

Natural Language Processing in Role Playing Games

by

William Ye
Z5061340

School of Computer Science and Engineering

The University of New South Wales

Bachelor of Engineering (Computer Engineering)

Submitted on
25, November 2020

Supervisor
Lorenzo Vigentini

Assessor
Aleksandar Ignjatovic

List of Figures

Figure 1 - Dialogue Interface in The Elder Scrolls: Skyrim	7
Figure 2 - Analysis of Relationship Score with Variability = 0.2	18
Figure 3 - Histogram and Normal Distribution of Relationship Scores, Post Task Completion with Variability = 0.2	19
Figure 4 - Analysis of Relationship Score with Variability = 0.1	19
Figure 5 - Histogram and Normal Distribution of Relationship Scores, Post Task Completion with Variability = 0.1	20
Figure 6 - System Design of Conversational Agent	21
Figure 7 - Example Conversation from model fine-tuned on Rick and Morty corpus	22
Figure 8 - Comparison of LIWC Categories between training data and generated data	24

List of Abbreviations

(NPC)	Non-Playable Character
(AI)	Artificial Intelligence
(RPG)	Role Playing Game
(ML)	Machine Learning
(NLP)	Natural Language Processing
(RNN)	Recurrent Neural Networks
(TFIDF)	Term Frequency – Inverse Document Frequency
(LSTM)	Long-Short Term Memory
(GRU)	Gated Recurrent Units
(PoS)	Parts of Speech

	3
LIST OF FIGURES	2
LIST OF ABBREVIATIONS	2
INTRODUCTION	5
BACKGROUND AND PROJECT DESCRIPTION	7
Immersion in Video Games	7
Character Immersion in Role Playing Video Games	8
Natural Language Processing	9
Conversational Artificial Intelligence	9
Approach	10
LITERATURE REVIEW	11
Language Models	11
Transformers	12
Transfer Learning	12
Sentiment Analysis	13
Limitations of Current Language Models	14
SOLUTION	15
Player-Character Relationships	15
Methodology	15
Results	16
Examining the Preconditions for Unlocking Information	17
Methodology	17
Results	18
Unlocking Information and System Design	20
System Design	20
Generative Model	21
Information Retrieval Model	22
ANALYSIS AND RESULTS	23
LIWC	23
Methodology	23
Results	23
Occurrence of Generated Text in Training Data	25
Methodology	25

	4
Results	25
CONCLUSION	27
Future Work	27
APPENDIX A	28
Histogram and Normal Distribution of Relationship Scores in Taskmaster Dataset	28
APPENDIX B	30
LIWC Comparison between Generated Conversation and Training Data	30
WORKS CITED	32

Introduction

There are an estimated 2.6 billion gamers in the world as of 2020, and this number is expected to exceed three billion by 2023, whilst generated revenue is expected to surpass \$200 billion. (newzoo 2020) With Video Games gaining mainstream popularity in the modern age, game design has been adapting rapidly in order to provide immersive and interactive experiences.

Historically, Video Game Development has always been an excellent testing ground for interesting technologies due to their constrained and virtual environments. Some more recent examples of this include demonstrating 4-dimensional Rigid Body Dynamics in a 3-dimensional slice (Bosch 2020), and using Reinforcement Learning to synthesise motion for quadrupeds in dynamic environments. (Luo, et al. 2020)

As expected, many of the technological implementations used in Video Games has aimed to improve immersiveness for players. However, most of the direction has been focused on increasing graphical realism in order to further immersivity, and there has been a lack of emphasis on narrative aspects of Video Game Development.

In particular, dialogue systems and their natural connection to narrative, has seen little change in the past two decades when compared to their graphical counterparts. Traditional dialogue systems have centred around branching decision trees and multiple-choice inputs, however, in this thesis we explore the use of Natural Language Processing to enhance player interactivity with non-playable characters (NPCs) in the game world.

We focus on providing a toolkit for game developers to train their own Conversational Artificial Intelligences (AI), utilising Transfer Learning to leverage state-of-the-art pre-trained language models such as BERT (Devlin, et al. 2019) and GPT2 (Radford, et al. 2018) in order to produce characters capable of open-domain dialogue.

The use of such dialogue systems has previously shown to increase user engagement (Fraser, Papaioannou and Lemon 2018), however the overall effectiveness of open-domain dialogue conversationalists is still an ongoing research topic. Awards such as the Alexa Prize poses a challenge to create a socialbot that can engage in conversation for 20-minutes and achieve an average rating of 4.0 out of 5.0, however, best efforts so far have only been capable of sustaining conversation for around 7-10 minutes before breaking down and with an average score of around 3.8. (Amazon 2020)

Background and Project Description

Immersion in Video Games

Immersion as studied from the perspective of Video Games (Taylor 2002) can take on many forms but mainly focuses on two distinct types, Diegetic Immersion and Non-Diegetic Immersion.

Diegetic Immersion refers to the immersion created by objects, characters and scenery that are in the game world and viewable by the playable character and hence can be closely linked to the graphical improvements seen in the Video Game industry. However, it is important that we consider dialogue and narrative as similarly important aspects of diegetic immersion.

Non-diegetic Immersion refers to interfaces and heads-up displays offered to the player, however, these elements are not perceivable in the world of the character. In this thesis report we focus on improvements for non-diegetic dialogue interfaces of modern Role Playing Games' which have traditionally implemented multiple-choice decision trees depicted in Figure 1.



