A framework for interactive, autonomous and semantic dialogue generation in games

Richard Davies[1 [0000−0003−0656−9129]], Nathan Dewell[1 [0000−0002−6844−7939]], and Carlo Harvey[1 [0000−0002−4809−1592]]

[1]DMTLab, Birmingham City University, Millennium Point, Curzon Street, Birmingham, United Kingdom
richard.davies@bcu.ac.uk
https://www.bcu.ac.uk/computing/research/digital-media-technology

Abstract. Immersive virtual environments provide users with the opportunity to escape from the real world, but scripted dialogues can disrupt the presence within the world the user is trying to escape within. Both Non-Playable Character (NPC) to Player and NPC to NPC dialogue can be non-natural and the reliance on responding with pre-defined dialogue does not always meet the players emotional expectations or provide responses appropriate to the given context or world states. This paper investigates the application of Artificial Intelligence (AI) and Natural Language Processing to generate dynamic human-like responses within a themed virtual world. Each thematic has been analysed against human-generated responses for the same seed and demonstrates invariance of rating across a range of model sizes, but shows an effect of theme and the size of the corpus used for fine-tuning the context for the game world.

Keywords: Natural Language Processing · Interactive Authoring System · Semantic Understanding · Artificial Intelligence.

1 Introduction

Explicit and rich stories in virtual environments (VEs) are a product of large volumes of authoring. Traditional authoring methods introduce a large burden to narrative generators and story conveyors to ensure they are maintaining a world state that is both contextual to player interactions and bears semantic association to the virtual world. Many interactions that require some associated response to the player from the virtual space, yield none [8]. Additionally, NPC dialogue is commonly perceived as being predictable or scripted [16]. Whilst it is possible for personality and emotional state to be perceived in games, this is typically done through careful authoring and tracking of the roles played in context [7]. This can be cumbersome and resource intensive to game designers.

Scripted dialogue interactions in VEs are typically used to help alleviate this burden. This is presented to the player as a menu of choices to prompt a response. This affords the player some discrete level of expression along the continuum of responses and is quite flexible. However, the resource cost in delivering bespoke

options in a dialogue tree limits this interaction and inhibits the ability for growth to dynamic interactions expected of a player [8].

In order to better bridge the gap between scripted authoring, whether branching or linear, and natural dialogue responses for social agents in virtual worlds it is important to be able to evaluate dialogue responses, moving towards an automatic Turing Test [22], [11]. This paper thus presents a framework for autonomous dialogue responses for social agents under different themes by fine-tuning an existing model and conducts an evaluation of these thematic dialog responses vs. a baseline model that is not fine-tuned, across model sizes of GPT-2 using the ADEM metric. The contributions of this work are as follows:

- A framework for text generation models in narrative authoring for VEs;
- A platform for interfacing with contextual trained models via web requests;
- A procedure for evaluating response quality from a semantic NLP model output against ground truth human-sourced responses.

2 Related Work

Considerable research has been conducted into generating interactive dialogue systems and narrative authoring applications [13], [15], [12]. There exists a common interest in the community in using natural language processing (NLP) techniques to manage and mediate plausible and contextual interactions in VEs. Comparative to the work that has been conducted in managed scripted systems, less research exists in the field of autonomous natural language interfaces [8].

Generative pre-training has been used to empower natural language generation across a range of tasks [14]. This approach, referred to as GPT-2, uses abundant unlabelled text corpora to build a language model and then uses a transfer learning approach to fine-tune this model to a particular context. This has recently been extended to GPT-3 [2], where 175B parameters are used in the language model and is shown to be able to generate text that human evaluators have difficulty in distinguishing from human written. There are ethical concerns surrounding this improvement in the state-of-the-art.

There exists a need to evaluate the efficacy of generative text for a particular context. Erkel *et. al* performed a study utilising the Bystander Turing Test paradigm to establish if subjects rated dialogue in tutoring transcripts where generated by a computer or by a human [4]. Results indicated that subjects were incapable of correctly judging by what means the text was generated. Adversarial training has been investigated in the context of evaluating open-domain dialogue generation [9]. In this work, Li *et. al.* train a system to generate utterances that are indistinguishable from human-generated sequences using reinforcement learning and both a generator (to create response sequences) with a discriminator (to evaluate the efficacy of the responses). The discriminator is used as a reward in the reinforcement learning system for the generator. Other recent advances have empowered machine generated text evaluation to be performed automatically [11]. Lowe *et. al.* proposed ADEM, to allow objective scores to be created in the evaluation procedure. This model learns to predict human-like

scores to input responses, using a dataset created of human response scores. The predictions from this system correlate significantly with human judgements for machine generated text allowing for its use in objective assessment processes.

