Understanding Players as if They Are Talking to the Game in a Customized Language: A Pilot Study

Tianze Wang[1,2*] **Maryam Honari-Jahromi**[2*]
Styliani Katsarou[2] **Olga Mikheeva**[1,2] **Theodoros Panagiotakopoulos**[2]
Oleg Smirnov[2†‡] **Lele Cao**[2†] **Sahar Asadi**[2†]
[1]KTH Royal Institute of Technology [2]King, Microsoft Gaming

Abstract

This pilot study explores the application of language models (LMs) to model game event sequences, treating them as a customized language. We investigate a popular mobile game, transforming raw event data into textual sequences and pretraining a Longformer model on this data. Our approach captures the rich and nuanced interactions within game sessions, effectively identifying meaningful player segments. The results demonstrate the potential of self-supervised LMs in enhancing game design and personalization without relying on ground-truth labels.

1 Introduction

The dominant form of human interaction is natural language, represented by a *stream of words*. Language Models (LMs) have become highly effective in understanding and representing these general-purpose natural languages. Similarly, when a human player interacts with a video game, the primary form of interaction is through game controls, which lead to visual and auditory feedback. This in-game interaction is typically recorded as a *stream of events*, each with rich attributes and categories. This pilot study explores **whether we can apply LMs, initially designed for word sequences, to model game event sequences**. Understanding player behavior through this modeling approach is crucial for designing engaging experiences, improving game mechanics, and personalizing content. For example, understanding the optimal balance between challenge and progression can enable dynamic game difficulty adjustments, maximizing the enjoyment experienced by players.

Traditionally, understanding game players has relied on surveys and interviews, such as those conducted in (Rodrigues et al., 2022). While these methods provide valuable insights, they are significantly limited by scalability. Deep Learning (DL) models, like those in (Cao et al., 2020), have been trained on aggregated (from game events) gameplay data to achieve in-game personalization, but they often neglect nuanced interactions. Recently, training DL models on sequential interactions between players and in-game items has been explored, as exemplified by (Villa et al., 2020). However, these modeled interactions are still relatively limited in type and richness compared to game events. Moreover, most of these DL models only optimize for specific personalization scenarios, requiring large amount of ground-truth labels, which are not always available.

As a consequence, self-supervised LM pretraining emerged as a promising approach to directly model the rich and fine-grained game events in a scalable way without requiring any labels. In principle, this pretrained model is not restricted to any specific personalization use case. To the best of our knowledge, this is the first attempt to pretrain an LM on game events by treating these events as a customized natural language. The highlights of this pilot study are: (§3) studying a popular mobile video game from King[1], Candy Crush Saga, (§4) developing a simple method for transforming a large amount of game events into language tokens, (§5) pretraining an LM on the customized "language" representing game events, (§6) reporting experimental results on the LM's intrinsic performance and its capability in understanding game players, and finally (§7) we outline measures employed to mitigate ethical considerations.

2 Related Work

Modeling sequential interactions between users and items has been extensively studied in recommendation systems. Initial approaches utilized Markovian

*Both authors contributed equally to this research.
†Joint senior authorship.
‡Corresponding author: oleg.smirnov@microsoft.com

[1]https://king.com

Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U), pages 47–52
November 16, 2024

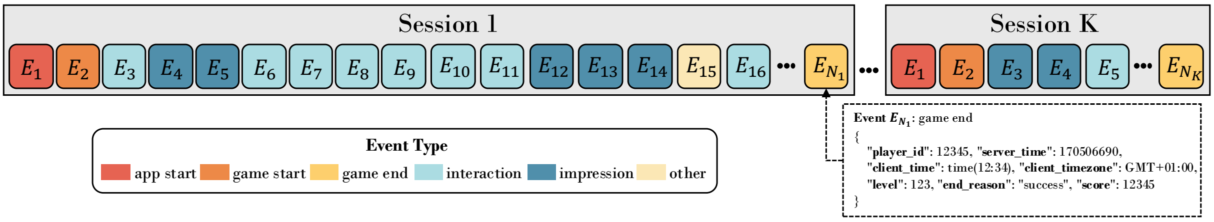


Figure 1: Example events segmented into semantic sessions. The final game-end event in “Session 1” is expanded to show details about its associated fields and values.

