Methods In Natural Language Processing - Final Project
Lecturer: Dr. Eyal Ben David

Ariel Cohen and **Dan Israeli**
Technion - Israel Institute of Technology

Abstract

The task of off-policy evaluation (OPE) is highly important in the domain of game theory. Many studies have taken interest in this task, and particularly Shapira et al. 2024. This paper serves a direct continuation of the mentioned research, by trying to improve the results using various tools from the NLP domain. Shapira et al. 2024 demonstrates how the use of simulated data to enrich the original dataset, improves the performance in an OPE task. A key part in the simulated data quality, as well as the simulation's success, was the incorporation of a language-based strategy. This strategy attempts to mimic the interaction between humans and textual information. In our paper, we extend this approach by introducing new language-based strategies. As opposed to the general approach presented in (Shapira et al., 2024), each of our suggested strategies focuses on different key components of the given textual information. We evaluate our strategies, and compare the highest performing one with the results presented in the original paper. Furthermore, Shapira et al. 2024 used several techniques to embed said textual information. In order to extend the scope of our research, this paper also experiments with similar, additional embedding methods. We examined their effect on the model's performance and compared it with the best results achieved in the original paper.

Introduction

In the original paper, Shapira et al. 2024 emphasized the importance of OPE in non-cooperative persuasion games. As widely known, OPE involves the prediction of human agents' actions with a given partner, based on their actions with different ones. This task is of high importance since it allows to better understand human nature regarding decision making. Thus, it holds many practical applications (e.g. recommendation systems, human-AI interactions, etc.). However, this nature is known to be extremely complex and greatly influenced by an abundance of exogenous variables. Hence, it is not an easy task for one to accomplish. Shapira et al. 2024 attempts to capture this notion by introducing four strategies:

- Random – mimics the random noise in human decision making.
- Trustful – mimics the human consideration of past behavior and its outcome.
- Language-based – mimics the human utilization of textual information.
- Oracle – mimics the human learning process and improvement upon a given task.

Although this approach might seem quite simple, it yields very impressive results (as reported in the original paper). As our primary research, this paper attempts to further improve these results by implementing new strategies as a means to capture said notion in a more natural and accurate manner. The language-based strategy presented in Shapira et al. 2024 uses the textual information as a whole. That is, the presented strategy does not make any distinction between the key components of the given information. However, it is reasonable to assume that people perceive textual information by granting various degrees of importance to the different components. Thus, in order to reflect this assumption, our paper focuses on modifying the language-based strategy. This approach aims to provide a more accurate modelling of human textual information utilization, and thus, reflect a more natural representation of the human decision making process.

As our secondary research, to engage with the original paper from multiple aspects, we chose a different direction to explore – textual embedding methods. One of the embedding methods used in the original paper involved the BERT embedding model, while performing PCA as a means to reduce the initial embedding dimensionality. This paper examines the prediction model's performance as a function of the PCA dimension size. This examination allows us to discover quite interesting insights, such as finding the optimal BERT embedding with respect to the performance/dimensionality ratio. In addition, we compare this optimal one with the best results achieved in the original paper.

Related Work

As mentioned above, this paper heavily relies on Shapira et al. 2024. Particularly, we replicate the experiments in the original paper, with multiple modifications that were discussed in the Introduction section (and will be further discussed in the Experiments section). That is, we use the same data, model architectures, simulation process and so on. This approach allows us to receive results that are comparable to those presented in the original paper. Thus, our findings and insights would be applicable.

Data

This paper uses the same data as that collected in Shapira et al. 2024. The data contains human DM actions when playing the following game: each DM plays successively with a series of rule-based experts. Each game is played with a single expert, and consists of 10 rounds. In each round, a hotel is chosen randomly from a predetermined pool. Then, the expert chooses a hotel review to present to the DM (from the collected ones), based on its designated strategy. Finally, the human DM has to conclude if the suggested hotel is "good", and thus decide whether to go or not, accordingly. The human DM's goal is to go only to "good" hotels, and the expert's goal is to sell as many hotels as possible. Note that this is indeed a non-cooperative persuasion game. For the full description regarding the exact the hotels and DMs data, please refer to the Data Appendix

Model

As mentioned in the previous section, throughout our research, we use model architectures identical to those in the original paper. Let us elaborate on the each of the two architectures used:

Transformer (Vaswani et al., 2017) - for each game, the model's input is the representation of all rounds up to the current round, inclusive. This architecture allows for information transfer between successive rounds within the same game.

LSTM – Long Short Term Memory Model (Hochreiter and Schmidhuber, 1997) - whenever a new DM action is being predicted, the model's cell state is initialized to a parameter vector that was learned during the training procedure. Then, during the transition between rounds, as well as games (played by the same DM), the hidden state is propagated onwards. As opposed to the previous model, this unique architecture allows us to capture the relationship not only between successive rounds, but also between successive games of the same DM.

Primary Research

Experiments

As discussed, our focus is on modifying the language-based strategy presented in the original paper, to reflect the emphasis on the different key components of the textual information (hotel review). Shapira et al. 2024 implemented the language-based strategy by utilizing an LLM, specifically the Text-Bison model (Anil et al., 2023), as follows: The model was prompted to rate a hotel based on its entire review (both positive and negative parts) between 1-100 scale, where a "good" hotel is considered to have a score of at least 80. Then, for each review, the probability of the hotel being considered "good" was extracted from the model's answer. We consider the same approach, but directly prompt the LLM model to output the desired probabilities. In addition, we chose to utilize a newer and more advanced LLM model – GPT-3.5 Turbo (via the OpenAI API). In order to define language-based strategies which reflect our goal, we prompt the LLM while only providing either of the positive or negative parts of the review. In addition, we also prompt the LLM while providing both parts of the review, to be able to create a baseline strategy which resembles the one in the original paper. As a result, for each review we received 3 different probabilities – one for each review component provided. For an in-depth review of the prompting method, please refer to the LLM Appendix. After receiving the different probabilities, we were able to define the following basic language-based strategies:

- Baseline – focuses on the entire given review (as a whole). This definition aims to embody a person with neutral perspective.
- Strictly Positive (S-Pos) – focuses only on the positive part of the given review. This definition aims to embody a highly optimistic person.
- Strictly Negative (S-Neg) – focuses only on the negative part of the given review. This definition aims to embody a highly pessimistic person.

