



UPPSALA
UNIVERSITET

UPTEC IT 20027

Examensarbete 30 hp
Juni 2020

Response Generation Using Large-scale Pre-trained Language Models

Jakob Nyberg

Institutionen för informationsteknologi
Department of Information Technology

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Response Generation Using Large-scale Pre-trained Language Models

Jakob Nyberg

In this project I studied how generative neural language models can be used for response generation. The purpose of the model is to generate responses for a social robot, instead of having responses be authored and evaluated by crowd-sourced workers. To achieve this task, I train a large-scale pre-trained neural language model on the collected data. I trained six model variations to study the changes in utterance quality, the models vary in the amount of pre-training they have. I also test three different decoding methods for the same purpose. One of the model variations utilize multi-task learning during training, where the model performs other tasks alongside response generation. The utterances produced by the models were evaluated through crowd-sourced human evaluation. Utterances were shown by the evaluation to be of roughly equal quality to the original utterances it was trained to replicate. The results show that a large-scale language model may be a viable alternative to crowd-sourced authoring and evaluation of utterances, reducing costs and providing more reliable results.

Handledare: Ramesh Manuvinakurike
Ämnesgranskare: Ginevra Castellano
Examinator: Lars-Åke Nordén
UPTEC IT 20027
Tryckt av: Reprocentralen ITC

1 SAMMANFATTNING

I detta projekt studerar jag hur neurala språkmodeller kan användas för att generera yttranden för en robot. Forskningsgruppen Social Robotics på Uppsala Universitet har bedrivit en studie där en människa och en robot spelar ett spel tillsammans. Spelet är kollaborativt och går ut på att roboten ska gissa vilket land människan beskriver. När roboten gissar rätt land får paret ett poäng. Spelet är tidsbegränsat, och paret ska samla så många poäng de kan under spelets gång.

Under spelets lopp pratar roboten med människan. Roboten kan bland annat kommentera på det nuvarande poängläget, ställa frågor till människan eller säga hejdå när spelet är slut. Roboten har även ett humör som beror på hur väl spelet går.

Roboten bestämmer vad den ska säga med ett dialogsystem, som består av flera komponenter. När roboten ska säga något väljer systemet ett passande yttrande från en lista av yttranden. Dessa yttranden är skrivna av människor med hjälp av Amazon Mechanical Turk, en massbaserad problemlösningstjänst. Användare blev presenterade med en beskrivning av situation samt robotens humör i den situationen. Deras uppgift var därefter att skriva vad roboten skulle kunna säga i den situationen. Svaren evaluerades därefter av andra personer för att sälla ut dåliga svar.

Det finns problem med att producera yttranden med den här metoden. Det kostar pengar att betala författare att skriva yttranden för olika tänkbara scenarier, som sedan även behöver bedömas av ytterligare personer. Det är inte heller säkert att det som skrivs är användbart, då svaren kan vara stötande eller meningslösa. Metoden innebär även att roboten inte kan generalisera, utan enbart kan säga saker för de scenarion som yttranden samlats in för.

Frågan som ställs i detta projekt är om en neural språkmodell kan användas i stället, eller som ett komplement till yttranden skrivna av människor. Är dessa maskinskrivna yttranden jämförbara, eller till och med bättre än de som samlats in via Mechanical Turk?

För att besvara detta har jag använt mig av den storskaliga generativa neurala språkmodellen GPT-2. Jag tränar sex olika modellvarianter, med den främsta skillnaden i hur mycket data de förtränat på. Jag studerar även hur kvalitén förändras av att använda olika avkodningsmetoder, samt extra träningsmål.

För att bedöma kvalitén av de maskingenererade yttrandena utfördes en studie där yttranden blev bedömdes med samma frågor som användes för att bedöma kvalitén av de insamlade yttrandena. Resultaten från studien visar att modellerna kan producera yttranden som är på ungefär samma nivå av relevans som ursprungliga yttrandena.

CONTENTS

1	Sammanfattning	
2	Glossary	1
3	Introduction	2
3.1	Problem background	3
3.2	Task	4
4	Theory and Background	5
4.1	Open-domain Dialog and Task-Oriented Dialog	5
4.2	Language Models	6
4.3	Loss & Validation Metrics	7
4.3.1	Cross-entropy and Perplexity	7
4.3.2	BLEU	8
4.4	Transformer Networks & Attention	8
4.5	Generative Pre-trained Transformer (GPT)	9
4.6	Decoding Methods	9
4.7	Response generation using Transformers	11
4.8	The choice of an intermediary dataset	12
4.8.1	Emotion Representation	12
4.9	Ethics of neural language models	13
4.10	Reflections on neural language models	14
5	Related Work	15
6	Datasets	15
6.1	Empathetic Dialogues	16
6.2	Neil Data	17
7	Implementation	20

7.1	Data Management	20
7.2	Training	21
7.2.1	Language modeling	22
7.3	Multi-task learning goals	22
7.3.1	Next-sentence prediction/Multiple choice prediction . . .	22
7.3.2	Emotion classification	23
8	Evaluation Methods	24
8.1	Automated metrics	24
8.2	Human Evaluation	24
8.2.1	Phase 1: Comparing decoders.	25
8.2.2	Phase 2: Comparing models	25
9	Results	26
9.1	Automated Metrics	26
9.1.1	Perplexity & Token Accuracy	26
9.1.2	BLEU scores	26
9.1.3	Confusion matrices	29
9.2	Human Evaluation Results	31
9.2.1	Phase 1 results	31
9.2.2	Phase 2 results	33
10	Discussion	37
10.1	How do the models compare to one another?	37
10.2	How well do the models fit the data?	38
10.3	How are the models perceived by humans?	39
10.4	Reflections on the implementation	40
10.5	Reflections on human evaluation using crowd workers	41
10.6	Future work	42
11	Conclusion	43

12 Acknowledgments	44
A Training Hyperparameters	48
B Rejection rates for Seen and Unseen scenarios	48

2 GLOSSARY

Tensor Multidimensional matrix.

Tokenization A process of segmenting a sequence into parts, called tokens.

Multi-Task Learning Machine learning with more than one goal during training.

Language Model Probability distribution of words in a sequence.

Decoding Scheme A method of converting word probabilities into words.