Figure 1 - Dialogue Interface in The Elder Scrolls: Skyrim

Character Immersion in Role Playing Video Games

Playable and Non-Playable character design in Role Playing Games (RPG's) lend to the immersion which is created by the Video Game. Previous work on the importance of character design has broken the concept into the notion of Character Immersion, Habitus and additionally the ability to Link Goals and Emotions (Lankoski 2002).

Character Immersion is crucial as it is what enables a player to pose as another entity within the game's existence, requiring the player to attempt to experience what the character is experiencing. This notion, however, operates in parallel with a player's knowledge that they are playing a video game and actions within the game world have no real-world consequences.

In order for game developers to manufacture Character Immersion, they utilize goals and objectives within the framework of the game in order to motivate a player. How clearly a player is able to identify with goals given to the character directly influences the ability for a player to experience what Lankoski calls "**shared emotion**", the notion of feeling what the character would feel in any particular scenario.

Habitus is the concept that individuals have their own way of solving problems. In order for full Character Immersion to be realized, the game must at the same time provide necessary mechanics for players to achieve those goals through a means that is resonant with their Habitus.

Natural Language Processing

At the forefront of the Machine Learning (ML) revolution, the number of Natural Language Processing (NLP) publications has risen significantly in recent years. (Rei 2020) The field itself draws knowledge from linguistics, computer science, artificial intelligence, psychology and philosophy, and can be broadly defined as a machine's ability to derive understanding from natural human language.

The field itself is of particular interest to many theoretical investigators as it is perceived to be a core limitation of current artificial intelligence, and has prevented researchers from unanimously attributing intelligent status to any system, contrary to the narrative told by mainstream media and consumer marketing in the world today.

Despite this, the field has advanced enormously in the past decade and the technology is being increasingly utilised in the real world. Some applications which we see in our daily lives include Machine Translation, Sentiment Analysis of Text, Summarisation and even voice assistants such as Alexa, Google Assistant and Siri.

Conversational Artificial Intelligence

Conversational Artificial Intelligence are interactive agents which users can converse with through natural language. Attempts at creating these agents have largely revolved around two approaches, however both approaches have considerable differences when considering the context within which they are used.

The more commonly utilised approach in the real world is an information retrieval approach where responses from the agent are drawn from a large dataset of responses which the system finds most relevant. This approach generates semantically good responses and is often used in assistant chatbots and contexts where the task space of the agent is relatively narrow.

However, this approach generally fails when the agent is asked to respond to utterances which are outside of the agent's training scope. Hence, open-domain dialogue systems are an approach to conversation which is more focused on general conversing. This approach often produces results which can be semantically incorrect or results where responses can be semantically correct but lack 'sense'. Despite this, there have been significant improvements in this space, mostly spurred by the development of large-scale language models based on transformer architectures (Vaswani, et al. 2017) which have performed significantly better than previous machine learning networks.

Approach

In this thesis report we explore the use of NLP in order to create an open-domain dialogue system, that makes use of sentiment analysis in order to establish a synthetic relationship score between the NPC and the player.

As such, players are able to interface with other NPCs in the game through natural language, expanding the restrictions on dialogue and allowing players to better experience Lankoski's "shared emotion".

More specifically, our approach tackles the laborious task game designer's are faced with when writing thousands of lines of dialogue for standard interactions between players and NPC's. We leverage transfer learning in order to give developers the tools to train conversational agents which are able to capture the essence of a character and deploy them in simple information extraction scenarios when players are faced with collecting game world information through conversation.

Literature Review

Language Models

Like in most areas of Machine Learning, Neural Networks provided a huge boost to the performance of language models. Neural net language models are constructed and trained as probability classifiers that learn to predict a probability distribution,

$P(w_t|context) \forall t \in V$, where w_t represents the next word, the context represents a fixed-sized window of previous words and V represents the vocabulary of the system. (Zhang, Jin and Zhou 2010) This representation of natural language, known as the Bag-of-Words model or more generally as Recurrent Neural Networks (RNN), in conjunction with the use of Term Frequency-Inverse Document Frequency (TFIDF) as a weighting factor in information retrieval, has been the basis of many practical applications of NLP such as machine translation, document search and named entity recognition.

Limitations of these models, however, become apparent when considering the size of the context, where long-range dependencies can often fail to be captured due to insufficient computational power. As a result, we can see explicitly that in conversational use-cases, the longer the conversation continues for, the less contextual information we have to move forward and thus the more likely that the conversation structure will break down. This limitation is more formally known as the vanishing gradient problem for RNNs.

Improvements to RNNs in the form of Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) attempt to alleviate the issue of vanishing gradients by including a Forget Gate, Input Gate and Output Gate. These gates can be simply understood to consider whether or not the previous hidden layer input will be considered in the next iteration or not, and hence more compactly representing the context required to produce the next word.

Transformers

Despite previous improvements to language models and their representations, the most significant improvement in recent years has been the through the development of Transformers in training large-scale language models, exemplified by BERT (Devlin, et al. 2019) and GPT-3 (Brown, et al. 2020). Transformers forgo historical use of recurrence and convolutions, previously used to handle sequence models, and adopt a purely attention based approach to network architecture. (Vaswani, et al. 2017)

Attention mechanisms allow for the modelling of dependencies without regard to their distance in the input or output sequences, alleviating the limitations of recurrent models. As such, the Transformer allows for significantly more parallelization and in the original paper, results stated that new state-of-the-art in translation quality could be reached after being trained for as little as twelve hours.