In order to implement the above strategies, we used the same technique used in the original paper, as explained in the Strategies Appendix. After defining the based strategies, we were able to use them to receive additional and more complex language-based strategies:

- Positive Oriented (Pos-O) – this strategy focuses twice as much on the positive part than the negative part of the given review. This definition aims to reflect a moderately optimistic person.
- Negative Oriented (Neg-O) – this strategy focuses twice as much on the negative part than the positive part of the given review. This definition aims to reflect a moderately pessimistic person.
- No Orientation (NO) – this strategy equally focuses on the positive and negative parts of the given review. This definition also reflects a neutral perspective person (as the Baseline strategy), but implemented differently.
- No Orientation Extended (NOE) – this strategy equally focuses on the positive part, negative part, and the entire given review. This definition aims to reflect the intricacy of the human perspective – optimistic, pessimistic and neutral.

For the technical description of the newly defined strategies above, please refer to the Strategies Appendix. After establishing our new language-based strategies, we evaluate them as follows: we run each strategy on the mentioned LSTM model, for 20 epochs, with simulation ratio 2.0. Moreover, since we used new strategies, we did not want to rely on the optimal improvement rate presented in the original paper. Thus, for each strategy, we tested different improvement rate values in order to find its best fitting one. In addition, as a means to minimize the influence of randomness on the received result, we run each strategy and improvement rate combination with 3 different random seeds, and average the results across them. Then, we cross-compare the optimal results of each strategy (using its optimal improvement rate) in order to find the best one[1]. After finding the best strategy, we compare its performance with the results in the original paper as follows: we run the best strategy on both models (LSTM and transformer) for 25 epochs, with the optimal improvement rate found, and across the 6 simulation ratios tested in the original paper (0.0, 0.5, 1.0, 2.0, 4.0, 10.0). Moreover, we use the same randomness-effect minimization method used in the previous procedure.

Results

From Figure 1, we can clearly conclude that the best strategy among the newly defined ones is the S-Pos strategy (from epoch 6 onwards). This result is very intriguing. The simulation's purpose is to improve the model's performance by enriching the original dataset with additional examples from the same distribution. Since the model performs best with the S-Pos strategy, we can conclude that the simulation data quality (with this strategy) is superior. Hence, the S-Pos strategy reflects the decision making process of participating human DMs most accurately. From the definition of this strategy, we can assume that participating humans in the original dataset utilized the given textual information in a highly optimistic manner. That is, they gave great importance to the positive part of the review (compared to the negative part) while making their decisions. Then, after obtaining the best strategy, we compared its performance across the mentioned simulation ratios, in order to find the best fitting one. As seen in Figure 2 and Figure 3, the best results of the LSTM and Transformer models are achieved with simulation ratio 2.0 for both. We compared the results with the best ones in the original paper (achieved with simulation ratio 4.0). Unfortunately, as can be seen in Figure 4 and Figure 5, our performance under both models is inferior to that of the original paper. This result is quite surprising, since we used a more advanced LLM in order to implement our new language-based strategies. A possible explanation could be the different method used to receive the probabilities from the textual information. Instead of extracting the probabilities from the LLM's output, we directly query the LLM for them. Since our query is non-standard,

[1] Note that to correctly choose the best strategy, we do not evaluate their performance on the test set. Instead, we split the train set into a sub-train set and a validation set, and use these sets in the evaluation process.

the LLM might have struggled to answer it appropriately. In order to further analyze our strategy's performance, we compare its results across the different mentioned simulation ratios, with the corresponding ones presented in the original paper. For the performance comparison under the LSTM and Transformer models, See Figure 6 and Figure 7, respectively. In Figure 6, we can see an interesting phenomenon: for all simulation ratios (apart from 10.0) the model's performance does not decay in larger epochs under our strategy, as opposed to the original one. In other words, our strategy allows to simulate data that increases the model's robustness to overfitting. This phenomenon is especially apparent in Figure 6.3 (simulation ratio 1.0). This phenomenon can also be seen in Figure 7 (for the Transformer model), but more subtly. In addition, as a result of the phenomenon discussed above, for most simulation ratios (4 out of 6), the LSTM model yields better results in later epochs when utilizing our strategy as opposed to the original one. Unfortunately, this does not hold for the Transformer model (only holds for 2 out of 6 ratios). Moreover, as can be seen in Figure 7.5 (simulation ratio 4.0), the model's performance when using our strategy seems to have potential to increase in later epochs (even after 25). This is in contrast to the model's performance when using the original strategy, which appears to have converged in those epochs.

Secondary Research

Experiments

As mentioned, our secondary research attempts to capture the impact of the BERT PCA embedding dimensionality (of the textual information) on the model's accuracy. As a means to find the PCA dimension range that we experiment within, we search for a range which significantly reduces the dimensionality while also preserving as much information as possible. We achive that by using the Elbow method to balance the explained variance VS dimensionality trade off. Then, after receiving said range, we experiment with different values within that range and test how they affect the model's performance, as follows: we choose the best performing model architecture and simulation parameters from the original paper[2], and compare the accuracy received when using each of the different dimension values. In addition, we use the same randomness-effect minimization method used in the Primary Research section. Moreover, let us mention that we use the same method to receive the BERT embeddings as used in the original paper. For further information regarding the embedding process, please refer to the BERT Embedding Appendix.