Loss Function A function that measures model error.

Training The process of fitting a model to a set of data.

Pre-training The process of training a model on a, usually more general, set of data before training on problem specific data.

Fine-tuning Training a pre-trained model on more specific data.

Dialogue Agent A program that communicates with a user using text or speech.

Dropout Regularization method that disables nodes in a neural network with a given probability.

Token A symbol or subset of symbols in a sequence.

RNN Recurrent neural network.

MC Multiple choice.

EC Emotion classification.

GPT Generative Pre-trained Transformer.

ED Empathetic Dialogues.

Neil The robot for which utterances are generated for.

ML Machine Learning.

MT Multitasking, see *multi-task learning*.

3 INTRODUCTION

Generative machine learning models are a novel byproduct from the rise of ML models as estimators and classifiers. Based on large collections of images, music or text, these models generate new data not existent in the original set. The results are not always perfect, as seen with the cats in Figure 1, but are sometimes shockingly similar to the real deal. The generative models are often based on neural networks, a machine learning method intended to mimic neurons, and tend to be very large in terms of the number of parameters they have. Generative neural networks have also been applied to the field of language processing, for purposes such as chatbots or translators. Large-scale models have demonstrated the ability to generate text that appear almost human-like in quality, as shown in Figure 2.



(a) An image of a fairly realistic cat.

(b) An image of a weird cat.

Figure 1: Two examples of images, depicting cats that presumably do not exist, produced by generative adversarial networks.

This thesis by Jakob Nyberg is about **the importance of the human voice, and about the importance of how people communicate with each other and with the environment.**

Written by Transformer · transformer.huggingface.co 

Figure 2: A description of a thesis that does not exist, generated by a neural language model. Text in bold was generated by the model.

In this project I study how a machine learning based generative model can be used to produce responses intended for a social robot. The responses should incorporate information from both the situation the robot is in, as well as the mood it has.

3.1 PROBLEM BACKGROUND

The Social Robotics Lab at Uppsala University have performed a set of experiments to study human-robot interactions. Humans play a cooperative map-based country-guessing game with a robot companion, nicknamed *Neil*. In each round the human describes a given country, and the robot tries to guess which country is being described. The game is timed, and the human-robot pair is scored based on the number of correct guesses by the robot. The human and robot have a brief social interaction before and after the game, where they may discuss things like the human’s profession or whether they enjoyed the game. An image of an ongoing game can be seen in Figure 3.



Figure 3: A person playing the geography game with the robot.

The robot has conversational capabilities, using an existing semi-automatic dialog system, and interacts with the human companion by spoken utterances. These utterances are intended to be reflective of the state of the game and the robot’s mood. The mood of the robot ranges on a scale from frustrated to excited, and depends on the performance of the team. If the team performs poorly, the robot will get more frustrated. On the other hand, the robot gets more excited when the game goes well.

One of the challenging problems in the experiment is that of natural language generation, where the robot has to produce utterances relevant to the context both emotionally and content-wise. Currently the lines available to the dialogue system were collected using the Amazon Mechanical Turk crowdsourcing platform. This is an approach called “Semi-situated Learning”, where crowd workers are shown a textual description of the game state and asked to write a line the robot might say in that situation given its mood [1]. The lines are then evaluated in a second stage, by other crowd workers, to verify their quality. The crowd-sourced

approach allow for rapid collection of utterances for many different scenarios, faster than a single author can produce. The crowd-sourcing approach also produces more variety in the utterances, since different workers with different experiences are likely to produce different responses to the same scenario [1].

When using crowd-sourced authoring, there is a non-zero possibility of workers producing lines that do not fulfill the specified levels of quality. Utterances may be considered as too offensive, or irrelevant to the context. Leite et al. [1] had a 7% rejection rate of collected utterances. Paetzel et al. [2], which also used crowdsourced utterances, rejected 5-7% of utterances between different categories.

In the geography-based game used for this thesis, collected utterances were evaluated in two stages. In the first stage, Turkers judge how relevant the utterance is to the given context and how offensive it may be. In the second stage, Turkers were asked to evaluate how excited or frustrated the speaker of the utterance appears to be. Ideally the intended emotion of the utterance should match the perceived emotion, but this is not always the case. 14% of utterances collected for Neil were rejected for either being irrelevant or too offensive.

Collecting utterances by having humans write and evaluate them is costly, collecting the 1512 utterances in the Neil dataset cost roughly 3200\$. The cost limits the amount of scenarios that have utterances written for them. Having a limited number of scenarios reduces the robot’s conversation abilities. If an unseen scenario occurs or a human responds in an unexpected way, the robot will have nothing fitting to say.

3.2 TASK

In this project, I investigate if the process of generating utterances for Neil can be automated using machine learning. The approach I decided on was using a machine learning based *language model* that can generate text based on a given context. The context in this case is the textual description of the situation the robot and human are currently in, along with the robot’s mood. The problem is thus treated as a machine learning problem, which asks how a model can best be fitted to the already collected data. This leads to the questions:

1. How can we train a model that fits the utterance data well?
2. How are the utterance produced by the model perceived by humans?
3. How does the utterances from different model variations compare to each other?
4. How do the model utterances compare to those written by humans?

To answer these questions, I trained six model variations to reproduce utterances from the dataset. The models are validated by their ability to produce utterances for previously unseen scenario descriptions, to address the issue of the robot being limited only to known scenarios. I evaluate the model’s response producing capabilities using a combination of automated metrics and human evaluation, to see how the models compare to one another and how the utterances are perceived by humans.

Ultimately, connecting back to the original goal of automating utterance production, I use the results to see how the utterances produced by a machine learning model compare to those originally written by humans on Mechanical Turk.

4 THEORY AND BACKGROUND

In this section I present theory relevant to understanding neural language models and response generation. I also present some recent neural language models that have been used for response generation.

4.1 OPEN-DOMAIN DIALOG AND TASK-ORIENTED DIALOG

Dialogue tasks are often divided into two major categories, *task-oriented* and *open-domain* dialogue [3]. Task-oriented dialogue is a conversation that is done with a specific goal in mind. An example of this is handling a human making a restaurant reservation, where the agent has to obtain the number of guest and what time they will arrive at. The objective of a task-oriented dialog agent can often be redefined as a numeric training goal, for example the minimization of time or number of dialog lines required to obtain the relevant information.

