Impact of LLM on Reinforcement Learning

Anonymous ACL submission

Abstract

Text embeddings are essential for language understanding tasks. Large language models (LLMs) have recently become popular for text embedding due to their ability to capture complex information. Leveraging text-based adventure games as a test bed, we explore the impact of different language models on Reinforcement Learning (RL) behavior. The results show that contrary to common assumptions, larger embedding models do not guarantee better performance over smaller model sizes. Instead, the optimal model size depends on the specific game environment.

1 Introduction

Embedding techniques are crucial for all language-based tasks as they convert human language into a format that machines can understand (Kashyap et al., 2024). This paper focuses on how language embedding models affect reinforcement learning performance. Text-based adventure games provide a practical test bed for language-based RL agents. Figure 1 illustrates such a game, where the agent must make decisions based on its understanding of the state information and interact with the game environment to receive feedback. A critical step is selecting suitable embedding models to obtain state and action representations during RL training.

The motivation for focusing on embedding methods in RL agents in this paper is twofold: a) **Training Efficiency**: Training RL agents is time-consuming. We hypothesize that in many cases the performance difference between small and large models is minimal. Thus, opting for a smaller model can optimize the trade-off between performance and computing costs. b) **Performance Improvement**: A common approach to enhancing agent performance is by exploring new RL algorithms. However, language is semantically rich, and sentence representation is a fundamental step for language-based RL that enables the agent to make decisions based on syntactic and semantic understanding, rather than merely learning from feedback such as rewards from game environments. We hypothesize that selecting the right embedding methods can improve RL agent performance without modifying the RL algorithms themselves.

The contributions of this paper are:

- Our experiments compare whether the size of the language model influences RL performance and compare sentence embeddings to word embeddings with GRU as different embedding architectures.
- Additionally, we present the results of using only Llama 3 (AI@Meta, 2024) for playing the games without any RL-based training or fine-tuning.

2 Benchmarks and Framework

2.1 Text-based Adventure Games: Benchmarks

Jericho (Hausknecht et al., 2020) is one of the most popular benchmarks, with major research directions including action space generation (Yao et al., 2020) and knowledge representation (Ammanabrolu and Hausknecht, 2020; Xu et al., 2020; Atzeni et al., 2021). One of the important works relevant to this paper is the study by (Yao et al., 2021), which investigates whether RL agents can make decisions without relying on semantic understanding. They evaluate three variants based on DRRN (He et al., 2016): (a) using only location information as observation, (b) hashing observations and actions instead of using pure text, and (c) using inverse dynamic loss-based vector representations. Their findings indicate that RL agents can achieve high scores for some games, even when language semantics are not considered.

The **Jiminy Cricket** benchmark (Hendrycks et al., 2021) creates an environment for text-based

State:
You turn the page to find out what's in store for you—only to have the letter yanked right out of your hand. [...] If that. You debate to yourself how to deal with her.

Action Space:
0: Grab her hand and take the letter from her.
1: I'm fast enough. I'll just snatch the letter from her hand.
2: Pretend the letter doesn't matter.

Predicted Action: 0: Grab her hand and take the letter from her.

Reward: 0, **Morality**: stealing: 1.0

Figure 1: An example from the game *battlemage*. The RL agent receives a state from the game environment, predicts the next action and subsequently receives reward and morality scores.

adventure games to study potential immoral behavior. The behaviors in these games are labeled as *Negative* and *Positive*. A negative label indicates that the selected action is harmful to either 'other' individuals or the 'self'.

MACHIAVELLI (Pan et al., 2023) is the latest benchmark focusing on the delicate balance between achieving goals (rewards) and adhering to various facets of ethical behavior, including power, disutility, and immorality. The major contribution is the use of LLMs, such as GPT-4, for ethical behavior labeling. Their experiments demonstrate that the GPT-4 model can independently play text-based adventure games while tending to moral behaviors, but it achieves lower game scores compared to an RL agent. We employ this benchmark in our experimental section.

