Human Choice Prediction in Language-Based Persuasion Games: Neural Networks and Architectures Exploration

Shachaf Haviv Nitzan Manor
Technion
{shachafhaviv@, nitzan.manor@}
campus.technion.ac.il

Abstract

Deep Neural Network (DNN) models have the potential to provide new insights in the study of human decision making. In this paper we will explore various architectural frameworks, aiming to find new ways to improve recent research. Shapira et al. (2024) examined two distinct methodologies for forecasting human decisions within language-driven persuasion games, employing off-policy evaluation techniques. Notably and unexpectedly, the study found that predictive accuracy of LSTM models was higher than that of Transformer models. Building upon this groundwork, our research aims to broaden the scope by using a diverse array of approaches. We have manipulated the input data fed into these models, leveraging sub-word tokenization techniques, making it the Transformer's new input. Additionally, we examined LoRA technology, we assessed the feasibility of integrating pre-trained large-scale models and leveraging their knowledge to address our research problem effectively. However, we chose not to use LoRA for this study, a decision will be discussed in detail later in the paper. Due to this decision, we have focused on analysis of Transformer architectures, aiming to optimize their performance and trying to perform better accuracy with Transformers. Finally, we will compare the performance of these methodologies, and find the approach that yields the most precise predictions. We aspire to contribute to the field of human decision-making research.

1 Introduction

The Transformer architecture, introduced by Vaswani et al. (2023), has revolutionized the field of natural language processing (NLP). Its ability to handle long-range dependencies and parallelize training has made it the backbone of many state-of-the-art models, including BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020). Transformers leverage self-attention mechanisms to weigh the influence of different words in a sentence, thereby capturing context more effectively than traditional recurrent neural networks (RNNs) like LSTMs (Hochreiter and Schmidhuber, 1997).

In addition to the above, we explored the use of LoRA (Low-Rank Adaptation) technology. LoRA has attracted significant interest for its ability to efficiently fine-tune pre-trained models. By enabling rapid adaptation of large-scale models to specific tasks, LoRA offers the potential for substantial performance gains (Hu et al., 2021).

Our research objectives mainly include proposing and testing several changes to the transformer architecture, focusing on components such as positional encoding, normalization, residual connections and changing the loss function.

Another aspect of our work based on tokenization, a NLP preprocessing step, offers various techniques, each with distinct advantages. In the following sections of this paper, we will explore the differences between these techniques and discuss our chosen approach in detail.

2 Related Work

This paper continues the work of Shapira et al. (2024), who contributed to the field of predicting human choices in persuasion games by utilizing off-policy evaluation. We extend their research by exploring advancements to the Transformer architecture described in their study.

The improved multi-stage persuasion game was introduced by Shapira et al. (2024) based on Apel et al. (2022) game. The game is played between an expert (sender) and a decision-maker (DM), where in this setup, the expert acting as a travel agent, tries to convince the DM to choose a randomly selected hotel over R rounds. The expert has access to both the written reviews and the scores of the hotels, while the DM only sees the chosen review text. The DM aims to maximize their points by

correctly identifying good hotels (average score $\geq$ predefined threshold) based on the expert's recommendation.

In order to predict the DM's decision, Shapira et al. (2024) implemented three machine learning models, including a Transformer model (Vaswani et al., 2023) that receives as input the representation of all rounds up to round t.

Various ideas have been tested and implemented to enhance the capabilities of Transformers. As example, several works explore positional encoding methods (in order to capture word order within a sequence), including Shaw et al. (2018) who propose relative positional encodings, and Vaswani et al. (2023) who introduce sinusoidal positional encodings.

He et al. (2016) pioneered residual connections, demonstrating their effectiveness in deep neural networks. Works such as Ebrahimi and Abadi (2018) analyze the impact of different residual connection configurations within Transformers, further exploring their influence on model performance.

Janocha and Czarnecki (2017) examine how specific selections of loss functions influence deep models and their learning dynamics, as well as the resulting robustness of classifiers to different factors.

Another change we have tested is the removal of a fully connected layer before feeding the input into the Transformer. This approach is similar to that proposed by Wu et al. (2020) of a Lite Transformer model that simplifies the traditional Transformer architecture by removing or modifying components to achieve a lightweight model.