Conversely, open-domain dialogue is a conversation without a clear end goal, such as social chat or small talk. There are often many equally appropriate responses an agent can produce, which makes objective scoring of utterances difficult. It is hard to translate what makes a conversation “good” with a numerical metric that can be optimized [3]. Open-domain dialogue agents are thus much more reliant on human ratings for validation [4], as humans can subjectively rate things like naturalness or consistency. The dialog for Neil is task-oriented in the sense that the robot has an objective to answer questions correctly and win in the game, but the utterances I focus on in this project are ones that are spoken in between those questions. The robot also chats before and after the game, in a social manner. The problem may thus be considered as open-domain dialogue.

Whether it is intended for open-domain or task-oriented dialogue, a dialogue agent needs some mechanism to produce the utterances it wants to say. The direction a dialogue agent wants a dialogue to take is usually decided by a *dialogue manager*, but this desire has to be expressed in text or speech. This

is usually referred to as *natural language generation*, or simply NLG. In this project I generate text using a *language model*.

4.2 LANGUAGE MODELS

Language generation is part of wider field called *language processing*, which includes tasks like sentiment analysis, speech recognition or machine translation. An important part of language processing is *tokenization*. Tokenization is the act of separating a text into words, or *tokens*, by some criteria. A simple tokenization method for English is to separate words by white-space.

A language model is a probability distribution which assigns likelihoods to tokens in a sequence [5]. A common language model is

$$p(t_1, t_2, \dots, t_n) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1}) \quad (1)$$

where the joint probability of a sequence $T = t_1, t_2, \dots, t_n$ is modeled as a product of conditional probabilities. It assumes that the likelihood of an arbitrary token t_i depends only on the preceding tokens in the sequence, the *context*.

The issue with this model is that the conditional probabilities are practically impossible to estimate empirically because of the many different permutations of contexts that exist. Consider the number of viable contexts for the word “the” for example.

By imposing a limit on the context’s length, the conditional probabilities can be estimated through statistical methods. A simple language model that is possible to implement is a *bigram model* which estimates the probability of a word based only on the preceding word. A problem with this simple model however is that a words tend to have relations that span more than two words.

Language models can be used for text generation, among other tasks. If the conditional probabilities of enough words are known, then given a context the model can predict what word should come next.

Neural networks can be used to estimate language models, referred to as *neural language models*. A neural network is a network of linear and non-linear computation units, which take a vectorized input and produce some output. Neural networks are commonly layered, where the output of one computation layer is passed as input to the next. Neural networks with many layers are usually called *deep neural networks*, which in combination with large amounts of data defines a field known as *deep learning* [5]. A distinguishing aspect of deep learning compared to classical statistical methods is that very little to no manual processing is done to the data that is analyzed. For example, in image analysis, instead of manually converting an image to components such as edges

and colors, the different layers of the deep neural network may learn to do this separation by itself during training.

With the recent popularity of deep learning and neural networks as estimators of probability distributions, these have been applied to the task of language modeling. Through training on a set of text, a machine learning model tries to estimate the conditional probabilities of words in the set. Neural language models project a tokenized input sequence to a vector space, commonly referred to as a *word embedding* space. The final layer of the network projects the word embeddings back to token space, where they can be interpreted.

4.3 LOSS & VALIDATION METRICS

Neural language models are often trained as multi-class classifiers, where each token in a given vocabulary is treated as a class that has to be predicted by the model. Cross entropy is commonly used as a loss metric for such classifiers. Along with a loss function, there is also the need for a validation metric. The validation metric is used to automatically evaluate the performance of the model during and after training. For this purpose, I use a metric called perplexity and the BLEU score.

4.3.1 CROSS-ENTROPY AND PERPLEXITY

The *cross entropy*, or *relative entropy*, for two probability distributions is

$$H(p, q) = - \sum_x p(x) \log_2 q(x) \quad (2)$$

and is only defined for values of x where $p(x) = 0$ when $q(x) = 0$. Cross entropy is frequently used as a loss function for multi-class classification models, with q being the predictions of the model and p the probability distribution to be modeled [5]. In this context, x is a token in a sequence and $q(x)$ is the predicted probability of x occurring in the sequence.

Perplexity is a metric derived from cross-entropy, and is often presented as a measure of quality for language models [5].

Perplexity is calculated as

$$2^{H(p, q)} \quad (3)$$

where $H(p, q)$ is the cross-entropy function as defined in (2). Perplexity is usually normalized over the length of the output, and can be seen as a measure of how well a model predicts a data set [5], where a lower perplexity represents a higher confidence.

4.3.2 BLEU

BLEU is a metric originally proposed to automatically evaluate machine translations [6]. The BLEU score is a modified precision score that caps the amount of times a word is allowed in a translation before not contributing to the score anymore. This prevents translations that only consist of a single word repeated endlessly. A *hypothesis* produced by a model is compared to one or more *references*, reflecting the fact that a sentence may have multiple valid translations.

Instead of counting single words, the BLEU score can also be modified to count bigrams or trigrams, denoted as BLEU-2 and BLEU-3 respectively. Commonly, BLEU-1 up to BLEU-4 are used, and can be combined to a singular BLEU score by taking the geometric mean of the separate scores [6]. A high BLEU score indicates that the output of a model is similar to the reference(s), content-wise.

Outside the field of machine translation, BLEU has also been used as a evaluation metric for dialogue generation. However, the score has been shown to have a poor correlation with human evaluation scores [4, 7].

4.4 TRANSFORMER NETWORKS & ATTENTION

The *transformer* is a deep neural language model by Vaswani et al. [8], originally intended for machine translation. The distinguishing feature of transformer networks is that they mainly consist of *attention* layers. Attention layers allow models to “pay attention” to different parts of the input sequence. A typical attention function is

$$\text{softmax}(Q \cdot K^T) V \quad (4)$$

where Q , K and V are matrices of parameters learned during the training process. The result of the attention function is a distribution of weights indicating how the current token relates to other tokens in the input sequence. The distribution is then passed to a single layer neural network. In a transformer, multiple layers of attention functions and neural networks are repeated.