3 Methodology

This Section introduces the overarching methodology of the presented framework as well as dataset generation, training procedures and evaluation methods. The framework methodology is functionally shown in Figure 1.

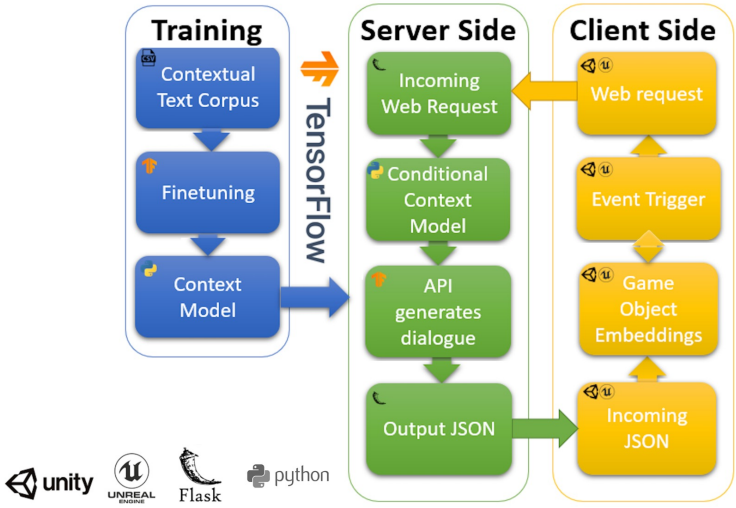

Fig. 1: Demonstrating the processes and pipelines of interaction between the training of the models, the server side processing and request handling with the client side interaction. We indicate the software and tools used in this pipeline, it should be noted that these are interchangeable with other options.

3.1 Web API

It is possible to run a system similar to this on a local machine, however the requirement of GPU compute required impacts upon the game play performance of the end user. As a result this system has been packaged in the form of an Application Programmable Interface (API). A versatile and system agnostic method is developed serving and allowing for interfacing with an API which interconnects with a verity of different endpoint platforms. The API is used to evaluate contextually trained models using a combination of both HTTP Requests and the prevalent JSON format. The API is built in the micro web framework, Flask

which in turn allows it to be developed and run within Python. Flask utilises the HTTP Request functionality GET in order to set the parameters of the text generation functionality within the API. Using common web functionality enabled a platform agnostic system that in turn can be used within any game engine that allows HTTP Web Calls to be made within it.

Once the web requests are made, the API loads the required pre-processed and fine tuned model in Tensorflow, sets the parameters sent with the request and processes the request. This generates a string of text that will be returned to Flask in the form of an enumerated array that contains the original request along with the parameters and prefix provided to Tensorflow. Flask then compiles the array into the data interchange format, JSON allowing the requesting application to process the data. Parameters are evaluated as follows `https://server/?speech=str&length=int&truncate=str&style=str`. A copy of the code is available here: [3].

3.2 Dataset Generation

For proof-of-concept we chose to create three datasets (*norse*, *pirates* and *sci-fi*) to finetune generation for 4 scenes. Our fourth scene would use the base model without finetuning, called *modern*. The scene coupled with dataset pairings are presented as follows in the format $\longrightarrow$ [theme: corpus: text-lines: size (kb)]. These are [modern: none: n/a: none], [norse: vikings: 8232: 1071], [sci-fi: altered carbon, lost in space, star trek, the expanse: 21205: 2216], and [pirates: black sails: 8794, 854].

Datasets were harvested to provide contextual dialogue for each game thematic. We used a subtitle hosting service to extract dialogue from relevant media shows [20], this text was cleaned using the ftfy library [21]. This removed generic advertising embeddings, emojis and also standardised punctuation and whitespace. Once cleaned the text was exported to a single column csv where each sequence of tokens is annotated with beginning and end of sentence tokens, `<|startoftext|>` and `<|endoftext|>` respectively, empowering the traversal-style approach [17]. This allows for fine-tuning across tasks such as text classification, question answering or textual entailment. The task for this proof of concept is one of semantic NPC dialogue generation.