Transfer Learning

With how expensive and energy exhaustive training a large-scale language model can be, Transfer Learning is a machine learning technique where the trained weights of a general model can be reused as a starting point for training a model on a related task.

This notion can be likened to the way humans learn new things, where we very rarely learn any concept from scratch, but attempt to draw parallels and previous learnings to adapt to a new task. For example, if we are learning how to ride a motorbike, we may leverage previous knowledge we have on how to ride a bike in order to learn faster.

As such, large language models such as Google's BERT (Devlin, et al. 2019) can be trained once, and then leveraged to solve a variety of downstream language tasks from Machine Translation to Question-Answering.

Other large-scale language models such as GPT-3 consist of 175 billion parameters and cost roughly \$4.6 million to train (Dickson 2020). GPT-3 demonstrated by scaling up the exact same model architecture from GPT-2, they were able to greatly improve task-agnostic performance.

Hence it becomes evident the benefits which can be leveraged through Transfer Learning, especially for resource and budget constrained individuals and organisations.

Sentiment Analysis

A common form of text classification, Sentiment Analysis is a tool for analysing incoming messages and deducing whether the underlying sentiment is neutral, positive or negative. Most techniques derived for this topic involve comparing occurrence of words in speech to a lexicon of manually annotated adjectives and other descriptive words. As such, we may consider Sentiment Analysis as an attempt to quantify or measure one of many aspects of speech through linguistic inquiry.

Machine Learning itself is used for sentiment analysis, primarily to improve and automate low-level text analysis such as Parts of Speech tagging (PoS), however a large part of the analytics involved in high level Sentiment Analysis remains rule based.

Hence, in our research we find that Sentiment Analysis data is often grouped to determine overall sentiment towards a topic. Use of Sentiment Analysis in the current real-world largely involves utilising sentiment analysis to classify social media content in order to deduce sentiment around an event or brand. (Bakshi, et al. n.d.)

Limitations of Current Language Models

Although there have been significant improvements in Natural Language Processing in a variety of downstream tasks, it must be considered that there are still limitations to the current designs. As discussed in previous sections, despite the introduction of transformer architecture, language models still often fail to capture long-range dependencies in text without utilising large-scale training networks.

Furthermore, despite demonstrating that scale can drastically improve the performance of a language model such as the previously discussed GPT-3, these models still display significant weaknesses in some tasks which appear trivial.

Firstly, GPT-3 has demonstrated the ability to complete many tasks, many of which developers and scientists did not believe were possible, even demonstrating the ability to convert natural language into code (Bussler 2020).

Interestingly, GPT-3 struggles with mathematics, which is surprising as we often associate computing with a proficiency in mathematics. GPT-3 has demonstrated an inability to draw causal links between seemingly simple mathematical concepts such as arithmetic. If we examine the results from the GPT-3 (Brown, et al. 2020) we see that the model performs extremely well in the realm of two digit addition, achieving accuracy upwards of 80% which is a large improvement from GPT-2 which was simply unable to complete the task altogether. However, if we expand the task to include three, four and even five digit addition, the accuracy drops below 40% in most cases.

What I believe that this exhibits, is that language models are still innately unable to reason but rely on pattern matching in order to provide the results it displays. GPT-3, which was trained on a scrape of the internet itself, is exposed to many more occurrences of numbers which exist in the two-digit range, and thus does well in those tasks. Numbers of higher order are simply less frequent on the internet and even less frequent in the context of arithmetic, thus reducing its accuracy when computing tasks related to them.

As such, it has become a topic of debate amongst deep learning researchers, whether scale can lead to exponential growth in performance inevitably, or whether other models and designs are required in order to introduce reasoning and logic capabilities.

Solution

Player-Character Relationships

In this section we explore the use of sentiment analysis in order to establish relationships between the playable character and the non-playable characters in the game world. Current open-dialogue conversational AI's do not have the capabilities of modelling conversational relationships, and often struggle to draw abstract links between concepts during conversation. In order to replicate such an experience, we use a "relationship score" as a proxy function of user-sentiment scores from previous utterances.

Methodology

In this section we make use of a previously trained conversational AI system in order to test the viability of a relationship score.

Firstly, the relationship score is initialized to zero, meaning that no relationship has yet been established. The score is intended to be normalized between zero and one, and thus changes to the relationship score must also be normalized.

We use a simple shell interface in order to converse with the conversational AI system, which begins by prompting input from the user. The input is processed by the model, and a response is generated and then displayed on the screen before prompting for more input from the user.

Next, we make use of Microsoft Azure Text Analytics in order to conduct sentiment analysis on each utterance. As each utterance is inputted by the user or generated by the model, it is passed to a Microsoft Azure Text Analytics API endpoint and from there the sentiment analysis is executed in the cloud.

The data returned by the system includes an overall sentiment value defined as positive, neutral or negative. Each of the above components, also includes a score from zero to one to two decimal points. Using this data, we then make changes to the relationship score according the following function:

$$R_{t+1} = neu \times R_t + (pos - neg)$$

Where R is the relationship score, neu , pos , neg are the sentiment scores returned by the webhook, and t is the timestep of each state in the conversation.