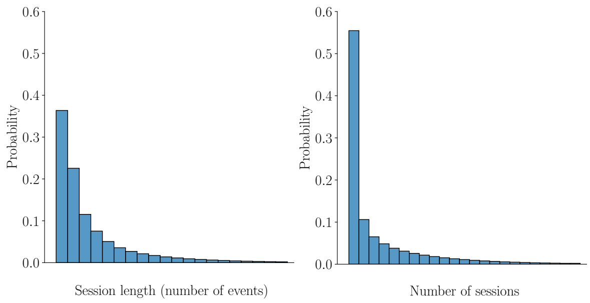
assumptions for collaborative filtering (Zimdars et al., 2001), later extended to Markov decision processes (Shani et al., 2005). Predicting future behavior trajectories using contextual and sequential information has been addressed with autoregressive Long Short-Term Memory models (Wu et al., 2017) and coupled Recurrent Neural Network (RNN) architectures for joint modeling of user/item interactions (Kumar et al., 2019). Explicitly modeling different types of user behavior, such as repeated consumption, has also shown to improve downstream performance metrics (Anderson et al., 2014; Ren et al., 2019).

LMs have been leveraged for embedding sequential data in recommendation settings, beginning with music track representations using the Word2Vec objective (Mehrotra et al., 2018) and extending to modeling sequences of listening sessions with RNNs (Hansen et al., 2020). More recently, self-attention sequential models have been introduced, such as BERT4Rec (Sun et al., 2019), which balance the trade-off between Markov chain models and neural network methods. Follow-up work on multi-task customer models for personalization has further advanced this field by integrating novel data augmentation and task-aware readout modules (Luo et al., 2023).

Despite these advancements, the application of LMs for user modeling in gaming remains underexplored. Our study proposes the first approach for learning representations of mobile game players by pretraining a Transformer architecture in a self-supervised manner, treating game event sequences as a customized natural language. This approach aims to capture the rich and nuanced interactions within game sessions.

3 The Game and Interaction Events

This pilot study focuses on Candy Crush Saga game. When a player interacts with this game on a mobile device, their behavior generates a sequence of time-ordered events, which are recorded locally on the user’s device and later sent to the central game server in batches. Example events include starting the game application, beginning a new game round, purchasing in-game items, and displaying pop-ups and notifications. The tracked player behavior events fall into 12 categories, each with an associated schema containing continuous and categorical features.



(a) (b)

Figure 2: (a) Histogram of session lengths and (b) the distribution of session quantities over a 15-day period shown up to the 99$^{\text{th}}$ percentile.

The player-game interaction events are segmented into sessions based on the player’s activity semantics, as illustrated in Figure 1. According to game analytics conventions recommended by the data scientists from the game producer, a session is considered to have ended if a player is inactive for 15 minutes or more. For this study, we collected a dataset of player event sessions over 15 days, with 10,000 players uniformly sampled from the entire player population. The resulting dataset consists of 125,000 sessions, split into a 2:1 train-test ratio. The distribution of session lengths in the dataset is shown in Figure 2a, while Figure 2b depicts the distribution of sessions quantities. Both session lengths and quantities approximately follow a geometric distribution.

Our collected event data, while superficially similar to tracking data in other domains like ecommerce, presents unique challenges. In-game interactions occur at a much higher frequency than in web browsing, resulting in large volumes of potentially redundant events that call for careful preprocessing and modeling of long-range dependencies. Additionally, game event sequences are often noisy, with incorrectly ordered events or missing ordering information due to users switching between online and offline modes, which can degrade model performance during training and inference.