Because of the dot-product in the attention function, attention layers have a computational complexity of $\mathcal{O}(L^2)$, where L is the length of the input sequence. This can be compared to RNN:s where the complexity increases linearly ($\mathcal{O}(L)$)¹. Due to the exponential increase in computation time, Transformers tend to use relatively short input lengths, which limits the amount of contextual information. Since the original transformer there have been variations presented that try to alleviate this issue [9, 10]. Transformers have been shown to be effective in

¹Or $\mathcal{O}(LN)$ where N is the number of layers in the network.

a number of fields, and as such there have been many variants following the original [11].

I decided to use transformers in this project because they were present in many of the related works I studied, and consistently showed good performance. Another reason was the availability of large-scale pre-trained transformer models for general use.

4.5 GENERATIVE PRE-TRAINED TRANSFORMER (GPT)

GPT-2 is a pre-trained large-scale Transformer model presented by Radford et al. in 2019 [12]. It is an upscaled version of the Generative Pre-trained Transformer (GPT), in regards to its number of parameters and the amount of data used during training [13]. GPT-2 is an unidirectional language model; its attention layers only have information about previous tokens in a given sequence.

GPT-2 was trained on a 40GB dataset consisting of text scraped from various sources on the internet, called *WebText*. The training goal was to predict the next token of a given input sequence, with loss calculated using cross-entropy. When trained, GPT-2 can be used for text generation. Feeding the model with a sequence of text will make the model predict the upcoming word, which can then be appended to the original sequence. The extended sentence can then be fed back into the model to continue extending it until some stop condition.

One of the main arguments presented by Radford et al. [12] is that given a sufficiently large corpus of text, a predictive language model will implicitly learn many of the tasks traditionally taught by supervised learning, such as translation, question answering or sentiment analysis. The model can then be fine-tuned to perform other tasks. GPT-2 does not outperform more specialized models in many of the tasks that were tested, but performs well for language modeling with previously unseen data [12].

Although the overall task performance of GPT-2 was middling, its text generating capabilities brought a lot of media attention to the model [14] [15]. The trained instances of GPT-2 were initially withheld by OpenAI out of fear for misuse, like the production of fake news, but have since been released to the public [16].

GPT-2 is trained on a variety of internet text, and will try to produce a logical continuation to the context it is presented with. It will however usually continue the text in a literary style, since the majority of the text in the corpus is in that style. This can be seen in Figure 2.

4.6 DECODING METHODS

A language model like GPT-2 produces probabilities of tokens in a sequence. To get text, the probabilities have to be *decoded*. I used and compared three

decoding methods in this project: *Greedy decoding*, *top-k sampling* and *nucleus sampling*. When generating text, only the likelihood of the last token in the sequence is of interest, since it is the continuation of the input sequence. A simple method of decoding is to always pick the token with the highest likelihood, called *greedy* decoding.

The decoding process can drastically affect the overall performance of a language model [17, 11]. Holtzman et al. [17] argues that maximum likelihood methods have a tendency to cause repetitive and unnatural output, especially for open-domain dialogue agents. A possible explanation cited by Holtzman et al. [17] is that humans tend to avoid obvious statements when communicating, to not sound trite, making the process of open-domain response generation inherently more random.

An alternative to picking the most likely token is to sample tokens from the multinomial tokens distribution. However, this may lead to a low probability token being picked. A solution to this problem is to limit the number of candidates to the k most likely tokens. This is called top- k sampling [18]. Top- k sampling was used by Radford et al. [12] when demonstrating the text generation of GPT-2.

A potential problem presented by Holtzman et al. [17] with top- k sampling is that depending on the context, there will be different numbers of viable options. For example, when picking a response to the prompt “Hello” many continuations may be considered as likely. A small k may not encompass all viable alternatives in that situation, and diversity is lost. Conversely, in a situation where only one response is realistically viable, such as the answer to the question “What is the capital of Sweden?” a large k may produce the wrong answer because a low probability candidate was sampled instead of the correct response.

Instead of sampling for a fixed number of candidates Holtzman et al. [17] instead proposes to sample from the set of candidates with a cumulative probability of p , where p is chosen as a hyperparameter. This decoding method is called top- p sampling, or *nucleus* sampling. Top- p sampling leads to the set of candidates growing and shrinking dynamically, depending on the context. Li [11] observed that nucleus sampling produces better results than top- k in some, but not all, cases.

There are also decoding methods that use other machine learning models to rank candidates, as used by Zhang et al. [19] for their model DialoGPT. In opposition to such methods, Adiwardana et al. [20] argue that if a model has sufficiently low perplexity, such complex decoding methods are not necessary, citing a correlation between low perplexity scores and high scores in human evaluations.

4.7 RESPONSE GENERATION USING TRANSFORMERS

Response generation is a subset of text generation where generated text is supposed to be a response to the context, rather than a continuation. For example, when presented with the prompt “Do you like my cat?”, the model is expected to produce a response like “Yes, I like your cat!” rather than “I said to the man by the bus stop.”, which GPT-2 may produce.

Wolf et al. [21] used a GPT-2 like model which was fine-tuned on a conversation dataset. They also added another training goal, where the model has to predict the correct response from two choices, one real and one random response. This approach won them the automated metric category in the 2018 ConvAI 2 challenge, indicating that this may be a good approach for response generation.

Zandie and Mahoor [22] continues on the work of Wolf et al. [21] but adds another training task, emotion prediction. They use the *DailyDialog* conversation dataset, where each utterance is annotated with an emotion. They have the model try to predict the emotion of the next utterance. No human evaluation was done, but the model reached a better perplexity on the *DailyDialog* dataset than the baseline models they compared it with.

DialoGPT is a fine-tuned version of GPT-2 trained on a set of 147 million conversations scraped from Reddit [19]. Comment chains are concatenated into a dialogue and used as context for the model. The goal of the work was to train a model suited for open-domain response generation, with multiple turns of dialogue. When evaluating their model, Zhang et al. [19] tests a decoding scheme where a reverse language model ranks responses based on the likelihood of them being connected the context.