Results

Following the execution of the Elbow method, as can be seen in Figure 8, the balance point is received for PCA of dimension 200 (captures over 80% of the explained variance). As a result, we choose to focus on the dimension range of 1 - 250. In order to obtain a reasonable number of experiments, we choose to test the PCA dimensions 5, 10, 15, ... 250. After performing the experiments, as can be seen in Figure 9, it appears that the graph's balance point is received around dimension 50. Furthermore, not only is the said dimension the apparent balance point of the trade off, it also appears to achieve the best performance. In particular, it performs better than the balance point of the explained variance trade off. This result is quite surprising to us, since it achieves better performance, despite capturing less variance. In addition, it appears from Figure 9 that the original paper's best result outperforms our best result. That is, the EFs embedding method used in the original paper is superior to all the embedding methods we tested. As shown in the original paper, the EFs embedding method with embedding size of 36, outperforms the BERT PCA method with the same dimension size. This paper manages to show that it is also better than the BERT PCA method with a much higher dimension size (up to 250). That is, although the EFs embedding dimension is approximately 7 times lower, it still manages to represent the information more accurately. As a result, we can conclude that the EFs embedding method is far more compact, yet without sacrificing performance.

Acknowledgments

As mentioned, this paper builds upon the article Shapira et al. 2024, and extends the findings by covering new and diverse multiple aspects.

The full data and code used in this project are provided in the following GitHub repository.

[2] This includes using the mentioned LSTM architecture, simulation ratio of 4.0, improvement rate of 0.01, and the EFs embedding method presented in the original paper.

References

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Shapira, E., Apel, R., Tennenholtz, M., and Reichart, R. (2024). Human choice prediction in language-based persuasion games: Simulation-based off-policy evaluation.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Data Appendix

Below is the full description of the data used in this paper (same as in the original paper):

The Hotels - the hotel reviews dataset was collected from Booking.com. It contains data of 7 reviews for each hotel, for a total of 1,068 hotels (7,476 reviews overall). Each review contains a numerical score (on a scale of 1 to 10), as well as two textual parts: positive and negative. Additionally, a hotel is defined as "good" if the average numerical score across its collected reviews is at least 8. The median score of hotels in the dataset is 8.01 (that is, about half of the hotels are considered as "good", and half as "bad").

Interaction Data - the human DMs actions data was collected using a dedicated app. Each DM was assigned to one of two groups, each consisting of 6 different experts. A total of 210 DMs were assigned to the first group, where 71,579 decisions were collected. In addition, a total of 35 DMs were assigned to the second group, where 15,625 decisions were collected. The first group's data is used as the training set, whereas the second group's data is used as the test set. Since both groups contain different DMs and experts, we can see that this indeed reflects an OPE task.

LLM Appendix

In this paper, we used GPT-3.5 as the mentioned LLM. In particular, we used the 'gpt-3.5-turbo' model from the OpenAI API. We iterated over all the hotel reviews in the dataset, while providing its different parts to the LLM (positive, negative or both), depending on the strategy currently being tested. Then, for each hotel review, we prompted the LLM to output the probability for the hotel having a score of at least 8 on Booking.com. The prompts we used are as follows:

1. **System prompt:** used to provide context to the model as preparation for future queries.

   You are a hotel recommender on Booking.com. That is, you are going to receive a hotel review from that website. Based on the review you receive, you should give a probability, between 0 and 1, that the overall score of the hotel is at least 8. Your answer should only be the probability, in float format. Remember that the review represents a subjective experience.

   In addition, when only the positive part of the review was provided to the LLM, the following line was added to the system prompt:

   Note that the review you are going to receive is only the positive part of the full review.

   Similarly, when only the negative part of the review was provided to the LLM, the following line was added to the system prompt:

   Note that the review you are going to receive is only the negative part of the full review.

   These two additions were added in order to try and prevent the LLM from being biased after having been exposed only to a specific part of a review.

   Moreover, when both components were provided to the LLM, the following line was added to the system prompt:

   Note that in the review you are going to receive, the positive and negative parts will be separated.

   This addition was added in order reflect the structure of the data.

2. **User prompt:** used to query the model.

   Positive part:

   {positive part}

   Negative part:

   {negative part}

   Where ***{positive part}*** and ***{negative part}*** are the positive and negative components of the review, respectively. If only the positive component is provided to the LLM, ***{negative part}***='No negative part'. Similarly, if only the negative component is provided, ***{positive part}***='No positive part'.

Let us mention that the hotel reviews contain some degree of missingness. That is, some reviews are missing the positive or negative part. In such cases, we decided to sample the probability from a uniform distribution as follows: if the part given to the LLM is the positive one, and it is missing, we sample from the distribution $U[0.3, 0.5]$. Note that the expectation is 0.4, which is strictly lower than 0.5. That is, there is a higher probability to reject the hotel. This attempts to capture the notion that the missingness of the positive part might indicate a poorer hotel quality. Similarly, if the part given to the LLM is the negative one, and it is missing, we sample from the distribution $U[0.5, 0.7]$. Note that the expectation is 0.6, which is strictly greater than 0.5. That is, there is a higher probability to go to the hotel. This attempts to capture the notion that the missingness of the negative part might indicate a better hotel quality.

Strategies Appendix

Let us elaborate on the basic language-based strategies implementation below:

- Baseline - associated with the probabilities received by providing both parts of the review to the LLM.
- Strictly Positive - associated with the probabilities received by providing only the positive part of the review to the LLM.
- Strictly Negative - associated with the probabilities received by providing only the negative part of the review to the LLM.

For each strategy, we follow the same approach in the original paper: given a review, we use the received probability associated with the strategy to sample a Bernoulli random variable. Then, the decision whether or not to go to the hotel is based on its outcome (go to the hotel IFF the value is 1).

Now, before elaborating on the more complex language-based strategies, let us briefly discuss the initial playing probability term: this term is heavily used in the simulation process. At the start of a game, each of the simulated human DM core strategies is assigned an initial playing probability. The probability of the Oracle strategy (making only correct decisions) is initialized to 0. The rest of the DM strategies are initialized to a positive one. Then, in each round, all the mentioned probabilities are updated. This update is based on a stochastic procedure[3]. The more complex language-based strategies are implemented by using the initial playing probabilities as follows:

- Positive Oriented - combines both the Strictly Positive and Strictly Negative strategies, and gives the positive one twice the initial playing probability.
- Negative Oriented - combines both the Strictly Positive and Strictly Negative strategies, and gives the negative one twice the initial playing probability.
- No Orientation - combines both the Strictly Positive and Strictly Negative strategies, and gives both equal initial playing probabilities.
- No Orientation Extended - combines all the basic strategies, and gives them equal initial playing probabilities.