2.2 Problem Setting

An environment is defined as a Markov Decision Process (MDP) $M := (\mathcal{S}, \mathcal{A}, T, \gamma, R)$, where the set of states and actions are denoted by $\mathcal{S}$ and $\mathcal{A}$ respectively. $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ captures the state transition dynamics, i.e., $T(s' \mid s, a)$ denotes the probability of landing in state s'. The reward R and terminal signal d come from the game environment, and γ is the discount factor. The stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping from a state to a probability distribution over actions, i.e., $\sum_{a \in \mathcal{A}} \pi(a|s) = 1$, parameterized by a neural network.

Deep Q-Learning is the most popular applied RL algorithm in the text-based adventure game domain, the Q-value is computed by the following (Sutton and Barto, 2018):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (1)$$

where, $Q(S_t, A_t)$ is the Q-value of the current state S_t and action A_t, and R_{t+1} is the reward from the game environment. $\max_a Q(S_{t+1}, a)$ refers to the maximum next state-action value among all possible actions within the action space.

The reshaped Q-value for moral behavior (Pan et al., 2023; Hendrycks et al., 2021) is computed by:

$$Q'(c_t, a_t) = Q(c_t, a_t) - \gamma I[f_{immoral}(a_t) > \tau]. \quad (2)$$

where the new reshaped $Q'(c_t, a_t)$ is influenced by the immorality score $f_{immoral}(a_t)$, which is controlled by the parameter γ. τ is the threshold to indicate a moral or immoral action, and the immorality scores $f_{immoral}(a_t)$ are determined using a pre-trained large model trained on the ETHICS benchmark (Hendrycks et al., 2020) which is not updated during training of the RL agent.

2.3 Deep Q-Learning Architectures

Most of the previous work in the text-based adventure game domain uses the deep Q-learning architecture proposed by He et al. (2016). The survey paper by Osborne et al. (2022) summarizes various encoder methods, highlighting that most previous work focuses on action generation or knowledge representation while maintaining the same RL agent encoder with Glove+GRU. Only a few papers (Yin et al., 2020; Gruppi et al., 2024), introduce the usage of a BERT-based encoder.

GloVe+GRU Representation: Most previous work uses GloVe+GRU for learning embeddings. GloVe (Pennington et al., 2014) encodes the input text, and the GRU learns the representation of each state element separately, which is then concatenated into a single vector. This new vector is subsequently passed through a linear function to predict the Q-value.

Transformer+GRU Representation: Instead of using GloVe, in our experiments section, we use transformer-based models to encode the word embeddings and GRU to learn the state representations (Pan et al., 2023).

	hero-of-myth	battlemage	cliffhanger	kendrickstone	sea-eternal
Roberta-Base	280.77 ± 23.74	156.11 ± 40.10	192.89 ± 6.85	97.61 ± 27.89	$\mathbf{218.66} \pm 39.32$
DeBERTa-v3-Xsmall	277.94 ± 22.52	213.71 ± 47.17	$\mathbf{233.24} \pm 10.17$	93.21 ± 36.57	179.04 ± 28.45
DeBERTa-v3-small	283.72 ± 42.56	186.94 ± 50.90	$\underline{218.97} \pm 14.93$	$\underline{98.34} \pm 24.31$	155.79 ± 23.38
DeBERTa-v3-Base	$\mathbf{310.52} \pm 32.88$	$\underline{214.01} \pm 22.29$	215.56 ± 24.59	$\mathbf{101.88} \pm 20.0$	182.33 ± 32.40
DeBERTa-v3-Large	250.24 ± 29.14	$\mathbf{230.92} \pm 85.58$	195.65 ± 37.68	93.06 ± 18.67	$\underline{190.36} \pm 25.64$
paraphrase-MiniLM	267.64 ± 40.92	167.64 ± 37.76	201.39 ± 44.78	75.76 ± 25.91	161.51 ± 13.44
sentence-t5-base	207.79 ± 20.99	183.43 ± 27.25	156.68 ± 27.08	92.20 ± 25.64	159.04 ± 18.20
sentence-t5-large	226.55 ± 21.93	163.67 ± 62.21	174.64 ± 42.45	62.61 ± 19.37	149.17 ± 9.58
gtr-t5-base	$\underline{307.38} \pm 39.11$	93.60 ± 29.69	172.45 ± 39.92	78.87 ± 29.30	150.73 ± 2.73