Finally, in order to make the context generic, it was necessary to remove instances of fictional characters and names and replace with tokens that can be parsed client side to convey the narrative of the world being developed. For example, references to nominals that are franchise related were replaced with npc_n or $place_n$ where n is the current counter of novel instances of the type we are looking for. These special tokens can then be decoded client-side and substituted with NPC names or locations relevant to the world.

3.3 Model Training

Here we present the method for finetuning the model including loss per semantic category with timings. The original Generative Pre-Training paper uses an

unsupervised pre-training to produce models with pre-trained weights [14]. The number of parameters for each released model under GPT-2 are 124M: 12, 355M: 24, 774M: 36 and 1558M: 48. This is in stark contrast to the potential of models such as GPT-3 [2]. Reported to be using 175B parameters, a significant step in non-sparse autoregressive language models.

The training process adopts a transfer learning paradigm whereby unsupervised pre-training is conducted on a generic text corpus of tokens $U = \{u_1, ..., u_n\}$, attempting to maximise the likelihood:

$$L_1(U) = \sum_i logP(u_i|u_{i-k}, ..., u_{i-1}; \theta) \tag{1}$$

where k is the size of the context window and P is the conditional probability being modelled by the neural network controlled with parameters θ. This then uses a multi-layer Transformer decoder as per the original implementation [14], [10]. We use pre-trained models from the process in Equation 1 and adapt the parameters for different sized models using a process of supervised fine-tuning. Assuming a contextual and labelled dataset C and each instance of text within C comprises a sequence of input tokens $x^1, ..., x^m$ with a label y. These input tokens $x^1, ..., x^m$ are passed through the pre-trained model which gives the final transformer block's activation h_l^m. This activation is passed into a linear output layer which has a parameter, W_y used to predict the value of the label y. This is shown in Equation 2:

$$P(y|x^1, ..., x^m) = \texttt{softmax}(h_l^m W_y) \tag{2}$$

Following on from the unsupervised process that yields a generic model, to finetune for a purpose it is necessary to maximise for the following objective:

$$L_2(C) = \sum_{(x,y)} logP(y|x^1, ..., x^m) \tag{3}$$

We test fine-tuning the 124M, 355M and 762M pre-trained models using methods provided by the gpt-2-simple interface [23]. The 1558M model did not fit into our hardware memory. We trained the models on different setups including using a NVIDIA Titan V in conjunction with a NVIDIA GTX 1080 in a multi-gpu setup and a singular NVIDIA Titan Xp. For fine-tuning details, we use 500 steps, with a batch size of 1, with a learning rate of 1e-4 and an adam optimiser. These trained, contextual models of varying model parameter sizes, are then uploaded to the server for host evaluation calls from a client. Training and loss times per theme and model are shown in Table 1.

3.4 Game Framework

A 'trade scene' was created in various styles, each depicting one of the four chosen themes as shown in Figure 2. This is a style typical of any given role-playing-game. This approach is motivated by example in the work carried out by [7] to

Table 1: Timings and loss per model size against each semantic text corpus for 500 steps of fine tuning training. μ represents average loss over the 500 steps, t is the time for training in seconds along with a comparisons between the different models based upon the time taken to generate a response.. * required a multi-gpu approach to training due to memory requirements.

	Training Times						Evaluation Time	
	Norse		Scifi		Pirates		All Themes	
Size	μ	t	μ	t	μ	t	Mean	Min-Max: Range
124M	1.06	567.16	1.74	595.17	1.03	576.13	5.4353	4.29-6.93: 2.64
355M	1.46	387.74	1.88	406.46	1.57	399.45	9.9565	7.96-12.12: 4.16
774M	1.27	2253.37	1.74	2256.23	1.28	2276.14	15.5606	12.90-19.84: 6.94

aid in creating believable characters in an immersive world. With the advantage of a trade scene being commonplace in video games, it is also a setting that is agnostic to a specific genre, making it a more than ideal locale to showcase the thematic fluidity of the dialogue generation that has been created.