3 Data

Shapira et al. (2024) dataset consists of 87k decisions from 245 DMs who played against 12 different automatic expert bots (each DM played against 6 bots).

3.1 Tokenization

Tokenization is a pre-processing step in natural language processing tasks, including those involving Transformer architectures. Several tokenization techniques have been proposed, each with its advantages and limitations. Word-based tokenization methods split input text into individual words or word stems, offering simplicity and intuitiveness. However, they may struggle with out-of-vocabulary words and fail to capture subword-level information. Subword-based tokenization methods, such as Byte Pair Encoding (BPE), address these limitations by breaking down words into smaller subword units. This approach enhances the model's ability to handle rare words and increases vocabulary coverage. Character-based tokenization treats each character in the input text as a separate token. It may struggle with capturing word-level semantics and require larger model sizes due to increased input dimensionality (Gurugubelli et al., 2024).

In our study, we employed several tokenization options, including BERT, GPT-2, and RoBERTa (all sub-word based), to process the reviews data for each hotel. While there are numerous tokenization methods available, we selected these options due to their widespread usage and availability. The tokenized reviews were then concatenated with the game features. A challenge we encountered was the variation in sequence lengths resulting from different tokenizers. To address this issue, we incorporated additional features, such as `reaction_time_bins` [1], to align the sequence lengths with the requirements of our transformer model. Specifically, we added one `reaction_time_bins` feature for BERT and two for RoBERTa and GPT-2, ensuring compatibility with our model architecture (that includes 4-heads).

Another challenge we faced was the inconsistency in the sentence's length returned from the tokenizer. To address this issue, we employed padding techniques. Specifically, we experimented with padding using both -inf and 0.

3.2 EF

We used the Engineered Feature (EF) introduced by Apel et al. (2022) and extended by Shapira et al. (2024) to represent hotel reviews. These binary features capture a range of review attributes, including both discussed topics and stylistic characteristics.

4 Model

4.1 LoRA

We have considered the use of LoRA (Low-Rank Adaptation) technology in our study. LoRA has attracted attention for its potential to fine-tune pre-trained models efficiently. LoRA operates by decomposing the weight matrices of the model into low-rank components, thus significantly reducing

[1] Feature description - DM Reaction time in range r seconds: r $\in$ [0, 0.5), [0.5, 1), [1, 2), [2, 3), [3, 4), [4, 6.5), [6.5, 12), [12, 20), [20, ∞)

the number of parameters needed for adaptation. This decomposition facilitates faster adaptation of large-scale models to specific tasks, potentially resulting in substantial performance improvements. However, one major challenge we encountered was the difficulty in creating a proper train and test set that LoRA could utilize effectively. This task requires data to be provided to the model in time periods, ensuring that future rounds in the game are not revealed prematurely. Additionally, constraints related to hardware resources further hindered our ability to experiment with LoRA. Hence, despite its potential benefits, we ultimately decided not to utilize LoRA in our final project. Consequently, we focused our efforts on alternative methodologies that aligned more closely with the scope and requirements of our study.

4.2 Transformer Advancements

Our transformer model incorporates several enhancements to improve its performance and flexibility.

4.2.1 Positional Encoding

Integration of positional encoding, a technique for capturing the sequential information of input tokens. This allows the model to better understand the relative or absolute positions of tokens within a sequence, which is particularly beneficial for tasks involving sequential data (Vaswani et al., 2023). We implemented the positional encoding for a token at position i, dimension d and maximum sequence length D as suggested in Vaswani et al. (2023):

$PE(i,d) = sin(\frac{i}{10000^{2d/D}})$ *if d is even*

$PE(i,d) = cos(\frac{i}{10000^{2d/D}})$ *if d is odd*

4.2.2 Residual Connection

Residual connections facilitate the flow of information through the network by reducing the vanishing gradient problem and promoting deeper model architectures He et al. (2016). By using this approach we should enhances the model's representational capacity and enables more effective training.

4.2.3 Normalization

We incorporate the option for norm first, a configuration that allows for greater control over the normalization process within the transformer layers (Ba et al., 2016).