Results

The originally proposed algorithm ended up displaying some critical flaws. It was found in border cases, where the neutral component of the sentiment analysis ended up equalling to 0, this would reset the running relationship score. Hence, we proceeded to trial some different algorithms to see what their effects would be.

The next algorithm proposed was $R_{t+1} = R_t + (neu + pos - neg) \times 0.1$ with the rationalisation behind this being that we wish for a minimum ten iterations of user input to occur before a relationship score reaches a maximum of 1.0. This also removed the previous issue, where the relationship score was largely dominated by the neutral component of the sentiment analysis. However we found that with this algorithm, the neutral component of the sentiment analysis still had a large impact on the final result, having essentially the same weighting as the positive and negative components and thus we propose another algorithm.

$$R_{t+1} = R_t + (neu \times 0.3 + pos - neg) \times variability$$

In this trial, the neutral component of the sentiment analysis was scaled down to 0.3, thus more appropriately reflecting the sentiment of each utterance and found that relationship scores changed with common-sense rationalisation under a series of neutral, positive and negatively dominated utterances.

We further extended this algorithm to allow for more flexibility, redefining the 0.1 as the *variability* factor to allow developers the ability to affect how quickly a user relationship score may change under their own use cases, serving as an upper bound for how much the relationship score may change by per turn in conversation.

We find that utilising a variability score inversely proportional to the average number of turns in a conversation produces the best results.

$$variability \propto \frac{1}{average\ turns}$$

Examining the Preconditions for Unlocking Information

In this section, we examine the use of a relationship score by applying our system to a corpus of conversations and then further analysing how the relationship score appears to change within different stages of conversation. More explicitly, we wish to examine how willing a person is to share information depending on the relationship score between two conversing people.

Methodology

We analyse how the relationship score changes throughout task-oriented conversations, meaning conversations which occur in order to achieve some end goal. The dataset chosen for this section was the Taskmaster dataset from Google Research (Byrne, et al. 2019), containing 13,215 task-based dialogues, with each conversation falling into one of six domains such as ordering pizza or ordering movie tickets.

Task-oriented conversations are often the type of conversations which occur throughout role-playing video games, and thus was a good starting point for exploring the role of the relationship score.

We then take the first 100 conversations from the dataset and manually annotate them for the point in the conversation where it appears there is a meaningful transfer of knowledge. For example, if the task domain was to order pizza, the transfer of meaningful knowledge is understood to be the point at which the order is successfully placed or rejected.

A python script can then be run in order to calculate the relationship score at each turn of a conversation with the *variability* factor associated with the relationship score initially set at 0.2.

We can then compare the relationship score before and after the point of knowledge transfer for all conversations.

Results

It was found that when the variability factor was set to 0.2, the results displayed a high mean post knowledge transfer relationship score of 0.82, with a relatively large standard deviation of 0.26. The difference in relationship score before and after the transfer of knowledge was minimal as shown in Figure 2, with standard deviations being effectively equal.

Conversation Stage	Mean	Standard Deviation
Post Knowledge Transfer	0.82	0.26
Pre Knowledge Transfer	0.81	0.26

Figure 2 - Analysis of Relationship Score with Variability = 0.2

As we can see in Figure 3, the histogram is skewed heavily to the right, with many scores reaching the maximum relationship score of 1.0. We believe this was largely due to variability factor being too high as the average number of turns of conversation in the dataset was 21.85.

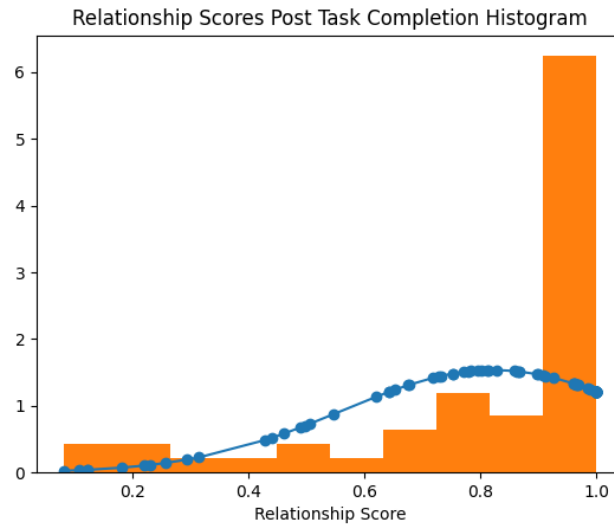


Figure 3 - Histogram and Normal Distribution of Relationship Scores, Post Task Completion with Variability = 0.2

Thus, we continued to explore the dataset by reducing the *variability* factor to 0.1. With further exploration, the post and pre knowledge mean transfer relationship scores dropped to 0.56 and 0.53 respectively, however standard deviation remained steady at 0.26.

Conversation Stage	Mean	Standard Deviation
Post Knowledge Transfer	0.56	0.26
Pre Knowledge Transfer	0.53	0.26

Figure 4 - Analysis of Relationship Score with Variability = 0.1

The histogram in Figure 5, shows a much more distributed curve, however we still see that there is a large distribution at the minimum and maximum ends of the relationship score. We believe this is largely due to the nature of task-oriented conversations, as very often meaningful knowledge can be transferred before any relationship has been established between conversing parties. Similarly, we found that at times, conversers can quickly raise the relationship score if they are being polite and cordial.