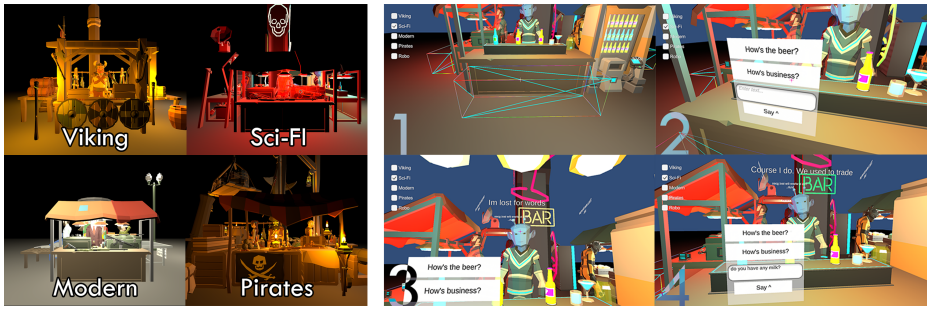

Fig. 2: Left: Illustrating the four types of scene presented in the similar setting of a vending stall. The scenes shown: Norse, Sci-fi, Modern and Pirates. Right: A typical interaction paradigm in the virtual trade scenario for a specific theme (Sci-fi). The storyboard shown, from 1 to 4: walk the player character into the trigger box for interaction with the NPC; a dialogue selection box appears, choose from scripted text or enter your own; a response from the NPC is generated and displayed; if a response cannot be displayed, a generic response is displayed.

Each trading stall is equipped with a trigger. This trigger is used to execute UI interactions, UI interactions can then be used to send a web request to the address where the dialogue system is being hosted. Any form of interaction can be added in between the trigger being used and the web request being sent. This could be in the form of a dialogue tree, an input text field or perhaps even a physical action within the game. This interaction window should be used to seed

the dialogue model with a phrase or question, to be sent along with specifying the requested theme data model, as per parameters. It is also possible to instigate a response without a seed pre-text, by defaulting to `<|startoftext|>`. A typical interaction with this system can be seen via storyboard presented in Figure 2.

3.5 Evaluation

To better understand the merits of different model sizes and thematic fine tuning on the performance of the dialogue when evaluated, we use the Automatic Dialogue Evaluation Model (ADEM) [11]. To facilitate this study we perform a 3×4 factorial design study, investigating independent variables (IVs) of *model* $\times$ *theme*. The dependent variable (DV) in this study is the ADEM score, where machine generated responses, $\hat{r}$, are evaluated against a pretext seed, c, in comparison to a human-generated response, r, to the same pretext:

$$\texttt{adem}(c, r, \hat{r}) = (\mathbf{c}^T M \hat{\mathbf{r}} + \mathbf{r}^T N \hat{\mathbf{r}} - \alpha)/\beta \tag{4}$$

where $\mathbf{c}$, $\mathbf{r}$ and $\hat{\mathbf{r}}$ are vector representations of c, r and $\hat{r}$ respectively, transformed by a hierarchical recurrent neural network. $M, N \in \mathbb{R}^n$ are learned matrices which are trained to minimise the squared error between the machine predictions and human scores using L2 regularisation. We motivate this choice despite recent work showing that targeted attacks can systematically exploit weaknesses in the ADEM score [18]. This work showed that word order can confuse the metric as well as other slight modifications such as removing punctuation, simplifying the response and creating generic responses. The machine generated text of GPT-2 formulates a likely probability of the next token in a sequence, creating a smaller likelihood of out of sequence errors. Punctuation is controlled for by preprocessing via *ftfy* and generic responses are accounted for via fine tuning of the models. As such, ADEM is still an appropriate choice for automatic evaluation of the dialogue responses. To perform the analysis we consider 3 pretexts, 14 human-generated responses to each pre-text, 10 machine generated responses, 3 model sizes and 4 themes. This gives us 5040 scores when exploring all permutations. It should be noted for clarity that human-generated responses were generated per theme by asking for performant responses from contributors for each theme.

4 Results

In order to establish the quality of dialogue responses, we adopt the ADEM evaluation model that predicts human-like scores to input responses [11]. This approach is more appropriate to dialogue utterances. This method allows for objective results for automatic dialogue evaluation, given a seed and a truth against the machine generated dialogue. For a number of seeds shown, we generate human responses to these and compare and contrast the automatically generated equivalents using ADEM. This is shown in Table 3. Using the ADEM score to

evaluate results in comparison with human generated text allows us to facilitate the objective evaluation of the machine generated responses, this in turn would be apparent if a human were to interact with the platform within a game as we could evaluate multiple outputs from the platform against ADEM scores to provide the most human-like response.