I decided to base my work around DialoGPT since it is a pre-trained transformer specialized for response generation. Without additional training, DialoGPT can be “talked” with, which means that it could out-of-the box be used to generate utterances for Neil. What prevents this, however, is that the Neil data has a scene description and affect associated with it, which breaks the expected format of the model and makes fine-tuning required. I decided to also use the base GPT-2 model to compare with the performance of DialoGPT, to compare the difference in performance the added training introduces.

A major question to consider is how to represent the relevant information to the model, which in this case are the scene description, the affect and the utterance. Zhou et al. [23] and Rashkin et al. [24] use dedicated sections of the models to encode the context and emotion, before passing it to another part responsible for producing the response. Wolf et al. [21] and Shirish Keskar et al. [25], which use more monolithic GPT-2 like models, instead focuses on the format of the input string, placing all information relevant to the model there and relying on its lingual abilities to understand what is what in the input. Wolf et al. [21] and Zandie and Mahoor [22] both use additional training goals to try guiding the

model into paying attention to aspects such as conversation flow and emotion. Since I decided to use GPT-2 like models, I too decided to add multi-task learning goals, with the goal of forcing the model to pay attention to the affect of the utterances.

In recent years, frameworks and libraries based on neural language models and transformers have appeared. HuggingFace maintains a Python library simply called “Transformers” which is a unified library of pre-trained transformer models, including GPT-2 [26]. The framework also provides a number of scripts intended to fine-tune models that are available in the library. ParlAI is a framework maintained by Facebook AI intended for the development of dialogue agents, and implements a lot of common functionality to train and evaluate agents [27]. It also provides a number of pre-trained models, including ones from HuggingFace’s Transformers.

4.8 THE CHOICE OF AN INTERMEDIARY DATASET

I was concerned that going from a very large and general dataset directly to the small and specialized Neil dataset would cause problems like overfitting. I thus decided to use an intermediary conversation dataset, and compare the difference in performance between fine-tuning directly on the Neil data and using this intermediary training step. This project deals with emotionally labeled responses. As such, I sought out data sets with conversations annotated with emotion labels. Rashkin et al. [24] produced a data set called *EmpatheticDialogues*. *EmpatheticDialogues* is an annotated dataset of 25k conversations. Each conversation in *EmpatheticDialogues* is associated with a description of a personal experience a crowd worker has had where they felt a given emotion. The worker then discussed their experiences with another worker for different numbers of dialog turns. The conversations are labeled with the emotion the worker was presented, for a total of 32 different emotion labels.

Another conversation data set with emotion annotations is *DailyDialog* by Li et al. [28]. *DailyDialog* consists of 13K conversations between two people, scraped from various websites for teaching how to speak English. An advantage to *DailyDialog* over *EmpatheticDialogues* is that each utterance is manually labeled with one of six emotions, compared to *EmpatheticDialogues* where the emotion label refers to the scene description rather than the utterances. I ultimately decided to use *EmpatheticDialogues* as my intermediary dataset since it has a textual scenario description, something *DailyDialog* lacks. Furthermore, *EmpatheticDialogues* consists of crowd sourced utterances, like the Neil data.

4.8.1 EMOTION REPRESENTATION

A challenge when it comes to emotion prediction is the choice of class labels. How should emotions be described? This is a question which partly relates

to psychology. Various attempts have been made to create a unified emotion classification, with varying degrees of granularity. The six emotions defined by Paul Ekman, happy, sad, angry, surprise fear and disgust often appear in corpora intended for sentiment analysis [29], including *DailyDialog*. *EmpatheticDialogues* classifies dialogues with 32 different emotion labels [24]. The Neil data uses a numeric scale which represents a range from frustration to excitement. The difference in labeling for emotions makes transfer learning for emotion predictions more difficult, as the number of labels may differ, or represent different things depending on the dataset.

In another approach, Felbo et al. [30] argues that the granular approach to classifying emotions leads to worse results for sentiment analysis, since manual labeling of emotions introduce bias. Their solution is instead to map textual input to a vector space representing emotions. This approach may be more friendly to transfer learning applications, since the numerical representations are easier to pass between models.

4.9 ETHICS OF NEURAL LANGUAGE MODELS

Like with many new technologies, generative language models have a number of ethical issues surrounding them that are worth discussing. Language models and word embeddings can be biased, depending on the data used to create them. They may reinforce outdated views that are present in corpora of text. Bolukbasi et al. [31] shows a number of gender based association and analogies that was produced when a word embedding model was trained on articles from Google News. The model associated “she” to occupations such as homemaker, nurse and receptionist, whereas “he” was associated to occupations like captain, architect and fighter pilot. Bolukbasi et al. [31] also points out that this is not an issue exclusive to data-driven language modeling, but many fields where machine learning models are trained on historical data.

DialoGPT was fine-tuned on a set of comments from Reddit. When training DialoGPT, Zhang et al. [19] filtered comments that contained a set of offensive words. Despite this effort, there is still a risk of the model producing responses that are offensive, biased or counter-factual. Although demographics vary between subreddits, the average Reddit user is from the United States and predominantly male with ages in the 20–30 range [32]. Biases associated with this particular demographic may thus be expected to arise in utterances produced by DialoGPT. The bias towards male produced content may also be present in the Neil data, since more men than women use Mechanical Turk [33] on average.

The tendency for transformers trained on web text to produce offensive output is also noted by Li [11] in his investigation.

As language models become better at replicating text, so does the fear that they will be used to impersonate humans. OpenAI decided not to release the largest

version of GPT-2 initially. Their fears were that the GPT-2 would be used in the production of fake news or spam. However, OpenAI has also been criticized for this decision, with arguments that it prevents security actors from properly assessing the threat and that openness is better than secrecy. There are also arguments that the concerns are overblown, with any lengthy text generated by GPT-2 being easy to spot as fake [34]. Although GPT-2 can produce output that appears fluent at a glance, the lack of understanding of context and common sense makes it unreliable. Furthermore, Marcus [34] writes that a system that is deliberately hidden from the public is easy to hype up as dangerous.

4.10 REFLECTIONS ON NEURAL LANGUAGE MODELS

Responses from neural response generators tend to be generic and uninteresting [3]. Being vague is also a way of avoiding contradictions, which is helpful to pass human evaluations. Traditional, modular and more manually designed dialogue systems may be more coherent than neural approaches, but are instead restricted to a limited field. A modular, manually crafted, system is more transparent may be easier to understand and adjust. Each component can be developed and evaluated separately, specialized for different tasks.