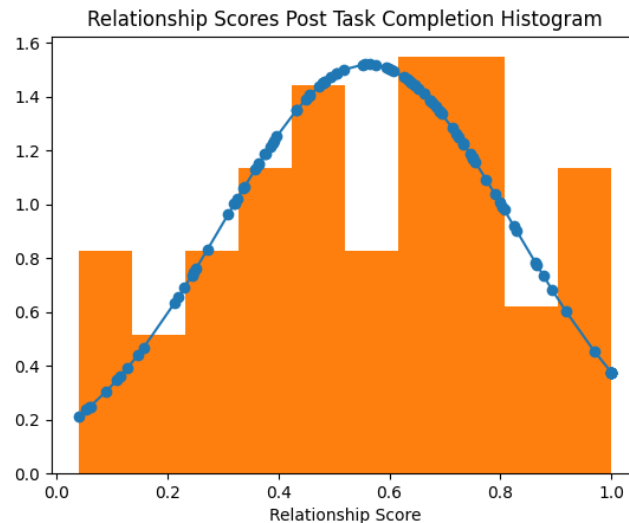


Figure 5 - Histogram and Normal Distribution of Relationship Scores, Post Task Completion with Variability = 0.1

Unlocking Information and System Design

In this section we use the insight into relationship score established from the previous section in order to unlock information about the game and to simulate an experience where, as your relationship with a character grows, so too does their willingness to share information.

System Design

In order to simulate an information base about the game world, each playable character is provided with a predetermined list of utterances intended to serve as answers to possible questions. As a result, this section employs an information retrieval approach to conversational AI in addition to the open-dialogue generative approach.

Upon instantiation of the agent, we randomly generate a relationship score threshold along the distribution found in the previous section. When conversing with the agent, the relationship score is calculated turn by turn and once the threshold is reached, the agent effectively ‘unlocks’ the knowledge base and from this point onwards an information retrieval-based approach is also incorporated. Prior to the unlocking of information, the agent retrieves responses from a generative conversational AI.

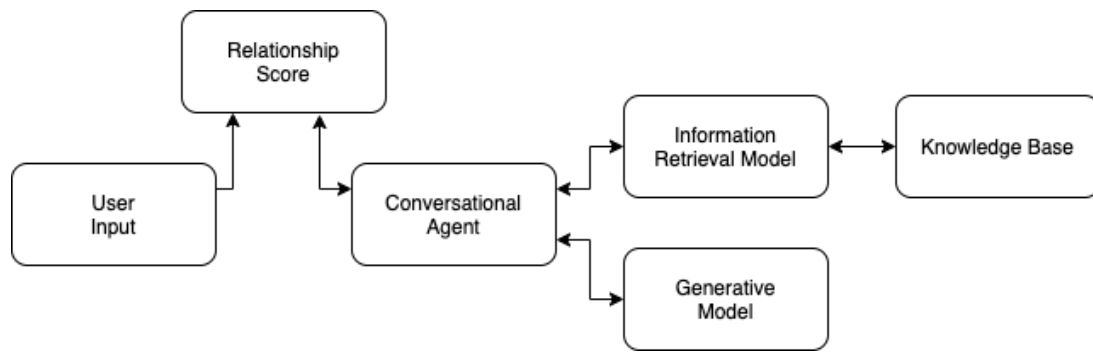


Figure 6 - System Design of Conversational Agent

Generative Model

As discussed previously in this report, the fine-tuning of pre-trained language models allows us to utilise the performance of large-scale language models without training from scratch. The open-source Hugging Face transformers library provides access to pre-trained language models with ease, allowing implementations in both major machine learning libraries, PyTorch and Tensorflow.

Our generative model made use of one such language model, DialoGPT (Zhang, et al. 2019), which is already a further fine-tuning of GPT-2 on 147M conversation-like exchanges extracted from Reddit comment chains. Hence this model is already fine-tuned for the task of generating conversational responses.

We further fine-tuned the above language model with thousands of lines of text from a TV Show corpus, attempting to capture personality traits and conversational styles relating to a context or character.

```
User: What's up, Rick!  
[Generative Model Response]  
AI: Whoa whoa wh-what the hell hell hell?  
User: You're not looking so good.  
[Generative Model Response]  
AI: That's enough! We need to talk!  
User: What is it now?  
[Generative Model Response]  
AI: I think it's time to fold the two of you.  
User: What does that even mean?  
[Generative Model Response]  
AI: It means we're F'd in the A.  
User:
```

Figure 7 - Example Conversation from model fine-tuned on Rick and Morty corpus

Each fine-tuning process on a different corpus intended to train a separate model for each character, with results demonstrating that the generative model is reasonably imitating the speech patterns of the training data. Analysis on the results of fine-tuning are discussed later in this report.

Information Retrieval Model

An information retrieval model is queried under two conditions. First, the relationship score at the point in conversation must be above the threshold score established during the instantiation of the agent. Second, the input user utterance must have a cosine similarity above a certain threshold with the knowledge base. We found that manually tweaking the similarity threshold depending on the knowledge base was necessary in order to get good results.

In order to calculate the similarity between the user input and the knowledge base, the input is first tokenized and then a TFIDF is calculated for the sequence. This TFIDF is then compared with the TFIDF of the knowledge base by calculating a cosine similarity. We then proceed to return the response with the highest cosine similarity if it is above a certain threshold.