4 From Events to Words

The raw format of game events is JSON. To make this data digestible by LMs, we designed a simple pipeline to transform raw events into textual sequences. As illustrated in Figure 3, the pipeline begins by removing unnecessary events and fields. Leveraging game-specific knowledge, we filter out non-informative data, such as device-specific logs, reducing the number of event fields by over 90%. We bin certain numerical features, such as the hour of the day, based on domain-specific knowledge to convert them into categorical variables. Additionally, we group similar in-game event identifiers, e.g., the name of the UI shown, to reduce the vocabulary size. The words are then grouped by users and sessions, ordered by timestamps to preserve the natural interaction flow, and concatenated to form a textual description of a player's interaction experience.

We use a word-level tokenizer that splits a space-separated string into tokens and maps them to unique identifiers. This approach suits the relatively small vocabulary of behavior data ($\sim$13,500 tokens), though the tokenized sequences are much longer than those in typical NLP tasks like sentiment analysis.

5 Pretrain a Language Model

The tokenized word sequences are often longer than 512 tokens, which are unmanageable for the conventional BERT (Kenton and Toutanova, 2019) architecture and its derivatives. Modeling long sequences poses a significant challenge to Transformer-based approaches due to the self-attention operation, which scales quadratically with input length in terms of memory and computational complexity. This challenge is intensified when modeling distant dependencies in extended gameplay experiences that involve concatenating multiple sessions. To overcome this, we adopt Longformer (Beltagy et al., 2020), a model designed specifically for processing long textual inputs.

model size	#layer	#head	dims	block size	#params
small	2	2	128	1024	2M
medium	6	6	384	2048	20M
large	12	12	768	4096	121M

Table 1: Hyperparameters for different model sizes.

model size	accuracy ↑	perplexity ↓	CE ↓
small	0.69 ± 0.06	3.27 ± 0.71	1.16 ± 0.22
medium	0.93 ± 0.01	1.28 ± 0.09	0.25 ± 0.07
large	$\mathbf{0.95 \pm 0.01}$	$\mathbf{1.16 \pm 0.05}$	$\mathbf{0.15 \pm 0.04}$

Table 2: Mean values and standard deviations of intrinsic language modeling metrics computed over five training runs.

Longformer combines dilated sliding window attention for local context and global attention on a few pre-selected input locations. This approach scales linearly with input size, enabling the processing of sequences up to 4,096 tokens in a single pass, which is sufficient for most behavior modeling scenarios. Additionally, Longformer's sparse attention pattern performs well in contexts where many tokens in the immediate local context may be redundant, as is often the case with high-frequency game events.

We pretrained several Longformer variants[2] from scratch with different capacities, based on the hyper-parameters listed in Table 1. We experimented with the baseline Longformer configuration, i.e., "*large*", and two smaller model variants with fewer internal layers and self-attention heads. The models were optimized with the masked language modeling (MLM) objective using Adam (Kingma, 2014) with a fixed learning rate of 2×10^{-5}. Each LM was trained from randomly initialized weights for 100 epochs with a batch size of 4 and gradient accumulation over 4 steps, resulting in an effective batch size of 16 (2^{16} tokens).

6 Results

First, we evaluate the intrinsic performance of the proposed approach using intrinsic MLM metrics. We report the Cross-Entropy (CE) loss and multi-class classification accuracy of predicting masked

[2]We use the HuggingFace Transformers (Wolf et al., 2019) library and PyTorch framework (Paszke et al., 2019) for model implementation. All models were trained with half-precision (FP16) on a single NVIDIA A100 GPU, with the *large* model taking approximately 50 hours to complete pretraining.