Note that in the original paper, the initial playing probabilities of the DM strategies are uniformly distributed across the Random, Trustful and LLM-based components. In all of our experiments, we follow this approach. In particular, we use the initial playing probability of the LLM-based component, and distribute it between the basic language-based strategies comprising it, based on the definition of the current strategy being tested.

Below are the initial playing probabilities for each of our defined language-based strategies:

Strategy Name/ Core Strategy	Random	Trustful	S-Pos	S-Neg	Baseline
Baseline	$1/3$	$1/3$	0	0	$1/3$
S-Pos	$1/3$	$1/3$	$1/3$	0	0
S-Neg	$1/3$	$1/3$	0	$1/3$	0
Pos-O	$1/3$	$1/3$	$2/9$	$1/9$	0
Neg-O	$1/3$	$1/3$	$1/9$	$2/9$	0
NO	$1/3$	$1/3$	$1/6$	$1/6$	0
NOE	$1/3$	$1/3$	$1/9$	$1/9$	$1/9$

*Note that the S-Pos, S-Neg and Baseline strategies fully comprise the language-based component of the simulated DM.

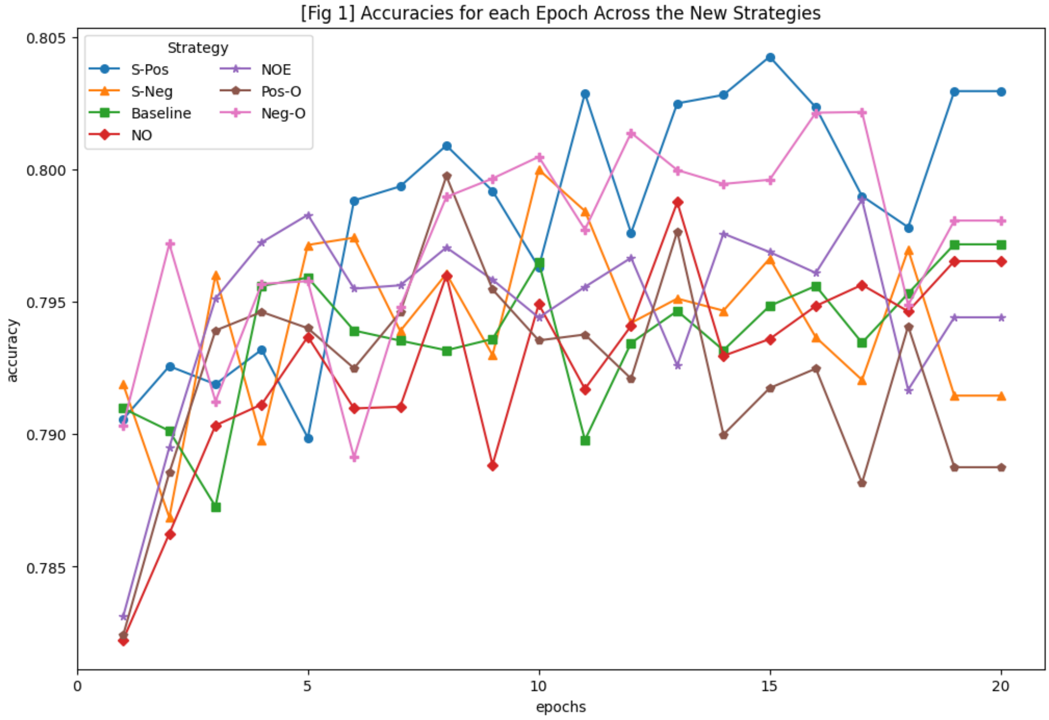
[3] The expected change of the Oracle strategy's playing probability at the end of the procedure is strictly positive. Thus, it guarantees the improvement of the simulated DM, which tries to capture the learning process of a real human.