If nothing fits the above criteria, then we revert back to gathering a response from the generative model in order to continue the conversation.

Analysis and Results

LIWC

Systematic associations between personality and individual differences in word use have been demonstrated in past works. (Yarkoni 2010) Correlations between Big Five personality traits and LIWC categories (Pennebaker and Tausczik 2010) demonstrate statistical significance in 66 analysed categories. Here we utilise LIWC in order to consider the similarities between the training corpus and conversation generated by agents in our system.

Methodology

In order to apply LIWC analysis to our conversational agents, we begin by instantiating two instances of agents and generating 200 conversations, each for 20 turns of conversation. These samples were then used to compare the personality and syntactical characteristics of the conversational agent's speech, with the original training data.

Results

Using the 2015 LIWC lexicon in order to gather counts, we found 47 categories which remained similar to those results found in past works. The analysis shows that for all 47 categories, the ground truth LIWC remained within one standard deviation of the mean LIWC from the generated conversations.

LIWC Category	\bar{x} Generated	σ Generated	Ground
Pronoun	24.31	3.75	21.70
First Person Singular	5.80	2.49	5.70
First Person Plural	1.92	1.64	1.72
First Person	8.68	2.41	7.17
Second Person	5.95	2.36	5.14
Third Person	0.50	1.23	0.65
Article	5.35	2.07	5.11
Preposition	9.40	2.07	10.09
Social	14.70	3.85	13.88
Cognitive Processes	10.36	3.71	11.25
Past Tense	2.63	2.00	2.84
Present Tense	17.54	3.28	15.76
Future Tense	1.52	1.26	1.75
Space	6.16	2.17	6.10

Figure 8 - Comparison of LIWC Categories between training data and generated data

It should be noted that for many of the more obscure categories analysed, the small size of each sample (20 turns of conversation) lead to rather small counts and larger than preferred standard deviations. However, in the attempt to generate longer conversations, we found that conversations lasting longer than 20 turns often lead to a breakdown in structure and thus we determined that analysis unsuitable.

Despite this, for many significant categories depicted in Figure 8, the conversational agent appears to do well in capturing the syntactical structure of the training data. This makes sense if we consider the underlying tokenisation of input into NLP models, making grammatical data visible to the training model.

The agent that was used for this analysis was trained on a corpus of Rick and Morty scripts and thus demonstrated a higher count in references to space in comparison to other descriptive categories. The results also display a higher use of present tense words, which demonstrates that the language model is fine-tuned for conversation.

Full results of analysis can be viewed in appendix B.

Occurrence of Generated Text in Training Data

In this section we explore the generated conversations and check to see if generated utterances make appearances in the original training data.

Methodology

We iteratively compute a cosine similarity between each line of generated conversation (4000 total utterances) and each line in the original training data. For each line of generated conversation, we then return the line from the original corpus that computed the highest cosine similarity.

Results

Most lines of generated conversation did not appear directly in the training data, instead often emulating the style of the training data, with small syntactical changes in sentence structure.

In most cases, phrases which occur in the training data appear within generated utterances, usually at the start. The remainder of the generated text however begins to deviate from the script, which makes sense if we consider the GPT-2 model structure which predicts what the next word token is given an input token sequence.

Original Line:	“That was amazing, Morty. Oh, my God. Wow. Okay, let’s wrap this. “
Generated Line:	“That was amazing, Morty. Oh, my God. Wow. I can’t believe we hit 70 Jessica. Wow!”

In some cases, we see named entities from the original training data replaced with more contextually appropriate named entities, demonstrating that the agent appears to at least be drawing dependencies from the context of the conversation.

Original Line: "Father, can we please go to the Ascension?"

Generated Line: "Rick, can we please go to the Ascension?"

Original Line: "That's enough Bradley. We don't want you injuring your ball-throwing arm"

Generated Line: "It's okay, Morty! We forgive you! We don't want you injuring your ball-throwing arm"

Conclusion

Overall, findings appear to demonstrate that fine-tuning pre-trained language models can appropriately emulate the personality traits of underlying training data and then utilise the syntactical structure of sentences to generate responses as a conversational agent. Hence, using existing corpora of character dialogue, game developers can create conversational agents for use in their games

We begin an exploration of how video games, especially role playing games, can leverage the use of language models in providing realistic narrative and dialogue focused systems. In our investigation we provide one such example of how dialogue itself can lead to discovery and exploration within the game world.

Future Work

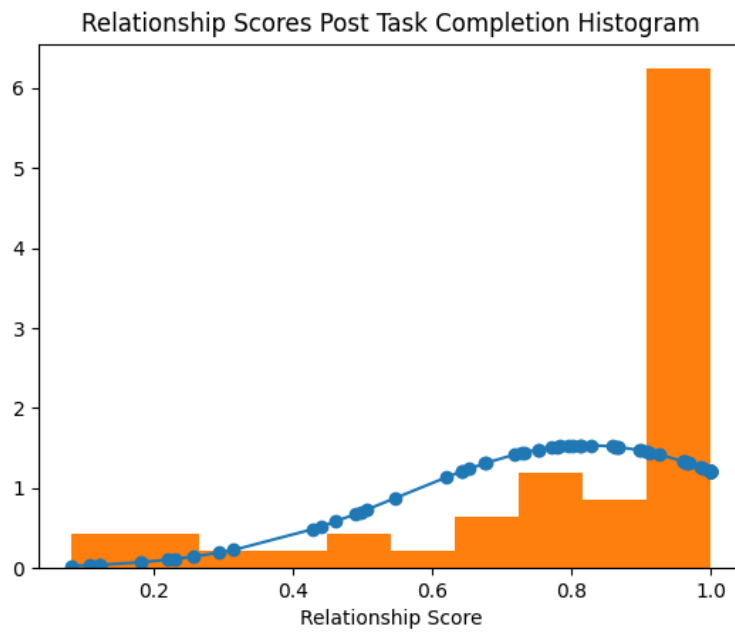
Whilst capturing personality from a large corpus appears possible given quality training data, the availability of such corpora may not be available to all game developers. The generation of conversational agents from more simple parameters would be highly beneficial for the usability of conversational agents in gaming.