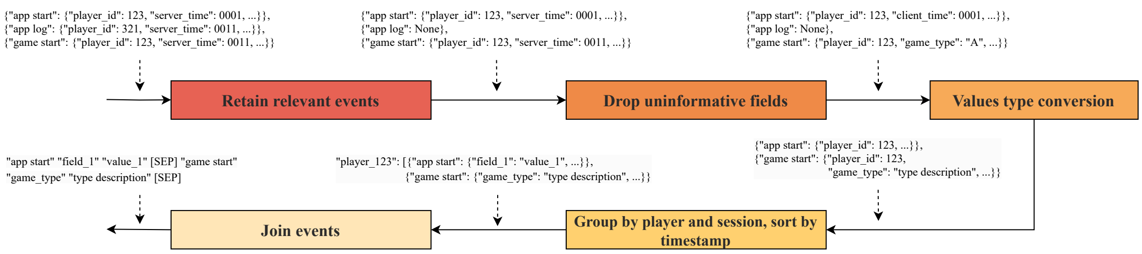


Figure 3: The pipeline to convert event streams to word streams.

tokens on the validation split for the tested model architectures, as shown in Table 2. Additionally, we report the perplexity score, following established methodologies for evaluating MLM pretraining performance (Liu et al., 2019). As expected, we observe that LMs with larger capacities achieve better fits for the behavior sessions without overfitting.

Next, we perform a qualitative analysis to identify spontaneous player clusters representing different behavioral persona. We extract embeddings of input token sequences from the pretrained *large* Longformer model. Using 4096×768-dim representations from the last Attention layer, we apply max pooling over sequence length to compute an embedding vector for each input sequence. These session embeddings are projected onto the first 50 principal components using linear PCA to reduce noise and speed up computation. The projections are then mapped to 2D space via t-SNE (Van der Maaten and Hinton, 2008) and clustered with a Gaussian Mixture Model (Reynolds et al., 2009) with eight components. The resulting t-SNE plot is shown in Figure 4a. Analyzing the average player behavior within the well-separated t-SNE clusters in Figure 4b, we collaboratively identified player segments with game analysts from a practical product perspective. Identified players' personas qualitatively resonate with what our user researchers extracted from self-reported behavioral surveys:

1. *Competitive devoted*: a skilled player who plays less often but long sessions, occasionally purchasing items and collecting utilities.
2. *Casual devoted*: a player who plays long sessions infrequently, engages in quests, collects rewards, and prefers free gameplay.
3. *Persistent devoted*: a player who plays frequent, long sessions without purchasing.
4. *Lean-in casual economy aware*: A skilled player who plays less often but for long sessions, occasionally buying items.
5. *Lean-in casual*: a skilled player who plays less often but for long sessions.
6. *Persistent casual*: a less skillful player who plays short, frequent sessions with little engagement in social and economic aspects.
7. *Persistent collector*: a player with frequent short sessions, collecting utilities to progress.

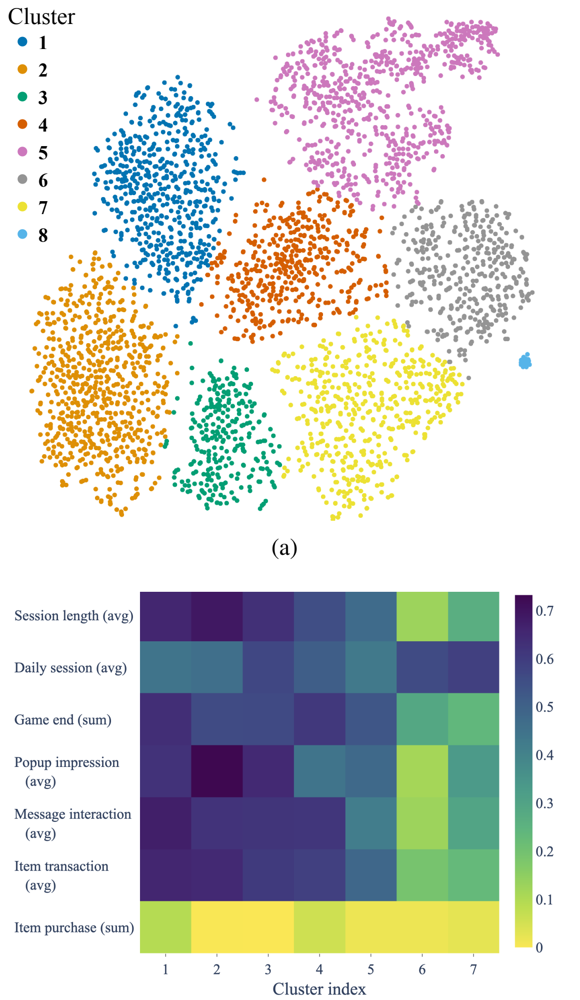
7 Ethical Considerations

Computational modeling of player behavior in games has raised various ethical concerns within both research and industry (Mikkelsen et al., 2017). In this pilot study, we utilize non-personally identifiable tracking data from in-game interactions to create vectorized representations of player behaviors. Our objective is to leverage these representations to support personalized and enhanced player experiences while maintaining ethical standards.