4.2.4 Loss Function

To potentially improve model accuracy, we investigated the use of an alternative loss function, Cross Entropy (CE) instead of Negative Log Loss (NLL) which used in Shapira et al. (2024). CE measures the difference between the model's predicted probability distribution and the true distribution.

4.2.5 Pre-Transformer FC Layer

We have added the option to exclude the fully connected layer before the transformer encoder (in contrast to Shapira's original paper) reducing computational complexity.

5 Experiments

We conducted a comprehensive evaluation to determine the optimal configuration for the Transformer model by systematically testing various permutations of additions and modifications.

To explore the impact of different features through parameter sweeps, we incorporated True/False options for each feature introduced in the last section into the model configuration. Our testing encompassed all conceivable permutations across 3 unique seeds, assessing their effects alongside our selected data representation methodologies, including original paper EF representation.

5.1 Phase 1: Exploring Data Representations and Network Architectures

We evaluated the impact of different data representations for the reviews: Engineered Features (EFs) and Tokenization. We experimented with incorporating the following elements into the base model architecture: residual connections, fully connected (FC) layers and positional encoding. To comprehensively assess the influence of each data representation and network architecture modification, we examined all possible permutations (the presence and absence of each element). Each permutation was trained for 10 epochs with 3 different seeds (1-3). This phase required approximately one week of computation time.

5.2 Phase 2: Hyperparameter Tuning with EFs

In this phase, our attention was directed solely towards the EF representation due to the low accuracy rate achieved through tokenization (further details on this matter will be presented in the results section). We incorporated a loss function (CE) during

training to guide the model optimization. Additionally, we employed dropout regularization to prevent overfitting. These enhancements were tested alongside the changes evaluated in the previous stage to determine their combined effect on the model's performance. This phase required approximately one week of computation time.

5.3 Phase 3: Refining the Best Configuration

To refine our findings, we conducted an additional round of testing focused on the 3 best-performing configurations from phase 2. This round involved extending the number of epochs to 20 to allow the models more time to converge and achieve optimal performance. Additionally, the number of seeds was increased to 6 to enhance the reliability and reproducibility of our results. This phase required approximately two days of computation time.

6 Results

In this analysis, the results of each model are presented as the mean of the average accuracy per player (DM) and the Sender's strategy. This is achieved by averaging across DMs rather than individual decisions. This approach prevents potential bias caused by certain human DMs participating in more games than others, thus ensuring a fair representation across all players.

6.1 Phase 1

All tokenization techniques, except EF's, result in low accuracy, averaging approximately 0.53. The transformer model showed no improvement over the epochs across all permutations, as detailed in the previous section. We attribute this to the model's inability to handle the large dimension of the input data we've tokenized and padded. Consequently, going forward, only results related to EF's are considered due to its achieved scores. Figure 1 in appendix B presents the confidence interval plot and illustrate the permutation results over 3 seeds and 10 epochs. The notable observation is that the model achieved better accuracy without a fully-connected layer preceding the transformer (as shown by the four leftmost permutations in Figure 1). This suggests that the transformer gained a deeper understanding of the interconnections among the input data features.

6.2 Phase 2

Using only the EF method, we evaluated the influence of different dropout rates (0.1, 0.2, 0.3) and loss functions (NLL, CE). Figure 2 in Appendix B presents accuracy as a function of the number of epochs for each combination of loss function and dropout rate. Consistent with the results from phase 1, the top 4 lines represents permutations without a fully connected layer before feeding the Transformer, making them achieving higher accuracy for all cases. It appears that omitting the fully connected layer, positional encoding, and residual connections consistently yielded higher accuracy, along with configurations that used only residual connections. Moreover, there is not significant difference of the model performance between loss functions.

6.3 Phase 3

Following last phases, we chose the best 3 configurations (which achieved the highest accuracy) and re-run it with 20 epochs and 6 seeds. Figure 3 in Appendix B presents accuracy as a function of the number of epochs for each configuration (Table 1 shows the parameters of each configuration).