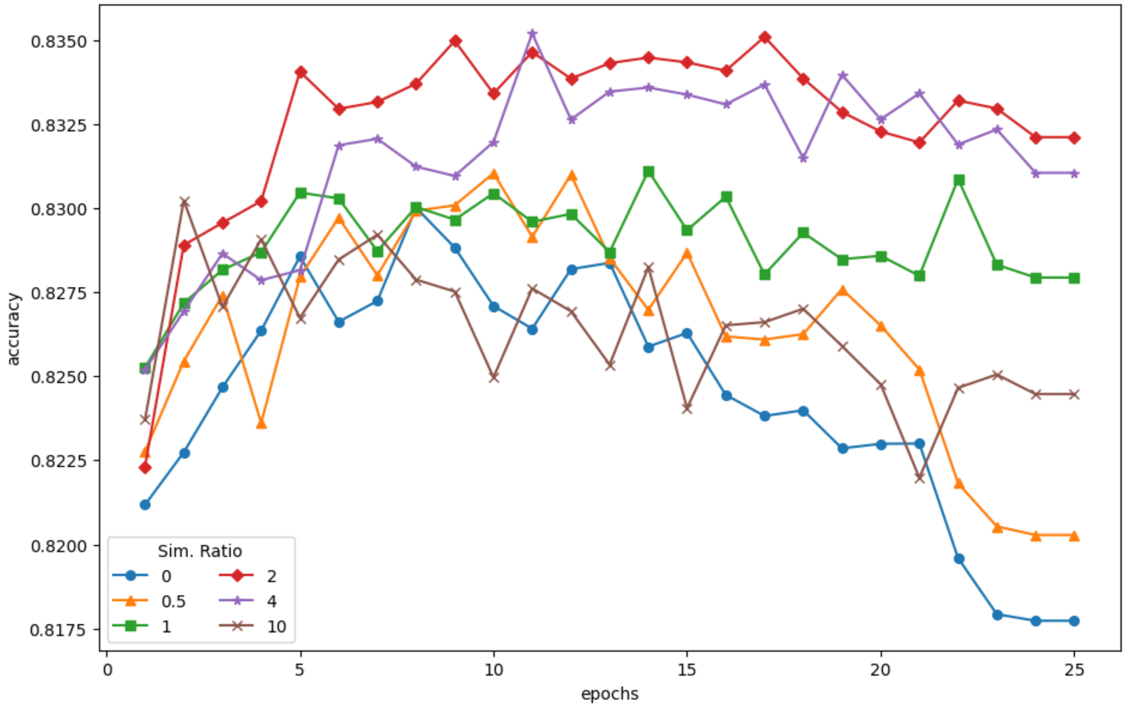
BERT Embedding Appendix

In order to create the BERT Embeddings, we used the 'all-MiniLM-L6-v2' embedding model from the Sentence Transformers Python library. This model takes a sentence as an input, and outputs a vector of dimension 384. Let us be reminded that every hotel review in the dataset consists of a positive part and a negative part. Thus, the embedding process is follows: we use the embedding model to embed both the positive and negative parts of the review separately. Then, we concatenate them into a vector of dimension 768. This vector represents the BERT embedding of the review.

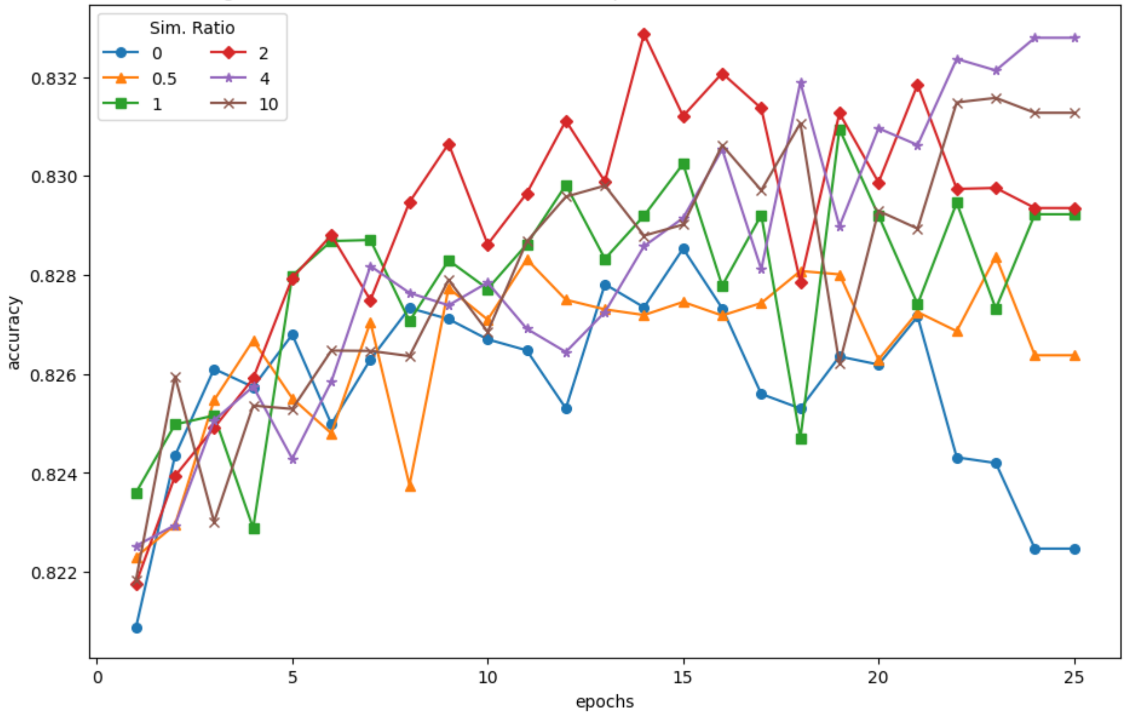
Plots Appendix

Below are the figures mentioned throughout the paper:



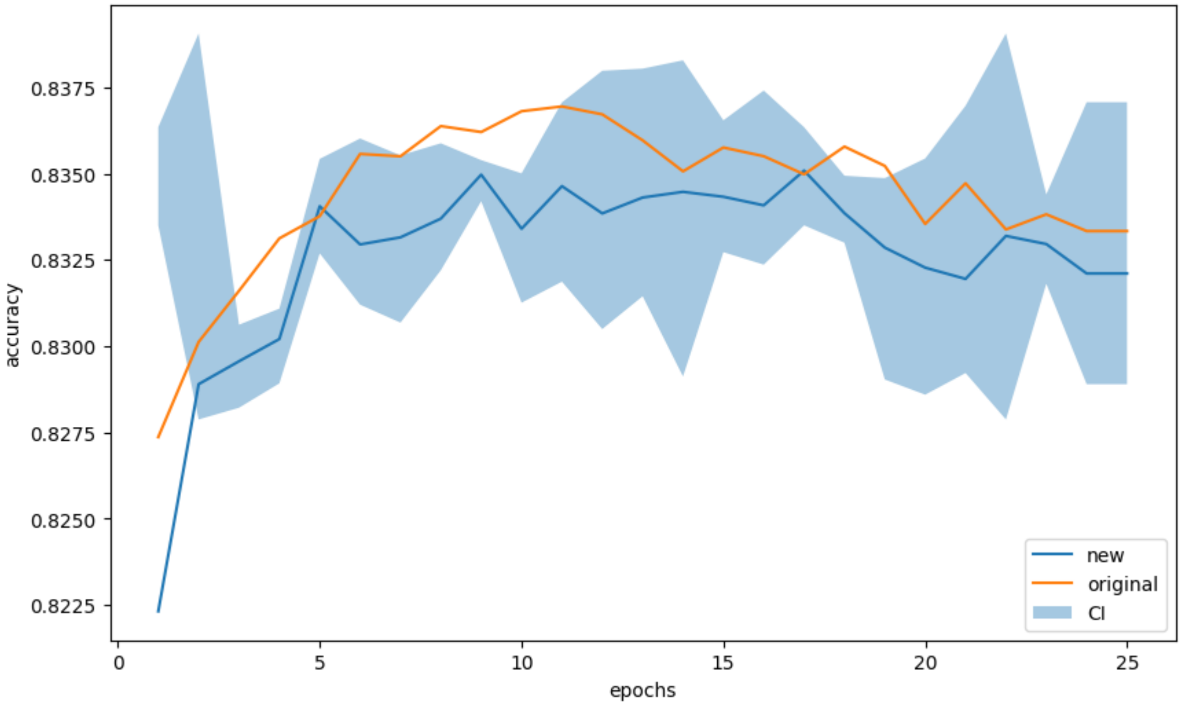


[Fig 2] LSTM Accuracies for each Epoch Across Different Simulation Ratios

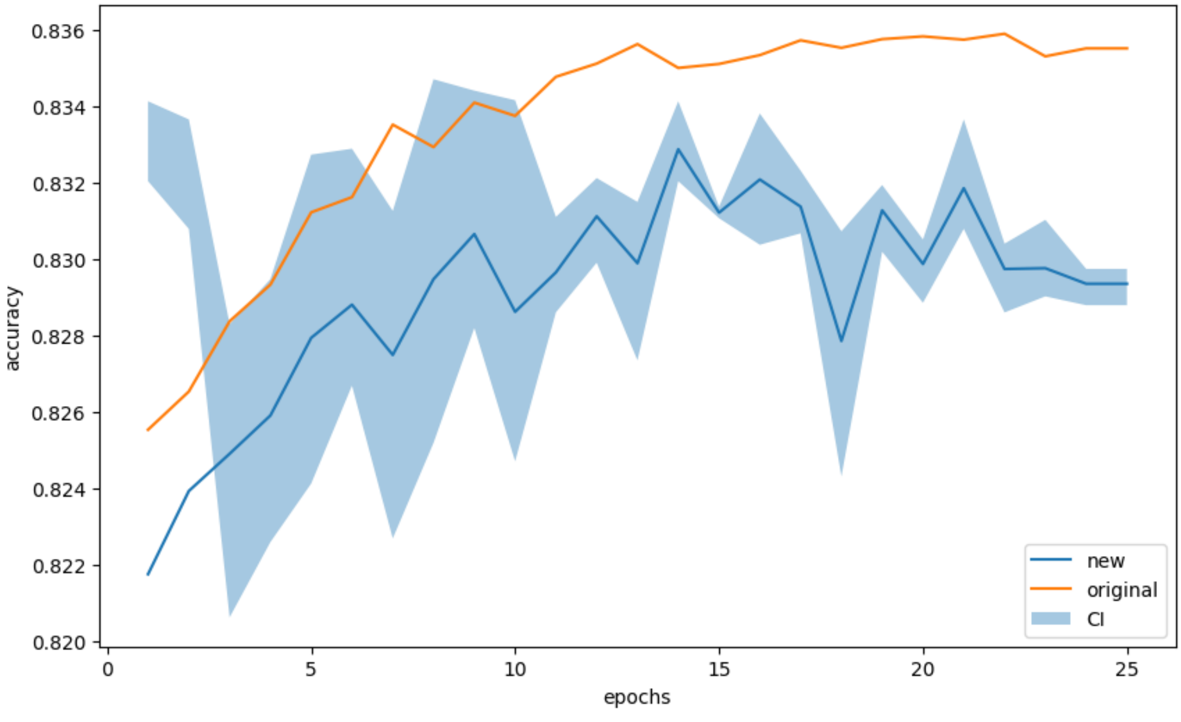


[Fig 3] Transformer Accuracies for each Epoch Across Different Simulation Ratios

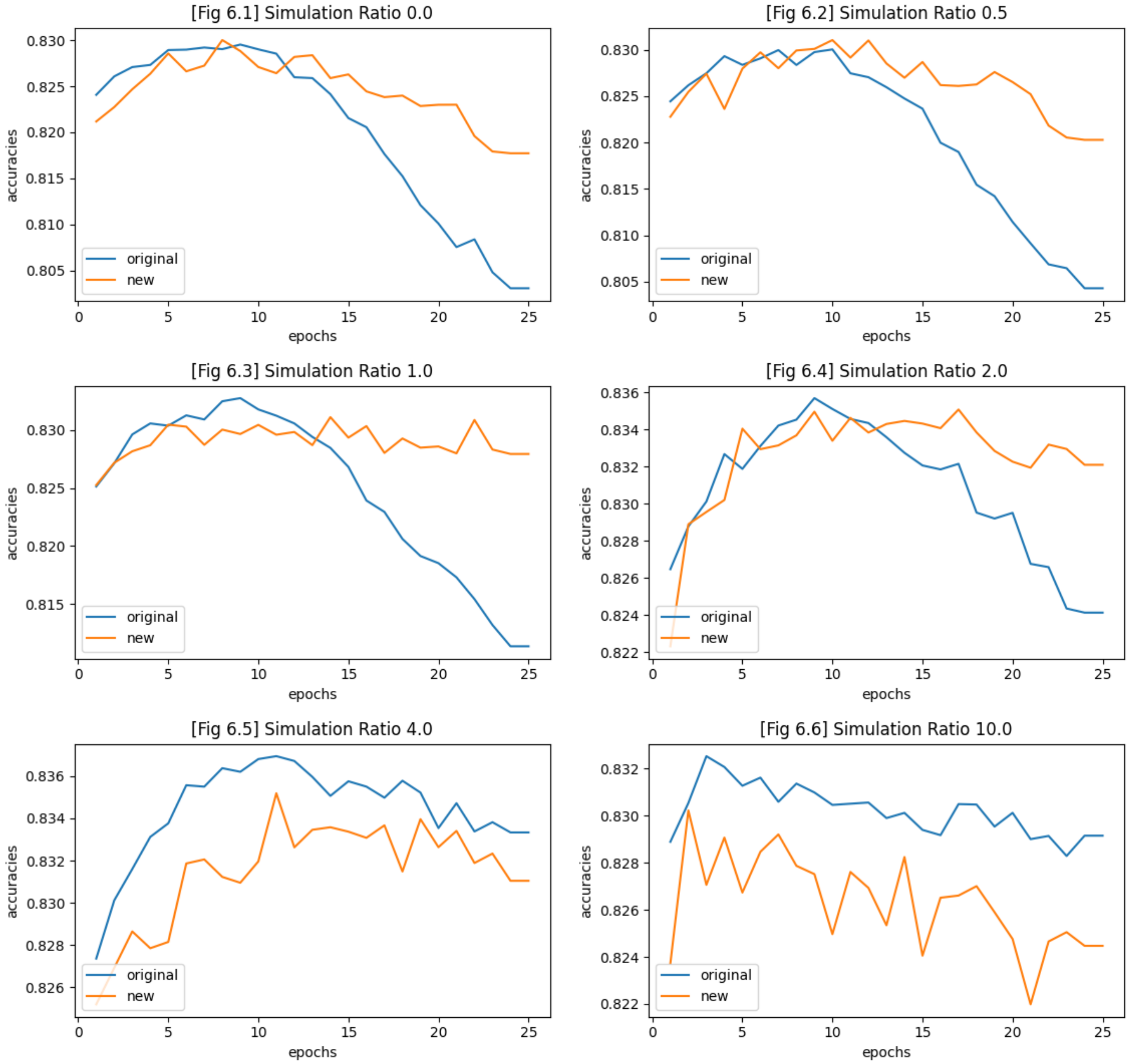
[Fig 4] Best Results Comparison - LSTM Model