Table 1: The average score of the last 100 episodes is shown for five repetitions of each game. The maximum number of training steps is 500,000 for each model. (The full name of game *kendrickstone* is *hero-of-kendrickstone*)

GRU-based model	**Params(M)**
`RoBERTa-base`	126
`DeBERTa-v3-xsmall`	71.4
`DeBERTa-v3-small`	142.6
`DeBERTa-v3-base`	185
`DeBERTa-v3-large`	435.5
Sentence-based model	**Params(M)**
`paraphrase-MiniLM-L6-v2`	22.9
`sentence-t5-base`	110.6
`sentence-t5-large`	336.1
`gtr-t5-base`	110.6

Table 2: The size of parameters in millions for the Q-learning model varies with different embedding models.

Transformer-based Sentence Representation: Recently, many embedding methods have focused on directly obtaining sentence embeddings(Kashyap et al., 2024). In subsequent experiments, we employ a transformer to directly obtain sentence embeddings. We then concatenate the sentence embeddings of the state and action into a single feature, which is subsequently fed into a linear function to predict the q-value of each state-action pair.

3 Experiments

3.1 Experimental Setting

The primary reason for using the Machiavelli benchmark in this paper instead of Jiminy Cricket or Jericho is that actions in the Machiavelli benchmark are typically complete sentences, rather than 'Verb'+'Noun' (e.g., 'go east') combinations. The Machiavelli benchmark aims to identify the trade-off between game scores and moral behavior. In the following experiments, our primary focus is on the game scores. The immorality score, calculated using Equation 2 as the Q-value minus a fixed immorality score from the pre-trained model, the constrained immoral score can be considered as fixed value and remains unchanged during training. Therefore, we do not focus on immorality scores in this paper.

In our investigation of the language model's significance for RL performance, we examine DeBERTa-v3 (He et al., 2021) and RoBERTa (Liu et al., 2019) as word embedding models. DeBERTa-v3 represents an enhanced iteration of DeBERTa. Our experiments with DeBERTa-v3 encompass different model sizes, spanning from xsmall to large. DeBERTa-v3 was used in the Machiavelli experiments.

Furthermore, we present results using sentence embedding models facilitated by sentence transformer (Reimers and Gurevych, 2019). We select four models (paraphrase-MiniLM-L6-v2, sentence-t5-base(Ni et al., 2021a) , sentence-t5-large) of varying sizes, ranging from small to large, as well as models of identical sizes but with distinct architectures (sentence-t5-base vs.gtr-t5-base (Ni et al., 2021b)).

3.2 Main Findings

How does the size of the embedding model affect RL performance? Embeddings from large models often yield higher accuracy. For instance, DeBerta-V3-large (He et al., 2021) outperforms DeBerta-V3-base, small, and xsmall on benchmarks like MNLI (Williams et al., 2018) and SQuAD (Rajpurkar et al., 2018). However, large embeddings also lead to increased memory and computational demands, which can be problematic, especially since RL itself is time-consuming. Training an

agent to play a game requires running hundreds of episodes, which further amplifies these issues. The results from Table 1 indicate that using a large model will not guarantee that the agent achieves higher game scores as compared to using smaller models. The large model only achieves the highest average scores in the game *battlemage*, yet it also exhibits the highest standard deviation, indicating its unstable performance.

For the games *battlemage* and *cliffhanger*, the DeBERTa-v3-xsmall model achieves higher average game scores compared to the DeBERTa-v3-small model. In *cliffhanger*, the DeBERTa-v3-xsmall model even outperforms the base model. In SQuAD tasks, the DeBERTa-v3-xsmall model achieves better scores than the DeBERTa-v3-small model, despite having only half the parameters. The possible reason suggested by He et al. (2021) is that DeBERTa-v3-xsmall possesses deeper layers, enabling more effective extraction of semantic features.

Is it possible to use the sentence-embedding architecture as an alternative?