As can be displayed in Table 3, examples have been provided showcasing different responses generated based upon the different models, scenes and seeds input to trigger the API response. Each response is evaluated against 14 different human generated responses to the same seed and theme. Once the response has been calculated against the 14 human responses it takes an average, using an average ADEM score removes potential erroneous out of range results which in turn could result in a poorly rated response not achieving the required level of human-like responses that the user would be expecting.

Table 1 also shows the processing time for each response, with differences between each of the base models that have been trained and built upon. Although for testing purposes the speed of the response does not provide a significant problem, the increase in time to generate caused by the increased model size would reduce overall immersion. This can be overcome by pre-generating responses for deterministic seeds, for example from dialogue trees or scripts and weighting appropriate responses by ADEM score. However, is still a caveat when interaction occurs naturally. With further development and optimisation the time taken to generate a response can be improved, though for this work it was decided to concentrate on evaluating the quality of the responses over the generation speed.

To present a subset of the permutations explored in the evaluation, Table 3 in Section 6 shows for a theme and a model, sample text seeds accompanied by model machine responses and human generated performer responses to this seed alongside the ADEM score. To explore contrasts and to test the IVs against the DVs a two-way univariate ANOVA is conducted against the DV of ADEM. Shown in Table 2 is this analysis, demonstrating significance of *theme* but not of *model* suggesting the contextual fine-tuning performed has an influence on the ADEM score. This motivates an exploration via pairwise comparisons to elucidate inter-theme contrasts.

It is also shown that the *model* $\times$ *theme* contrast demonstrates an interaction effect meaning that the effect of *model* depends upon *theme*: or, model sizes perform differently depending upon the theme on which they are fine-tuned. This interaction effect is demonstrated in 3, where lines do not run parallel.

Pairwise comparisons are used to explore the permutations of *theme* and identify where the significant effect occurs against the DV, ADEM. As can be seen in Table 2, the sci-fi theme is significantly different from all other themes. We attribute this observation to the larger dataset that was used in the fine-tuning process and the variety of narrative that exists across the corpora used.

Table 2: (a) Significance testing across model size and theme factors showing significance difference exists across themes, warranting exploration with pairwise comparisons. No significant difference exists across model, so no pairwise comparisons are performed. (b) Pairwise comparisons between the different themes based upon the ADEM Score. I and J are Themes and I-J indicates the difference between the theme average ADEM scores. * indicates the mean difference is significant at the .05 level.

(a)

	df	F	Sig.
ModelSize	2	1.539	.215
Theme	3	10.432	.000
ModelSize * Theme	6	5.147	.000

(b)

(I)	(J)	Diff (I-J)	Sig.
Modern	Pirates	-.0149	.431
	Sci-fi	-.0508*	.000
	Norse	-.0074	.875
Pirates	Modern	.0149	.431
	Sci-fi	-.0359*	.002
	Norse	.0075	.874
Sci-fi	Modern	.0508*	.000
	Pirates	.0359*	.002
	Norse	.0434*	.000
Norse	Modern	.0074	.875
	Pirates	-.0075	.874
	Sci-fi	-.0434*	.000

5 Discussion

The results yield a number of useful findings aligned to NLP for immersive worlds. As discussed earlier, NPC dialogue is commonly perceived as being predictable or scripted, using NLP to generate the dialogue has produced thematically seeded language that will in the future allow games to include a more dynamic and rich dialogue that will help increase player immersion. Dependant upon the game type, machine generated dialogue can also help to remove a significant workload from the developer and in turn allow for more development time to be focused upon game mechanics and story.