Table 1: Configurations

Name	fc	res	pe	dropout	loss
Configuration 1	X	X	X	0.1	CE
Configuration 2	X	X	V	0.1	NLL
Configuration 3	X	V	X	0	NLL

'fc' - fully connected layer, 'res' - residual connections, 'pe' - positional encoding

Increasing the number of epochs appears to have a negative effect on the accuracy rate, particularly for configuration 3. The maximum accuracy tends to be reached around epoch 10 for all selected configurations. Final results and performance is shown in Table 2 in appendix A.

7 Conclusions

Our study aimed to enhance predictive accuracy in human decision-making using deep neural network models. By experimenting with Transformer architecture, we discovered that omitting fully connected layers and the change of the loss function and dropout value can enhance the accuracy. It's evident from our study that among the tokenization techniques tested, the use of EF's yielded the best results.

References

Reut Apel, Ido Erev, Roi Reichart, and Moshe Tennenholtz. 2022. Predicting decisions in language based persuasion games. *Preprint*, arXiv:2012.09966.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *Preprint*, arXiv:1607.06450.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Mohammad Sadegh Ebrahimi and Hossein Karkeh Abadi. 2018. Study of residual networks for image recognition. *Preprint*, arXiv:1805.00325.

Krishna Gurugubelli, Sahil Mohamed, and Rajesh Krishna K S. 2024. Comparative study of tokenization algorithms for end-to-end open vocabulary keyword detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12431–12435.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. *Preprint*, arXiv:1603.05027.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Katarzyna Janocha and Wojciech Marian Czarnecki. 2017. On loss functions for deep neural networks in classification. *Preprint*, arXiv:1702.05659.

Eilam Shapira, Reut Apel, Moshe Tennenholtz, and Roi Reichart. 2024. Human choice prediction in language-based persuasion games: Simulation-based off-policy evaluation. *Preprint*, arXiv:2305.10361.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *Preprint*, arXiv:1803.02155.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2020. Lite transformer with long-short range attention. *Preprint*, arXiv:2004.11886.

A Statistics

Table 2: Model Performance

Configure	Mean	Std	CI(95%)
1	**0.8383**	**0.00171**	**[0.8371, 0.8396]**
2	0.8366	0.00079	[0.8361, 0.8372]
3	0.8374	0.002451	[0.8357, 0.8390]