[Fig 5] Best Results Comparison - Transformer Model



[Fig 6] LSTM Original VS New Accuracies across Different Simulation Ratios



[Fig 6.1] Simulation Ratio 0.0

[Fig 6.2] Simulation Ratio 0.5

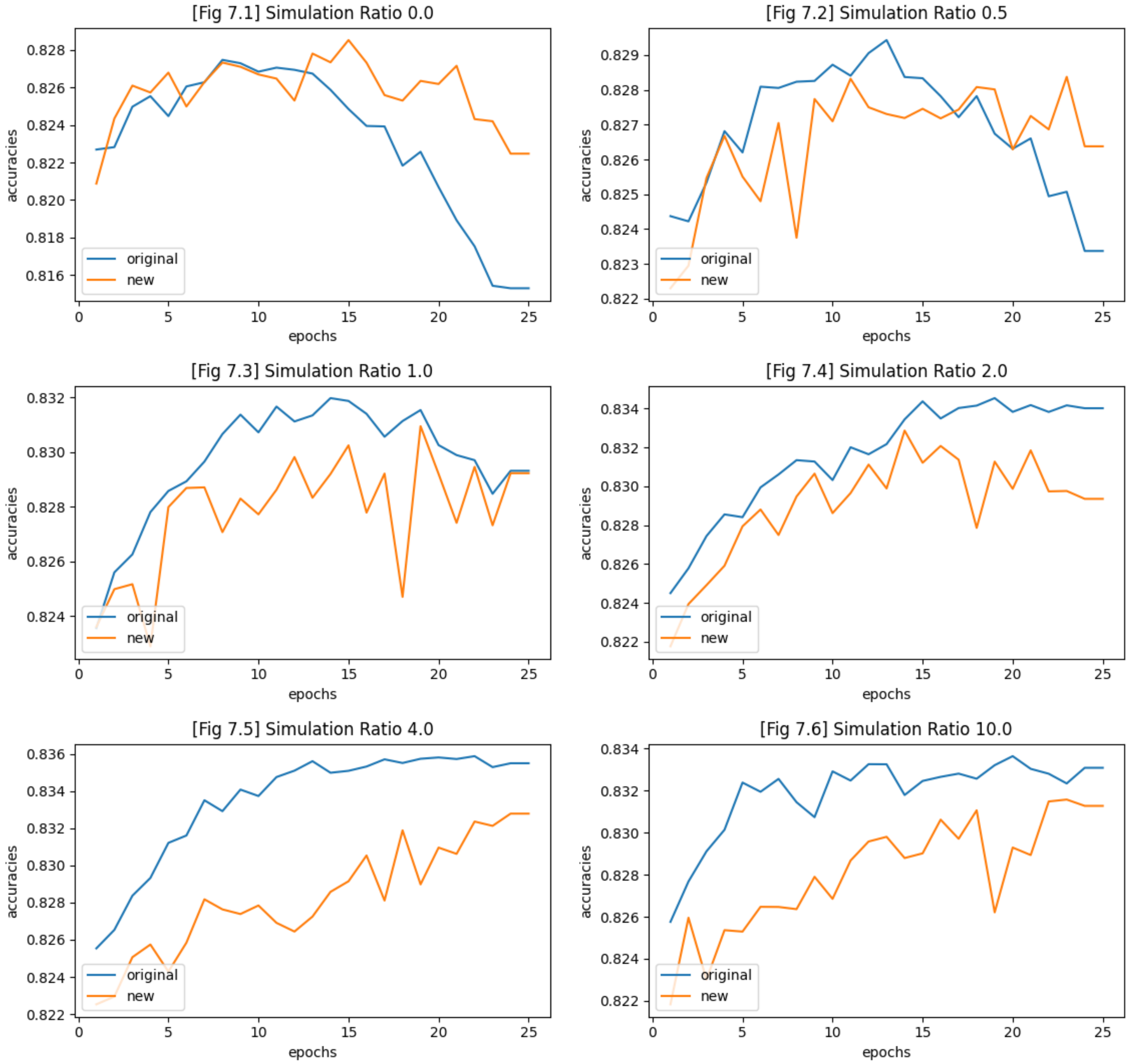
[Fig 6.3] Simulation Ratio 1.0

[Fig 6.4] Simulation Ratio 2.0

[Fig 6.5] Simulation Ratio 4.0

[Fig 6.6] Simulation Ratio 10.0

[Fig 7] Transformer Original VS New Accuracies across Different Simulation Ratios