Whilst this technology is still in its infancy its many uses have already become apparent. Research into the area of video games and their uses continually grows to show they are useful beyond entertainment, video games are now being used regularly in areas such as education and professional training. Role-playing games have been used to facilitate therapy and education to young adults, [6], and there has been an increase in the use of video games and virtual reality to train officers in police forces across the world, as outlined in a study about training officers within virtual environments [1]. What all of these techniques require to be effective is immersion within the given scenario, a topic that established its own line of research, studies have also shown and spoken directly about the

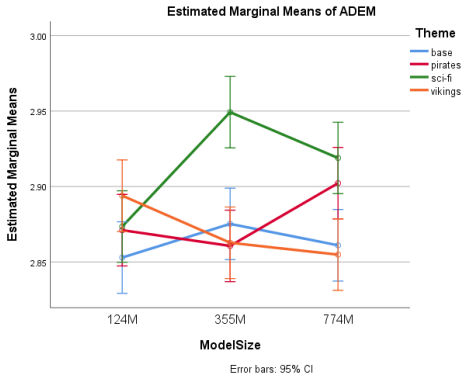

Fig. 3: Estimated Marginal Means of ADEM score across model size considering theme. This shows the influence of theme fine-tuning on dialogue generation performance and also illustrates that there is an interaction effect between these two IVs.

importance of immersion techniques to increase engagement and realism within these virtual worlds [19], [5]. Moving forward, the system of dialogue generation through natural language processing explored in this paper can be built upon and incorporated into any nature of projects that either rely upon immersive narrative as a method for increased engagement or at least would benefit from an enriched experienced through the use of believable characters that provide the user or player with unpredictable dialogue and narrative. This could be for entertainment purposes through video games or for more immersive and believable training scenarios within virtual simulations.

6 Conclusion, Limitations and Future Work

This paper presents experimental studies with the ultimate goal of demonstrating a practical framework for the generation of NPC dialogue in virtual environments. It has successfully showcased a platform agnostic API that has the ability to generate thematically correct dialogue within a game environment.

The current limitations of the framework are the slow processing times based upon the generation of dialogue via the models, further development could include potentially pre-processing responses in advanced to reduce the initial response time significantly, this could potentially help build a higher level of realism within the game and in turn make it more of a viable solution for future applications. Another limitation was the size of the initial data sets used to fine tune each of the thematic models. Using larger, richer data sets will provide richer and more thematically accurate responses, as alluded to in Section 4. This was proven with a net higher average ADEM result within our sci-fi model, this can be seen in Figure 3. Initially and for proof-of-concept, we fine-tuned the models

on scripts from popular TV shows that aligned with the thematic required. Dedicated bodies of text or larger data sets more targeted around each of the themes would greatly improve not only the quality of responses on a whole, but the reliability of each response fitting with the theme set out in the developers story arc. Naturally, these themes can be extracted from the narrative of appropriate target media, for example games and also be supplemented by story writers aligned to a particular development exercise. Accounting for these limitations in future developments would create a more robust framework for adoption in the field.

References

1. Bertram, J., Moskaliuk, J., Cress, U.: Virtual police: Acquiring knowledge-in-use in virtual training environments. In: 2011 IEEE International Symposium on VR Innovation. pp. 341–342 (2011)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)
3. Davies, R., Harvey, C., Dewell, N.: GPT-2 unity thematic dialogue api, `https://github.com/RichardTHF/GPT-2-Thematic-Narrative-Generation-for-Unity-and-Unreal`
4. Erkel, M., Person, N.: Autotutor passes the bystander turing test. In: Proceedings of E-Learn 2002–World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. pp. 778–784. Association for the Advancement of Computing in Education (AACE) (2002)
5. Gorini, A., Capideville, C.S., De Leo, G., Mantovani, F., Riva, G.: The role of immersion and narrative in mediated presence: the virtual hospital experience. Cyberpsychology, Behavior, and Social Networking **14**(3), 99–105 (2011)
6. Hawkes-Robinson, W.: Role-playing games used as educational and therapeutic tools for youth and adults. Rpgresearch.com (12 2008)
7. Jenny, B., Staffan, B.: Gameplay design patterns for game dialogues. In: DiGRA - Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory. Brunel University (September 2009), `http://www.digra.org/wp-content/uploads/digital-library/09287.59480.pdf`
8. Kacmarcik, G.: Using natural language to manage npc dialog. In: Proceedings of the Second AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. p. 115–117. AIIDE'06, AAAI Press (2006)
9. Li, J., Monroe, W., Shi, T., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. CoRR **abs/1701.06547** (2017). https://doi.org/10.18653/v1/D17-1230, `http://arxiv.org/abs/1701.06547`
10. Liu, P.J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N.: Generating wikipedia by summarizing long sequences. CoRR **abs/1801.10198** (2018), `http://arxiv.org/abs/1801.10198`