Additionally, whilst generative approaches may be able to converse in large context scopes, there are improvements that can be made to information retrieval based approaches to video games. For example, intent can be captured in the user input, which can be translated into changes in game state such as for the management of inventory for a shopkeeper.

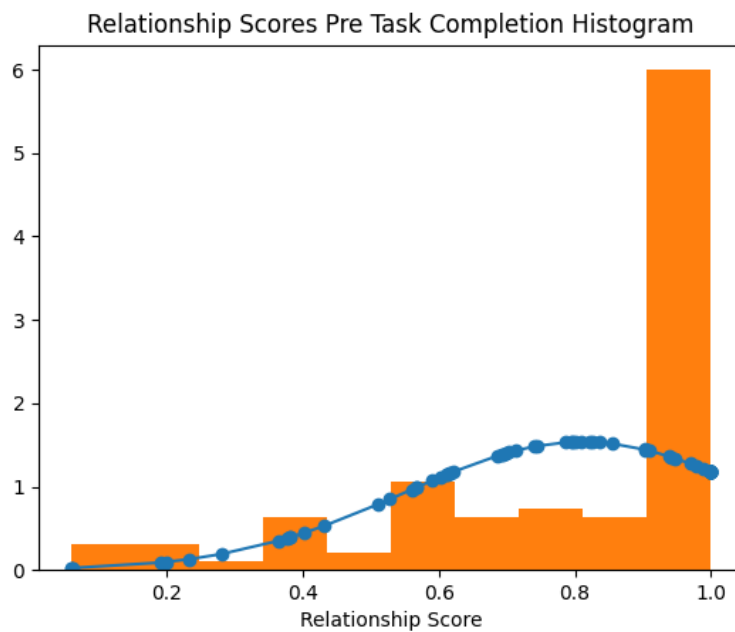
Appendix A

Histogram and Normal Distribution of Relationship Scores in Taskmaster Dataset

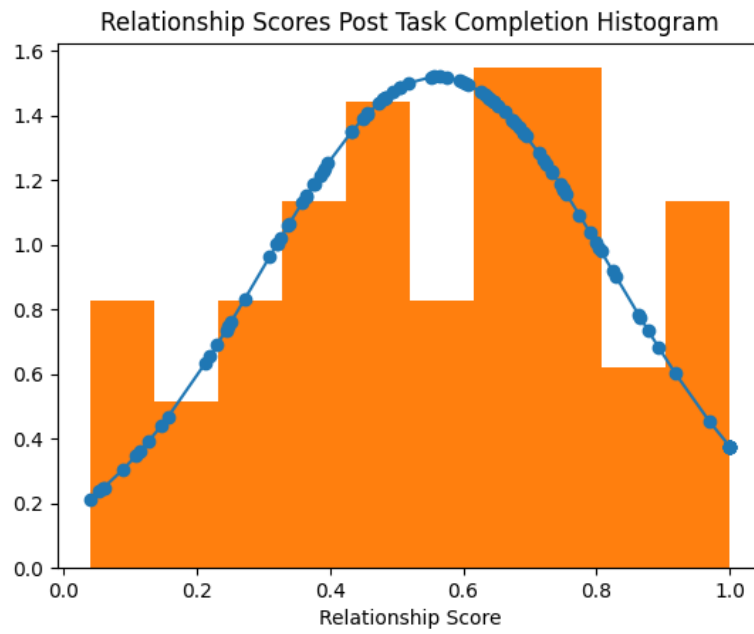
Post Knowledge Transfer (Variability = 0.2)



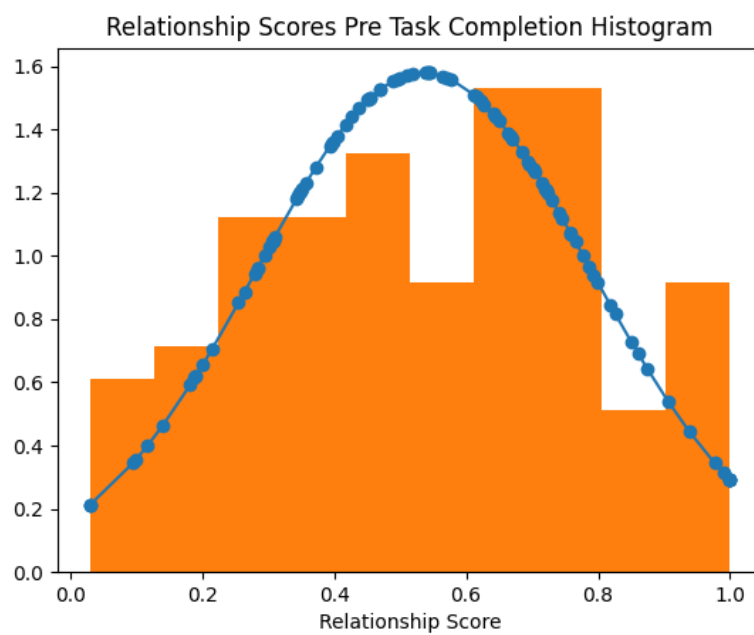
Pre Knowledge Transfer (Variability = 0.2)