[Fig 7.1] Simulation Ratio 0.0

[Fig 7.2] Simulation Ratio 0.5

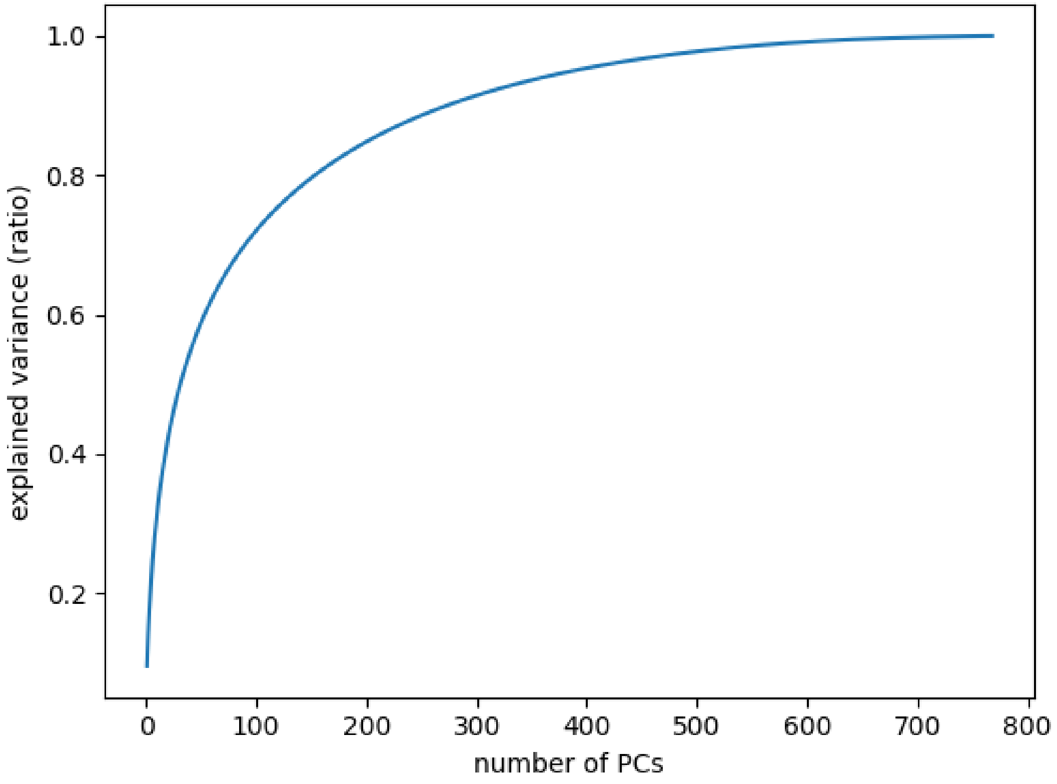
[Fig 7.3] Simulation Ratio 1.0

[Fig 7.4] Simulation Ratio 2.0

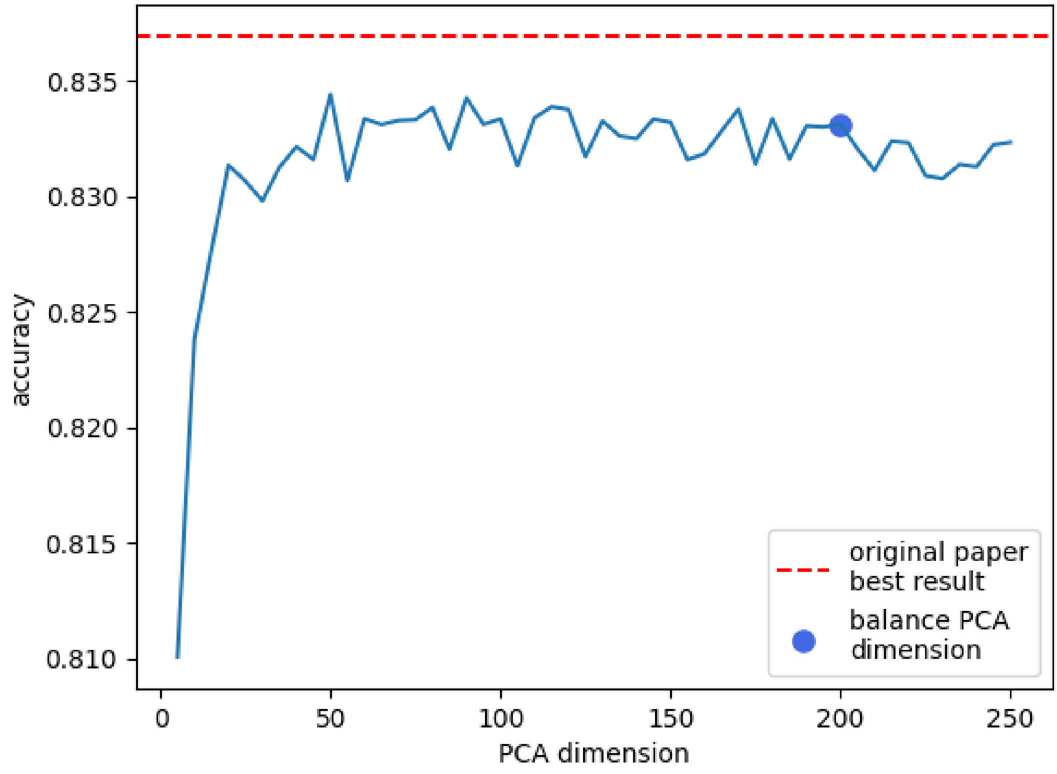
[Fig 7.5] Simulation Ratio 4.0

[Fig 7.6] Simulation Ratio 10.0

[Fig 8] Explained Variance Ratio as a Function of the Number of PCs



[Fig 9] Accuracy as a function of PCA Dimensions


accuracy
PCA dimension
original paper best result
balance PCA dimension