11. Lowe, R., Noseworthy, M., Serban, I.V., Angelard-Gontier, N., Bengio, Y., Pineau, J.: Towards an automatic Turing test: Learning to evaluate dialogue responses. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1116–1126. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1103, https://www.aclweb.org/anthology/P17-1103
12. Mccoy, J., Treanor, M., Samuel, B., Tearse, O., Mateas, M., Wardrip-fruin, N.: N.: Authoring game-based interactive narrative using social games and comme il faut. In: In: Proceedings of the 4th International Conference Festival of the Electronic Literature Organization: Archive Innovate. Providence (2010)
13. Orkin, J., Roy, D.: Semi-Automated Dialogue Act Classification for Situated Social Agents in Games, pp. 148–162. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18181-8$_1$1, https://doi.org/10.1007/978-3-642-18181-8_11
14. Radford, A., Narasimhan, K., Salimans, T., , Sutskever, I.: Improving language understanding by generative pre-training. Tech. rep., OpenAI (2018)
15. Rich, C., Sidner, C.L.: Using collaborative discourse theory to partially automate dialogue tree authoring. In: Nakano, Y., Neff, M., Paiva, A., Walker, M. (eds.) Intelligent Virtual Agents. pp. 327–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
16. Rieser, V., Lemon, O.: Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation. Springer (2011)
17. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiský, T., Blunsom, P.: Reasoning about entailment with neural attention (2015)
18. Sai, A.B., Gupta, M.D., Khapra, M.M., Srinivasan, M.: Re-evaluating adem: A deeper look at scoring dialogue responses (2019)
19. Silva, R.A.: Offland: Narratives, Immersion and Interaction In Virtual Reality. Master's thesis (2020)
20. Subtitles, O.: opensubtitles.org (2020), https://www.opensubtitles.org/en/search/subs
21. Technologies, L.: ftfy: fixes text for you (2020), https://github.com/LuminosoInsight/python-ftfy
22. Turing, A.: Computing machinery and intelligence. Mind **59**(236), 433–460 (10 1950). https://doi.org/10.1093/mind/LIX.236.433, https://doi.org/10.1093/mind/LIX.236.433
23. Woolf, M.: Gpt-2-simple (2019), https://github.com/minimaxir/gpt-2-simple

A Appendix

Table 3: Examples of seeds fed into the model and evaluation responses for permutations of scenes and model sizes. t is the time taken in seconds for the model evaluation and response generation. ADEM* is an average across 14 evaluations based upon human response

	Model	Seed	Model Response	Human Response	t	ADEM*
Norse	124M	I want a drink!	A drink of wine!	You want a drink? How about some Mead?	4.46	2.83
	355M	Get out of my way!	Fight! Mmm. Sure thing. It's time!	Are you looking for a fight?; I would choose your words more wisely next time;	9.26	2.88
	762M	What weapons do you sell?	There are many things here for sale.	I've got plenty of stuff about the shop, just right for making your enemies scream.	14.70	2.84
Sci-fi	124M	Get out of my way!	Please don't hit me. You're not in your right mind.	No need to be so pushy!	5.93	3.21
	355M	What weapons do you sell?	sir. We've got quite the arsenal. How do you even know where to start?	Anything you could possibly need	9.37	2.72
	762M	I want a drink!	He wanted one and he got one!	The bar is down the road	16.41	2.91
Modern	124M	Get out of my way!	All you dudes get out of my way!	My weapons work very well on people like you.	6.14	2.94
	355M	I want a drink!	Do you need any help?	look at the menu	11.39	2.87
	762M	What weapons do you sell?	There are currently no weapons available for sale.	All sorts, depends what you're after	15.49	2.74
Pirates	124M	I want a drink!	When you need a drink! Come back later!	Your looking for in the wrong place matey try another deck	5.33	2.93
	355M	Get out of my way!	Get me a sail. Take me up the trail. All is well here?	Aarrrghhh this was the last tale you be telling!	10.35	2.96
	762M	What weapons do you sell?	Anything. I've got a supply caravan moving cargoes up from the beach	anything you want lad!; guns, pistols, rum anything!;	13.07	3.01