On the other side, organizations such as Google, Facebook and OpenAI often argue that excessive modeling is inherently bad, and instead use empirical models that are entirely based on the large amounts of data they have access to. Although the results they present are often impressive, these actors have been accused of chasing singular scores, such as perplexity or BLEU [35]. They then work on increasing the score by adding more parameters to the model or by using more data, even if the score may not actually be meaningful.

The large amounts of data and expensive hardware required to train large-scale models raise the barrier of entry for smaller actors, and makes reproducing the work more difficult. DialoGPT was trained with machines using 6 Nvidia V100 [19], which at a minimum gives about 96 GB of available GPU memory. In comparison, the Titan X I used has 12 GB of available memory. Pre-trained models alleviate this issue somewhat, but even if model weights or source code are made available, the work should ideally be reproducible from scratch.

To complicate things further, many of the recent developments in transformers are published in papers that are not peer reviewed. It sometimes seems like that presenting state of the art performance is good enough for others to continue on the work displayed. The original transformer paper by Google has 368 citations but is not published in a journal nor formally peer-reviewed. This can lead to the authors of these papers overestimating or exaggerating the capabilities, or novelty, of their models.

5 RELATED WORK

Li [11] performed an extensive study of different GPT-2 like transformer-based response generators. They note that the transformers can reach high scores in relevance and diversity, but also have a number of issues associated with them, such as tendency to drift away from the conversation topic and a risk of producing offensive responses.

Adiwardana et al. [20] showcases a neural response generator called “Meena”. The model is an “Evolved Transformer” and has 2.6B parameters, trained on a 341 GB dataset. Although the model is presented in the paper, the main focus lies on demonstrating the correlation between a low perplexity and the new “SSA” human evaluation measure presented.

Lin et al. [36] created a response generator using *EmpatheticDialogues* and a modified Transformer architecture that uses multiple parallel decoder layers. Each decoder layer was calibrated to respond with a certain emotion. An additional layer then chose the most fitting response based on the context. An interesting aspect of this approach is that the responses for each emotion can be studied. Although the design of the model is interesting, the sheer size of it makes it difficult to implement on the hardware I have available.

A set of methods not explored fully in this project are probabilistic, latent-space models, such as conditional variational autoencoders (CVAE). Results from Li [11] and Zhou and Wang [37] projects suggest that the random nature of probabilistic models is an advantage in open-domain dialogue settings, as they produce more varied responses than deterministic models. As these were encountered late in the research process, I did not have time to fully explore them. There is also less support for these types of models in existing frameworks, so more implementation work would have had to be done.

Apart from transformers, another commonly used type of neural language model are recurrent neural networks. Zhou et al. [23]’s “Emotional Chatting Machine” is a RNN-based dialogue agent that tries to incorporate the emotion of the context and produce an appropriate response.

6 DATASETS

I used two sets of data in this project: These are the utterances collected by crowd-workers for the robot Neil and the dataset *EmpatheticDialogues*. I will denote the training combination of *EmpatheticDialogues*, followed by Neil data as “ED→Neil”. As I am using pre-trained models, GPT-2 and DialoGPT, these are each associated to datasets they were trained on. These are the WebText dataset for GPT-2, and Reddit comment data for DialoGPT. As I am not interacting with these datasets directly, I will not go into the details of them.

6.1 EMPATHETIC DIALOGUES

Empathetic Dialogues is a dataset which consists of 24850 conversations that are connected to a personal experience [24]. Crowd workers were asked to think of a situation where they felt a given emotion. The worker then talks with another crowd worker about the situation for up to six dialog turns, see Figure 4 for examples. The data is split into training, validation and test sets that make up of 80%, 10% and 10% of the data respectively.

<p>Label: Afraid Situation: Speaker felt this when... "I've been hearing noises around the house at night" Conversation: Speaker: I've been hearing some strange noises around the house at night. Listener: oh no! That's scary! What do you think it is? Speaker: I don't know, that's what's making me anxious. Listener: I'm sorry to hear that. I wish I could help you figure it out</p>	<p>Label: Proud Situation: Speaker felt this when... "I finally got that promotion at work! I have tried so hard for so long to get it!" Conversation: Speaker: I finally got promoted today at work! Listener: Congrats! That's great! Speaker: Thank you! I've been trying to get it for a while now! Listener: That is quite an accomplishment and you should be proud!</p>
--	--

Figure 4: Examples of conversations from ED, from the original paper by Rashkin et al. [24].

There are 32 emotion labels in Empathetic Dialogues. These are: surprised, excited, angry, proud, sad, annoyed, grateful, lonely, afraid, terrified, guilty, impressed, disgusted, hopeful, confident, furious, anxious, anticipating, joyful, nostalgic, disappointed, prepared, jealous, content, devastated, embarrassed, caring, sentimental, trusting, ashamed, apprehensive and faithful.

The content of ED differs in certain ways to that of the Neil data. For example, the emotion label refers to the situation rather than the utterances themselves. The situation is also narrated in first-person, rather than the third-person format of the scenarios description in the Neil data. The situation description also refers to a prior situation, rather than one the two speakers are in currently. Despite these differences, I still decided to use ED as an intermediary dataset because of the alike data fields, these being a situation description, an emotion label and at least one utterance.