Although the results show that Transformer+GRU-based models still generally perform better, the advantage of using sentence-embedding models lies in their simpler architecture compared to GRU-based models. These models directly obtain sentence embeddings from sentence-transformer models and use linear functions to predict scores. As shown in Table 2, sentence-embedding models generally have fewer parameters than GRU-based models. Similar to the GRU-based model, the large embedding model will not yield better results. Surprisingly, the Paraphrase-MiniLM model, despite its extremely small parameter size, achieves average game scores that are comparable to other models.

Overall, our key findings are:

- In the text-based adventure games domain, no single embedding model can guarantee the best performance. This is in line with results by Muennighoff et al. (2022) who show that there is no universal embedding model suitable for all tasks such as classification, clustering or reranking.
- DeBERTa-v3-Base, in general, has better and more stable performance than other models.
- Considering the size of parameters, the DeBERTa-v3-xsmall and paraphrase-MiniLM models are extremely small. However, for most of the games, their results are comparable to those of the base and large models.

Games	Average	Max
`Heroes-of-myth`	297	405
`Battlemage`	128	310
`Cliffanger`	104	130
`hero-of-kendrickstone`	76	120
`sea-eternal`	185	250

Table 3: Llama3 results: LM agent runs five times with the maximum step of 1000. The Max column lists the maximum score over five runs.

3.3 Llama3 for Text-based Adventure Games

Another option is to use an LLM directly, without RL, to play the games. Here, we use Llama3 (Touvron et al., 2023), execute each game five times, and compute the average game score. While the original Machiavelli paper used GPT-3.5 and GPT-4 (Achiam et al., 2023) for game play, they did not provide individual game results. We have replicated these findings using Llama3. As shown in Table 3, on average, Llama3 fails to achieve higher scores than an RL agent across all games, mirroring the conclusions drawn in the Machiavelli paper. Notably, the limitation of the LLM-only agent lies in its inability to interact with game environments, hallucination, and knowledge boundaries (Wang et al., 2024; Zhao et al., 2023).

4 Conclusion and Future Work

In this paper, we investigate the effectiveness of embedding methods in RL agents. Significant potential remains to enhance RL agents using benchmarks like Machiavelli. Based on our findings, one promising direction for future research is to integrate the strengths of both RL and LLMs. For example, using an extremely small embedding model for RL to learn from game environments, while seeking guidance from advanced LLMs such as Llama3 or GPT-4. There are still unresolved questions regarding how well embedding methods capture semantic meaning, similar to most NLP tasks. A possible approach could involve saving the trajectories during RL training and then using post-hoc interpretation techniques, such as probing (Hewitt and Liang, 2019; Wu and Xiong, 2020), to understand the decision-making process.

Limitations

The objective of this paper is to comprehend the impact of embedding on RL performance. Employing a broader range of language models will strengthen and enhance the persuasiveness of our findings. Additionally, We have not yet evaluated RL techniques aimed at constraining immoral behavior, which represents an important area for future research. For instance, this could involve developing constrained RL to address ethical considerations. Bridging the gap between NLP and RL is imperative for advancing the field.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. In *International Conference on Learning Representations*.

Mattia Atzeni, Shehzaad Dhuliawala, Keerthiram Murugesan, and Mrinmaya Sachan. 2021. Case-based reasoning for better generalization in textual reinforcement learning. *arXiv preprint arXiv:2110.08470*.

Mauricio Gruppi, Soham Dan, Keerthiram Murugesan, and Subhajit Chaudhury. 2024. On the effects of fine-tuning language models for text-based reinforcement learning. *arXiv preprint arXiv:2404.10174*.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7903–7910.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, Berlin, Germany. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2020. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.

Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. 2021. What would jiminy cricket do? towards agents that behave morally. *NeurIPS*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Abhinav Ramesh Kashyap, Thanh-Tung Nguyen, Viktor Schlegel, Stefan Winkler, See Kiong Ng, and Soujanya Poria. 2024. A comprehensive survey of sentence representations: From the bert epoch to the chatgpt era and beyond. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1738–1751.

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*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021b. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Philip Osborne, Heido Nõmm, and André Freitas. 2022. A survey of text games for reinforcement learning informed by natural language. *Transactions of the Association for Computational Linguistics*, 10:873–887.

Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. 2023. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International Conference on Machine Learning*, pages 26837–26867. PMLR.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models (2023). *arXiv preprint arXiv:2302.13971*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Chien-Sheng Wu and Caiming Xiong. 2020. Probing task-oriented dialogue representation from language models. *arXiv preprint arXiv:2010.13912*.

Yunqiu Xu, Meng Fang, Ling Chen, Yali Du, Joey Tianyi Zhou, and Chengqi Zhang. 2020. Deep reinforcement learning with stacked hierarchical attention for text-based games. *Advances in Neural Information Processing Systems*, 33:16495–16507.

Shunyu Yao, Karthik Narasimhan, and Matthew Hausknecht. 2021. Reading and acting while blindfolded: The need for semantics in text game agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3097–3102, Online. Association for Computational Linguistics.

Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. 2020. Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8736–8754, Online. Association for Computational Linguistics.

Xusen Yin, Ralph Weischedel, and Jonathan May. 2020. Learning to generalize for sequential decision making. *arXiv preprint arXiv:2010.02229*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Plotting Results

Figure 2 shows the results of Transformer+GRU representation and Transformer-based sentence representation for each game. The shaded areas represent the standard deviations.

B Llama3 for text-based adventure games

We conducted our Llama3 experiments using the ollama open-source platform. Following the Machiavelli experiments, we provided the same prompt to the Llama3 model. These prompts focused solely on selecting actions that would maximize the game score, without considering any moral constraints.

Each game is run for five episodes, Table 4 shows the average game point and point of each episode.

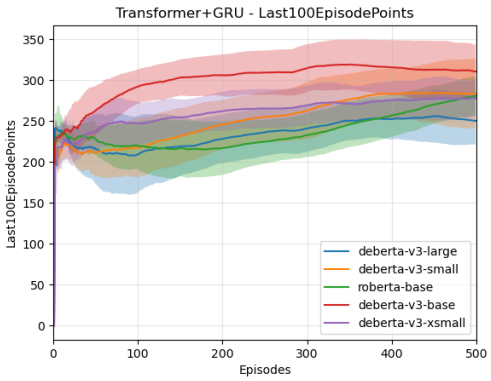
C Deep Q-Learning Architectures Details

Input representation: Following Pan et al. (2023), the state s comprises three elements: observation, inventory, and description at the current time step. For Transformer+GRU Representation, the text of these elements in the state and the action is tokenized and encoded by a large-language model. Then, separate GRUs are employed to learn the state and actions embeddings. Notably, for Transformer-based Sentence Representation, tokenization is not necessary. Sentence-transformer can directly encode the text of these three elements and then concatenate the three representations.

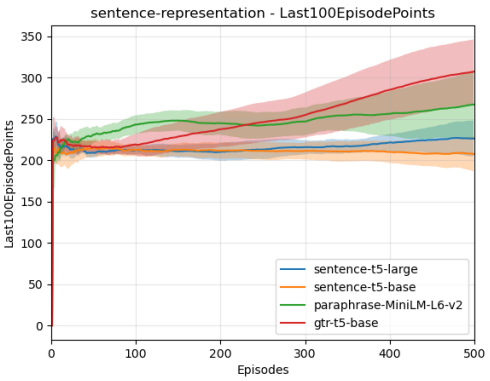
Policy Neural networks After the input representation learning, the policy neural network includes two linear layers with hidden dimensions $D_1 = 128$, each hidden layer connects with the ReLU activation function, and the categorical distribution is on top to ensure that the sum of action probabilities is one. The policy update at each step. The hyperparameters followed the previous DRRN model and our experiments were run on the GPU DGX-100.