Configuration 1 (As described in Table 1) achieved the best confidence interval

B Plotted Results

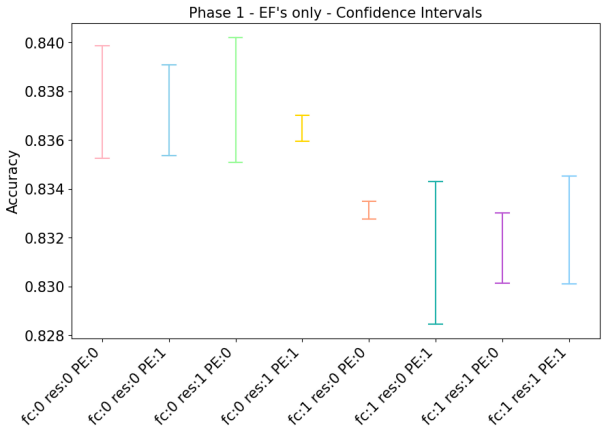

Figure 1: Confidence Interval for each permutation using only EF representation

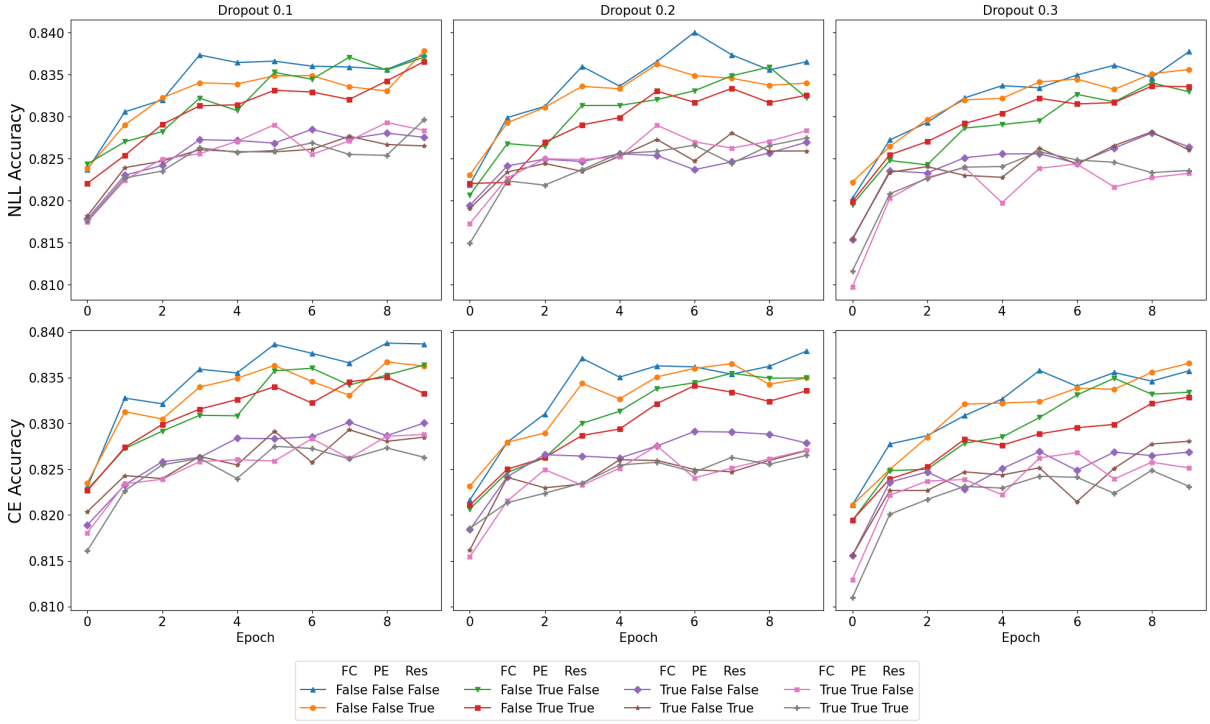

Figure 2: Model performance as a function of the number of epochs for each combination of loss function and dropout rate

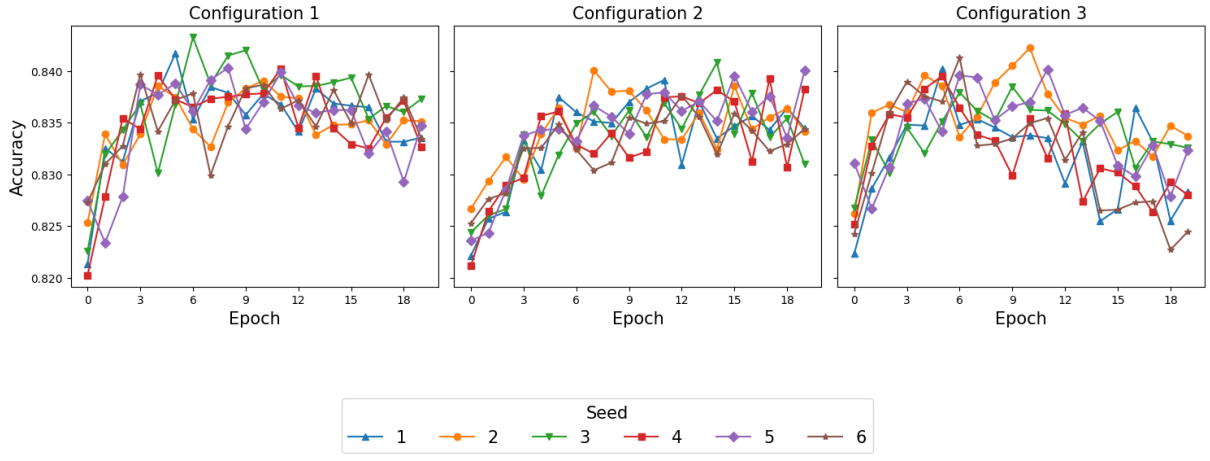

Figure 3: Model performance as a function of the number of epochs for each configuration.