In my implementation, the emotion labels are only relevant to the multitasking model, which tries to predict an emotion label or rating based on the context and an utterance, see Section 7.3.2 for details. The emotions that are used to label the scenarios in ED differ from those in the Neil data, with ED using 32 different emotion categories and Neil a numeric scale ranging from excited to frustrated. There are emotions within the 32 that corresponds more or less to those in the Neil data; excited and joyful or annoyed and disappointed for example. One approach would be to only pre-train using conversations from ED that have corresponding emotions in the Neil data. I decided to include all emotion labels, even those with no connection to the Neil data, like "sentimental". My reasoning for this choice was a hypothesis that a variety of emotion labels would improve

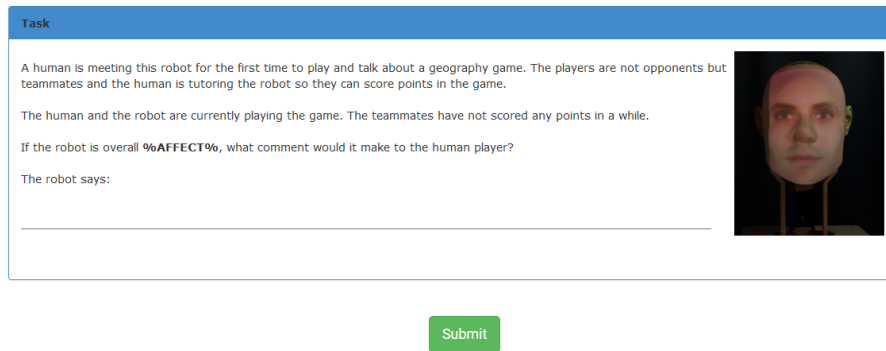
the model’s ability analyze sentiment in general, before specializing it on the Neil data.

6.2 NEIL DATA

The Neil dataset consists of utterances associated to different scenarios. Each scenario has a description of the situation the human and robot are currently in, and a direction of what the robot should say, see Figure 5 for an example scene. For each scene there are multiple intended *affects*. These are moods that the robot can be in at a given time. The affects are:

- Extremely excited and encouraging.
- Slightly excited and encouraging.
- Indifferent.
- Slightly impatient and provocative.
- Extremely impatient and provocative.

There are 61 unique scenarios, with 1512 utterances in total. The scenarios belong to one of three phases: *pre-game*, *ingame* and *post-game*. The utterances were collected in three stages. In the first stage, Turkers were asked to write a response based on the scene description and affect, like the one seen in Figure 5.



The screenshot shows a web interface for a task. At the top, there is a blue header with the word "Task" in white. Below the header, the text describes a scenario: "A human is meeting this robot for the first time to play and talk about a geography game. The players are not opponents but teammates and the human is tutoring the robot so they can score points in the game." It then states, "The human and the robot are currently playing the game. The teammates have not scored any points in a while." The task instruction is: "If the robot is overall **%AFFECT%**, what comment would it make to the human player?" Below this, it says "The robot says:" followed by a text input field. To the right of the text is a small image of a robot's head. At the bottom of the interface is a green "Submit" button.

Figure 5: A scenario description presented to Turkers. **%affect%** is replaced with one of the target affects.

In the second stage, responses are rated by other Turkers. The person is first asked to judge if the answer is nonsensical. If the utterance is not nonsensical, the following questions are asked (answers are denoted in square brackets):

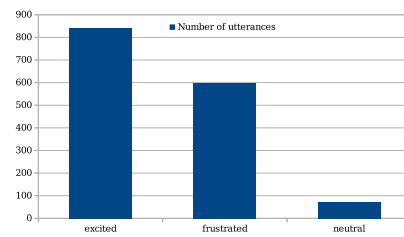
1. The response is ordinary and typical for the character’s given mood. [Strongly Disagree, Disagree, Neither, Agree, Strongly Agree]
2. How offensive is the utterance? [Not at all, Slightly, Moderately, Very, Extremely]

Answers are converted to a numerical scale. The typicality question is converted to a 0–5 scale with 0 being given if the answer is nonsensical. The offensiveness question is converted to a 1–5 scale.

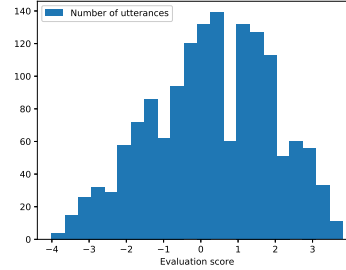
The third stage mainly focus on the emotion of the utterance. The person filling in the form is asked to answer *one* of the following questions, depending on the perceived affect:

- If the speaker of the line appears to be frustrated and provocative, how frustrated and provocative does the speaker appear to be? [Slightly, Moderately, Very, Extremely]
- If the speaker of the line appears to be excited and encouraging, how excited and encouraging does the speaker appear to be? [Slightly, Moderately, Very, Extremely]
- The speaker of the line appears neither excited and encouraging nor frustrated and provocative. The speaker seems rather indifferent. [Indifferent]

The answers are converted to a -4–4 scale with negative values indicating frustration, 0 indicating indifference and positive values indicating excitement. Example utterances, with evaluation scores, can be seen in Figure 7.



(a) Number of affects in Neil data, divided into three categories.



(b) Number of affects in the Neil data, grouped by evaluation score.

Figure 6: Two visualizations of the emotion distribution in the Neil dataset.

As can be seen in Figure 6, there are more utterances rated as excited than ones rated as frustrated. There are also a lot fewer utterances rated as indifferent/neutral compared to the other affects. One of the reasons for this is that

only utterances with an average evaluation score of exactly 0 are considered indifferent.

An utterance is rejected if the number of nonsensical ratings is ≥ 2 or the average score on the “typical and ordinary” question is < 2.6 . A line is also rejected if it is rated as > 2.6 for offensiveness. The reasoning for these criteria is that an utterance is rejected if it is scored as above average in offensiveness, and below average in typicality. I chose to include rejected lines in the data used for training, to preserve as much of the problem-specific data as possible. Even if an utterance has been rejected as offensive, it may still relate to the context which is information that the model theoretically can utilize.

The human and the robot are currently playing the game.
The teammates have not scored any points in a while. If the robot is overall [extremely excited and encouraging | extremely impatient and provocative], what comment would it make to the human player?

Excited: Keep up the good work! We can score! (4.6, 1.8, 2.8)
Frustrated: Please stop messing around and focus on the game! (3.4, 2.2, -1.4)

Figure 7: Two utterances for two variations of a scenario in the Neil dataset, one where the robot is excited and one where it is frustrated. The parenthesis after the utterances contain evaluation scores for typicality, offensiveness and emotion.