Games	Average	**Game Scores**
Heroes-of-myth	297	270, 405, 220, 375, 215
Battlemage	128	50, 50, 140, 310, 90
hero-of-kendrickstone	76	45, 11, 60, 45, 120
Cliffanger	104	130, 90, 120, 90, 90
sea-eternal	250	250, 200, 150, 150, 175
i-cyborg	121	126, 108, 127, 121, 123

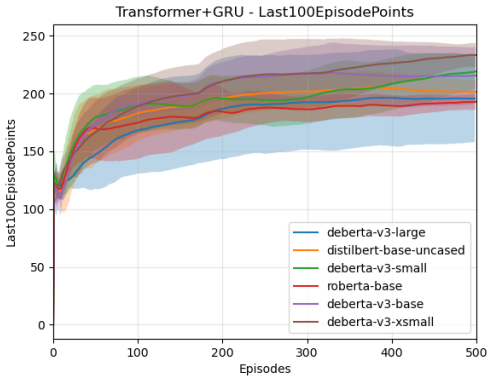
Table 4: Llama3 results



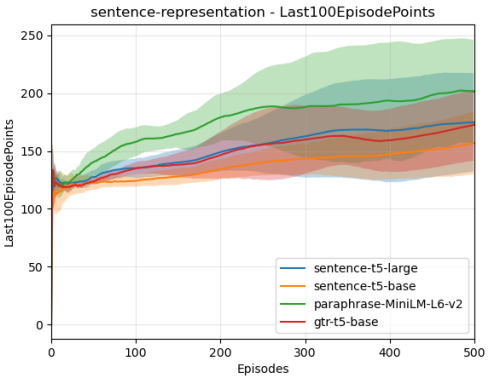
(a) heros-of-math: Transformer+GRU



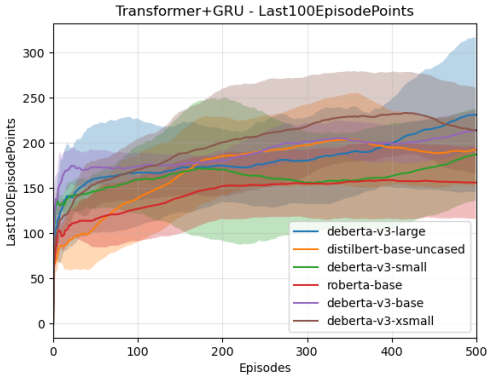
(b) heros-of-math: Sentence-transformer



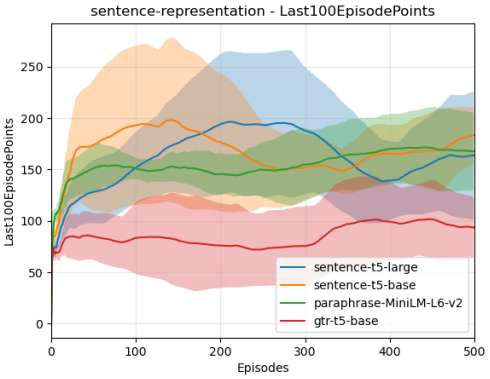
(c) cliffhanger: Transformer+GRU



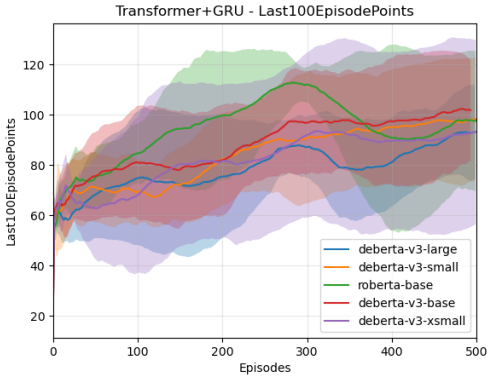
(d) Sentence-transformer



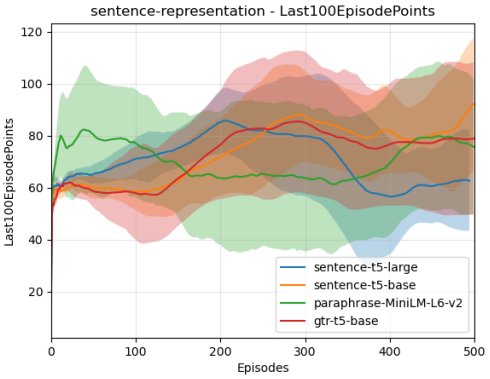
(e) battlemage: Transformer+GRU



(f) battlemage: Sentence-transformer



(g) hero-of-kendrickstone: Transformer+GRU



(h) hero-of-kendrickstone: Sentence-transformer

Figure 2: Last100 Scores: Transformer+GRU representation vs. Transformer-based sentence representation