Potential ethical risks include (1) biases in the input dataset, such as under-representing less frequent player behaviors, and (2) the misapplication of models to different data distributions, known as Type III errors (Mikkelsen et al., 2017). To mitigate these risks, we use robust data validation and automated model analysis tools available in production-ready machine learning frameworks (Modi et al., 2017).

We address under-represented player behaviors through qualitative evaluation methods, such as embedding space visualization. Additionally, we periodically retrain the model with recent data to address distribution shifts, with retraining intervals determined empirically based on model performance and data drift.

For the downstream recommendation system, we plan to implement model explainability and uncertainty estimation methods to better understand the model's robustness, biases, and other ethical considerations. These measures aim to ensure that our



(b)

Figure 4: (a) t-SNE of the latent embedding space from the pretrained *large* Longformer with Gaussian Mixture Model clustering. (b) Histogram of quantized player events in clusters (excluding cluster 8 due to small size and lack of gameplay).

modeling approach supports ethical and responsible AI deployment.

8 Conclusion and Future Work

This pilot study demonstrates the potential of using self-supervised language models to understand player behavior by modeling game event sequences as a customized natural language. Our approach, leverages the Longformer model to effectively captures the rich and nuanced interactions within game sessions in a self-supervised manner, agnostic to downstream use-cases. The results highlight the model's ability to identify meaningful player segments, providing valuable insights for game design and personalization. For future work, we plan to extend training to single- and multitask fine-tuning with labeled datasets to benchmark against fully-supervised baselines. We anticipate that our approach can be extended to other event-based game datasets as well.

References

Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *Proceedings of the 23rd international conference on World wide web*, pages 419–430.

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

Lele Cao, Sahar Asadi, Matteo Biasielli, and Michael Sjöberg. 2020. Debiasing few-shot recommendation in mobile games. In *ORSUM@ RecSys*.

Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 53–62.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

DP Kingma. 2014. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1269–1278.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rui Luo, Tianxin Wang, Jingyuan Deng, and Peng Wan. 2023. Mcm: A multi-task pre-trained customer model for personalization. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 637–639.

Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 2243–2251.

Benedikte Mikkelsen, Christoffer Holmgård, and Julian Togelius. 2017. Ethical considerations for player modeling. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Akshay Naresh Modi, Chiu Yuen Koo, Chuan Yu Foo, Clemens Mewald, Denis M. Baylor, Eric Breck, Heng-Tze Cheng, Jarek Wilkiewicz, Levent Koc, Lukasz Lew, Martin A. Zinkevich, Martin Wicke, Mustafa Ispir, Neoklis Polyzotis, Noah Fiedel, Salem Elie Haykal, Steven Whang, Sudip Roy, Sukriti Ramesh, Vihan Jain, Xin Zhang, and Zakaria Haque. 2017. Tfx: A tensorflow-based production-scale machine learning platform. In *KDD 2017*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4806–4813.

Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Luiz Rodrigues, Armando M Toda, Wilk Oliveira, Paula Toledo Palomino, Julita Vassileva, and Seiji Isotani. 2022. Automating gamification personalization to the user and beyond. *IEEE Transactions on Learning Technologies*, 15(2):199–212.

Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. 2005. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(9).

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Andrés Villa, Vladimir Araujo, Francisca Cattan, and Denis Parra. 2020. Interpretable contextual team-aware item recommendation: application in multi-player online battle arena games. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 503–508.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503.

Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 580–588.