ms
- $h = 3$ encoders: 54.7 ms ± 0.6 ms
- $h = 4$ encoders: 67.7 ms ± 0.4 ms

Our final model design used only a single encoder $h = 1$, which is sufficiently fast taking only 27.5 milliseconds on average for the forward pass. This is well within our self-imposed upper limit of 100ms latency, which we find to be practical for real-time applications.

Table 4
Mean XGBoost performance uplift % on evaluation tasks for TRACE variants' embeddings vs myopic baseline.

Pos. Encodings	Num. Encoders	Chrono. Features	AUROC	AUPRC	F1	Accuracy
Event-Session[†]	1[†]	Timestamp & Session[†]	**+7.23**	**+13.58**	**+2.73**	+2.15
Trigonometric			+6.64	+12.22	+2.47	**+2.17**
	2		+6.87	+12.62	+2.65	+2.07
	3		+6.84	+12.76	+2.53	+1.98
	4		+6.84	+12.76	+2.61	+2.06
		Timestamp	+6.62	+12.24	+2.45	+1.94
		None	+6.36	+12.0	+2.17	+1.70

[†] *Final variant used in proposed TRACE model.*

3.3.3. Chronological Features

To better understand the specific performance gains from chronological features, we train variants which omit these. The "Timestamp" variant retains event timestamps but removes session ID, thereby eliminating explicit information about session continuity. The "None" variant excludes both session IDs and timestamps, retaining only the sequential order of events. Results demonstrate that including timestamp features enhances performance, but the greatest improvement arises from incorporating TRACE's session encoding on top of this, as used in our final variant. This highlights the effectiveness of TRACE in exploiting the multi-session structure of the sequences, and its significance for applications in e-commerce recommendation systems.

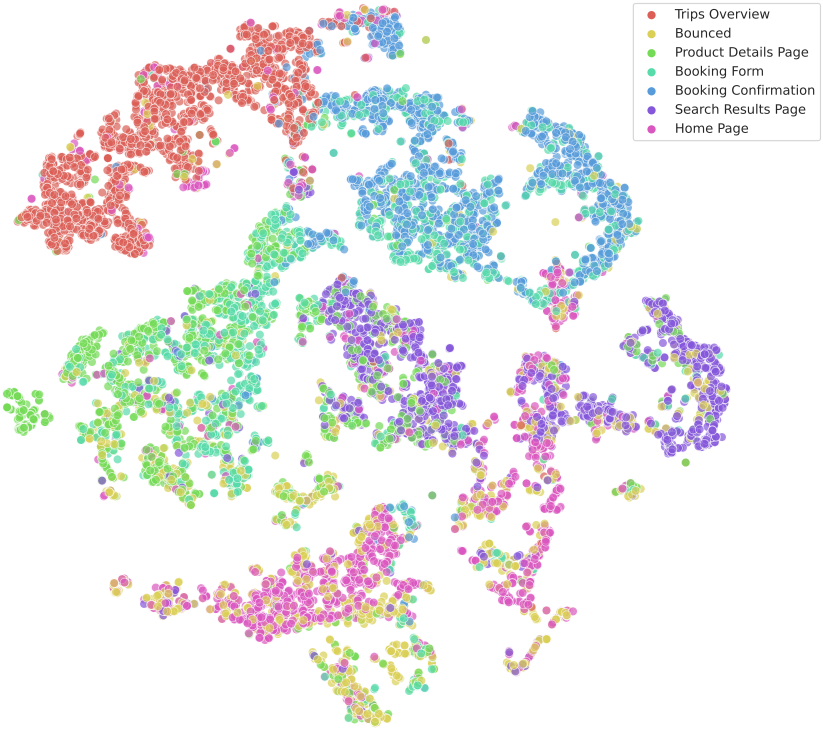

Figure 2: t-SNE projections of TRACE page sequence embeddings, colored according to the next page the users visited.

3.4. Visualisation of Learned Embeddings

In Fig. 2, we present a visualization of the 32-dimensional embeddings learned by TRACE, reduced to 2 dimensions using t-SNE [28]. This subset of observations was uniformly sampled with respect to users' next visited page, ensuring equal representation from seven common pages. We note the emergence of clusters corresponding to the next page visited by users, despite TRACE never being explicitly exposed to this information during training. Qualitatively, the clusters appear to loosely align with how a user traverses a website, going from homepage at the bottom progressing through to search and product pages, before reaching checkout and order confirmation. This underscores TRACE's ability to identify and encode patterns in user journeys, showcasing the effectiveness of our approach for generating information-rich embeddings.

4. Conclusion

In this work, we have presented TRACE, a novel approach for generating user embeddings from multi-session page view sequences through a multi-task learning (MTL) framework, which employs a lightweight, encoder-only transformer to process real-time cross-session clickstream data. Our experiments on a large-scale, real-world travel e-commerce dataset demonstrate the superior performance of TRACE embeddings compared to traditional single-task and LSTM-based models, and highlight its potential for enhancing tourism recommender systems. The learned embeddings exhibit strong results on a diverse set of targets and demonstrate the ability to generalize well to unseen tasks, underscoring their utility for applications like content personalization and user modeling. Visualizations reveal that TRACE can effectively capture meaningful clusters corresponding to latent user intents and behaviors.

To reinforce the performance of TRACE, we plan to publish results showing its strength on a public e-commerce user-journey dataset produced by Coveo [29]. Although this dataset is neither multi-session nor tourism-specific, its user journeys exhibit comparable navigation patterns, which will underscore the robustness of the TRACE architecture. Additionally, we intend to integrate these embeddings into our in-house recommendation systems and evaluate their effectiveness in online experiments.