Post Knowledge Transfer (Variability = 0.1)



Pre Knowledge Transfer (Variability = 0.1)



Appendix B

LIWC Comparison between Generated Conversation and Training Data

LIWC Category	\bar{x} Generated	σ Generated	Ground
Pronoun	24.31	3.75	21.70
First Person Singular	5.80	2.49	5.70
First Person Plural	1.92	1.64	1.72
First Person	8.68	2.41	7.17
Second Person	5.95	2.36	5.14
Third Person	0.50	1.23	0.65
Article	5.35	2.07	5.11
Preposition	9.40	2.07	10.09
Negation	2.50	1.37	2.54
Number	0.92	1.23	1.31
Affect	6.52	2.57	6.14
Positive Emotion	3.31	1.98	3.50
Negative Emotion	3.22	1.87	2.61
Anxiety	0.38	0.62	0.31
Anger	1.86	1.44	1.28
Sadness	0.34	0.52	0.38
Social	14.70	3.85	13.88
Family	0.59	0.84	0.89
Friend	0.47	0.60	0.46
Cognitive Processes	10.36	3.71	11.25
Insight	2.83	1.77	2.42
Causation	1.18	1.12	1.55
Discrepancy	1.46	1.24	1.70
Tentative	1.94	1.50	2.44
Certainty	0.99	0.96	1.61
Perception	1.83	1.42	2.11
Seeing	0.66	0.90	0.78
Hearing	0.57	0.87	0.70
Feeling	0.44	0.56	0.47
Sexuality	0.30	0.53	0.39
Ingestion	0.22	0.50	0.51
Achievement	0.92	0.86	0.91
Past Tense	2.63	2.00	2.84
Present Tense	17.54	3.28	15.76
Future Tense	1.52	1.26	1.75

Motion	2.45	1.47	2.50
Space	6.16	2.17	6.10
Time	2.80	2.11	3.67
Work	0.81	0.83	1.07
Leisure	0.63	0.82	0.89
Home	0.37	0.62	0.46
Money	0.12	0.30	0.30
Religion	1.46	1.50	0.56
Death	0.51	0.76	0.41
Swear Words	1.34	1.28	0.97
Assent	1.39	1.15	1.00

Works Cited

- Amazon. 2020. *Alexa Prize*. <https://developer.amazon.com/alexaprize>.
- Bakshi, Rushlene Kaur, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. n.d. "Opinion mining and sentiment analysis ." *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE Xplore.
- Bosch, Marc Ten. 2020. "N-dimensional rigid body dynamics." *ACM Transactions on Graphics*.
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. *Language Models are Few-Shot Learners*. OpenAI.
- Bussler, Frederik. 2020. *Will The Latest AI Kill Coding?* . 21 July. <https://towardsdatascience.com/will-gpt-3-kill-coding-630e4518c04d>.
- Byrne, Bill, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. *Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset*. arXiv.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Google.
- Dickson, Ben. 2020. *The GPT-3 economy* . 21 September. <https://bdtechtalks.com/2020/09/21/gpt-3-economy-business-model/>.
- Fraser, Jamie, Ioannis Papaioannou, and Oliver Lemon. 2018. "Spoken Conversational AI in Video Games: Emotional Dialogue Management Increases User Engagement ." *IWA '18: Proceedings of the 18th International Conference on Intelligent Virtual Agents*. 179-184.
- Lankoski, Petri. 2002. "Character Design Fundamentals for Role-Playing Games." In *Beyond Role and Play: Tools, Toys and Theory for Harnessing the Imagination*. Ropecon RY.
- Luo, Ying-Sheng, Jonathan Hans Soeseno, Trista Pei-Chun Chen, and Wei-Chao Chen. 2020. "CARL: Controllable Agent with Reinforcement Learning for Quadruped Locomotion." *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2020)*.
- newzoo. 2020. *Newzoo Global Games Market Report 2020*. newzoo.
- Pennebaker, James W., and Yla R. Tausczik. 2010. "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods." *Journal of Language and Social Psychology* 24-54.

- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Armodei, and Ilya Sutskever. 2018. *Language Models are Unsupervised Multitask Learners*. openAI.
- Rei, Marek. 2020. *ML and NLP Publications in 2019* . 6 February.
<http://www.marekrei.com/blog/ml-and-nlp-publications-in-2019/>.
- Taylor, Laurie N. 2002. *VIDEO GAMES: PERSPECTIVE, POINT-OF-VIEW, AND IMMERSION* .
 Graduate School of the University of Florida.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention Is All You Need*. Google.
- Yarkoni, Tal. 2010. "Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers ." *Journal of research in personality* 363-373.
- Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. 2010. "Understanding bag-of-words model: A statistical framework." *International Journal of Machine Learning and Cybernetics* 43-52.
- Zhang, Yizhe, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. *DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation* . Microsoft.