In the future, we plan to explore the integration of LLMs, as in [30, 31], and investigate hierarchical models to further improve the model's representational capacity.

References

[1] H. Zhao, L. Si, X. Li, Q. Zhang, Recommending complementary products in e-commerce push notifications with a mixture model approach, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 909–912.

[2] X. Shen, J. Shi, S. Yoon, J. Katzur, H. Wang, J. Chan, J. Li, Learning to personalize recommendation based on customers' shopping intents, 2023. arXiv:2305.05279.

[3] I. Kangas, M. Schwoerer, L. J. Bernardi, Recommender systems for personalized user experience: Lessons learned at booking.com, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 583–586. URL: https://doi.org/10.1145/3460231.3474611. doi:10.1145/3460231.3474611.

[4] W. Black, E. Ilhan, A. Marchini, V. Markeviciute, Adaptex: A self-service contextual bandit platform, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, ACM, 2023. URL: http://dx.doi.org/10.1145/3604915.3608870. doi:10.1145/3604915.3608870.

[5] M. Grbovic, H. Cheng, Real-time personalization using embeddings for search ranking at airbnb, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 311–320.

[6] E. Olmezogullari, M. S. Aktas, Representation of click-stream datasequences for learning user navigational behavior by using embeddings, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 3173–3179.

[7] S. D. Bernhard, C. K. Leung, V. J. Reimer, J. Westlake, Clickstream prediction using sequential stream mining techniques with markov chains, in: Proceedings of the 20th international database engineering & applications symposium, 2016, pp. 24–33.

[8] Y. S. Kim, B.-J. Yum, Recommender system based on click stream data using association rule mining, Expert Systems with Applications 38 (2011) 13320–13327.

[9] G. Wang, X. Zhang, S. Tang, H. Zheng, B. Y. Zhao, Unsupervised clickstream clustering for user behavior analysis, in: Proceedings of the 2016 CHI conference on human factors in computing systems, 2016, pp. 225–236.

[10] Q. Su, L. Chen, A method for discovering clusters of e-commerce interest patterns using clickstream data, electronic commerce research and applications 14 (2015) 1–13.

[11] J. Wei, Z. Shen, N. Sundaresan, K.-L. Ma, Visual cluster exploration of web clickstream data, in: 2012 IEEE conference on visual analytics science and technology (VAST), IEEE, 2012, pp. 3–12.

[12] M. Zavali, E. Lacka, J. De Smedt, Shopping hard or hardly shopping: Revealing consumer segments using clickstream data, IEEE Transactions on Engineering Management 70 (2021) 1353–1364.

[13] H. Bai, D. Liu, T. Hirtz, A. Boulenger, Expressive user embedding from churn and recommendation multi-task learning, in: Companion Proceedings of the ACM Web Conference 2023, 2023, pp. 37–40.

[14] M. Alves Gomes, R. Meyes, P. Meisen, T. Meisen, Will this online shopping session succeed? predicting customer's purchase intention using embeddings, in: Proceedings of the 31st ACM international conference on information & knowledge management, 2022, pp. 2873–2882.

[15] B. Requena, G. Cassani, J. Tagliabue, C. Greco, L. Lacasa, Shopper intent prediction from clickstream e-commerce data with minimal browsing information, Scientific reports 10 (2020) 16983.

[16] C. H. Tan, A. Chan, M. Haldar, J. Tang, X. Liu, M. Abdool, H. Gao, L. He, S. Katariya, Optimizing airbnb search journey with multi-task learning, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, ACM, 2023. URL: http://dx.doi.org/10.1145/3580305.3599881. doi:10.1145/3580305.3599881.

[17] N. Pancha, A. Zhai, J. Leskovec, C. Rosenberg, Pinnerformer: Sequence modeling for user representation at pinterest, in: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 2022, pp. 3702–3712.

[18] Z. Zhuang, X. Kong, R. Elke, J. Zouaoui, A. Arora, Attributed sequence embedding, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 1723–1728.

[19] M. Rahmani, J. Caverlee, F. Wang, Incorporating time in sequential recommendation models, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 784–790.

[20] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das, et al., What do you learn from context? probing for sentence structure in contextualized word representations, in: 7th International Conference on Learning Representations, ICLR 2019, 2019.

[21] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4129–4138.

[22] J. Hewitt, P. Liang, Designing and interpreting probes with control tasks, 2019.

[23] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[24] D. Koehn, S. Lessmann, M. Schaal, Predicting online shopping behaviour from clickstream data using deep learning, Expert Systems with Applications 150 (2020) 113342.

[25] C. O. Sakar, S. O. Polat, M. Katircioglu, Y. Kastro, Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks, Neural Computing and Applications 31 (2019) 6893–6908.

[26] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin,

Attention is all you need, Advances in neural information processing systems 30 (2017).

[28] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[29] J. Tagliabue, C. Greco, J.-F. Roy, B. Yu, P. J. Chia, F. Bianchi, G. Cassani, Sigir 2021 e-commerce workshop data challenge, 2021. URL: https://arxiv.org/abs/2104.09423. arXiv:2104.09423.

[30] K. Christakopoulou, A. Lalama, C. Adams, I. Qu, Y. Amir, S. Chucri, P. Vollucci, F. Soldo, D. Bseiso, S. Scodel, et al., Large language models for user interest journeys, arXiv preprint arXiv:2305.15498 (2023).

[31] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang, et al., Recommender systems in the era of large language models (llms), IEEE Transactions on Knowledge and Data Engineering (2024).