The Neil data consists of a single-turn conversation. There is only one turn of dialogue, one utterance, recorded per scenario. In the prompts originally presented to crowd workers, there were sometimes authored lines of dialogue that the robot and human might make in that situation, see Figure 13. One example of this is the robot asking the human if it has been to Sweden, and the human responding with either “Yes” or “No”. I chose not to include these authored lines in my training data, for different reasons. One reason is that these authored sentences are the same for all variations of the scenario, which may lead the model overfitting to that specific reply. Another reason is consistency, since only some of the scenarios have authored lines I thought it better to only use the crowd-sourced utterances, which all scenarios have.

7 IMPLEMENTATION

Three variations of transformer-based response generators were trained and evaluated: GPT-2, DialoGPT and DialoGPT with added training objectives. I call the additional training objectives *multiple choice* (MC) and *emotion classification* (EC), explained in more detail below. I will denote the multitasking model as “DialoGPT (MT)”. These three were in turn each trained with two levels of fine-tuning, with three being fine-tuned only with Neil data, and three with *EmpatheticDialogues* followed by Neil data. This leads to a total of six model variations, which are listed in Table 1.

All training and automatic evaluation was done in ParlAI. To use a model in ParlAI, it has to be implemented as an *Agent*, which conforms to an specific interface [27]. A GPT-2 agent was already implemented in the framework. A DialoGPT agent was not present, and was implemented by me based on the GPT-2 agent implementation. This was made easy by the fact that DialoGPT is architecturally the same model as GPT-2, with different pre-training. Functionality for multi-task learning was added by me onto the DialoGPT implementation. All model classes, configurations and pre-trained parameters were sourced from HuggingFace’s Transformer library. All models are “medium” sized models with 24 attention layers, which amounts to about 345 million trainable parameters. Both models have a dictionary of roughly 50000 tokens [26].

Table 1: Summary of trained models, with total training time over all data used.

Model	Data	Total Training Time (h)
GPT-2	Neil	1.2
DialoGPT	Neil	1.2
DialoGPT (MT)	Neil	1.0
GPT-2	ED → Neil	8.5
DialoGPT	ED → Neil	8.0
DialoGPT (MT)	ED → Neil	6.2

7.1 DATA MANAGEMENT

In ParlAI, a dataset is defined as a *task*. Each task has a *teacher* associated with it, which presents prompts to an agent and scores the responses it produces [27]. The Neil dataset thus had to be implemented as a task, which requires the data to be split into three sets. These are: A training set, a validation set and a test set. The number of utterances in each set can be seen in Table 2. Three scenarios, one from each phase, were excluded from the training and validation set and are only present in the test set. 67 utterances are associated to these scenarios.

Table 2: Proportions of Neil data used to train the models.

Set	Lines	% of Data
Train	867	57%
Validate	404	27%
Test	242	16%

During training on the Neil task, a situation as seen in Figure 5, is used as a prompt. An utterance associated to the combination of scenario and affect is used as a label, which the model tries to predict.

A teacher for *EmpatheticDialogues* was already implemented in ParlAI, with the only modifications required being those related to multi-task learning. When training on *EmpatheticDialogues* the model tries to predict the next utterance in the conversation, with the preceding conversation and scene description as context. For both the Neil and *EmpatheticDialogues* tasks, a new token is introduced in the dictionary to indicate the end of the context portion. This token is inserted between the context and label during tokenization.

7.2 TRAINING

The ParlAI training schedule consists of a period of training where the model is presented data from the training set, produces a prediction, a loss is calculated and the model parameters are adjusted based on the loss. The training period is cycled with validation rounds. During the validation rounds, the model is presented with the validation set and an average validation score is calculated, without the model weights being adjusted. The validation score is a chosen metric, which in this case was per-token accuracy. This cycle of training and validation repeats until the validation score stops increasing for a set number of validation rounds. When this occurs, training stops and the model is tested one last time on the test set. Training was done on a machine equipped with a Nvidia Titan X GPU, with 12GB of VRAM. A list of values for hyper-parameters can be seen in Appendix A.

Due to limitations in the implementation and GPU memory, a batch size of only one sample is used. This is in stark contrast to the original training procedures of the models, which use much larger batch sizes. Batching have been noted to have a stabilizing effect when training transformers [38]. To counteract the single batch issue, gradient accumulation is used instead during training. The calculated gradients for each training step are collected and summed for a number of steps before being sent to the optimization function. This should give same results as with batching, but with more training steps required since more forward passes have to be performed for each backwards pass.

As mentioned in Section 4.2, a transformer model produces a multidimensional word embedding. For GPT-2 and DialoGPT without multi-task training, the embedding is just used for language modeling. For the multitasking version of DialoGPT, the word embedding is passed to three final layers, also referred to as *heads*. These were the language modeling head, the multiple choice head and the emotion estimation head. Each head has a loss metric associated to it, and the loss from each head is summed before being passed to the optimizer.

7.2.1 LANGUAGE MODELING

The goal of the language model task for all models is to generate a reply based on the given context. The language modeling layer maps the word embedding of the Transformer to the token space. The output is a matrix of size (L, D) , where L is the length of the input and D is the dictionary size. For each position in the sequence the head produces a probability distribution over the roughly 50000 tokens in the dictionary. The language modeling loss is then calculated with the cross entropy function. As explained in Section 4.3.1, the language modeling task for a neural network is usually defined as a multinomial classification task, where the model tries to predict each token in the label utterance.

7.3 MULTI-TASK LEARNING GOALS

What follows are brief explanations of how the additional tasks for the multi-tasking version of DialoGPT were implemented.

7.3.1 NEXT-SENTENCE PREDICTION/MULTIPLE CHOICE PREDICTION

Following the works of Devlin et al. [39] and Wolf et al. [21], *next-sentence prediction*, or *multiple choice* (MC), is added as an additional learning objective for one of the models. Along with the true label, a random label utterance is picked from the dataset. The model is then presented with both labels and tasked with deciding which one of the two options is the actual response to the context.

The multiple choice classifier consists of a single linear layer, with dropout. The input of the classifier is the embedding of the last token in the input sequence. The head’s output is a binary choice of which one of the two input sentences is real, with 0 being the first and 1 the second. The index of the correct and fake sentence is shuffled for each training round, to prevent the model from always choosing one of them.

7.3.2 EMOTION CLASSIFICATION

In addition to multiple choice prediction, an emotion classification (EC) head was added to one model following the example of Zandie and Mahoor [22]. Because of the difference in emotion labels between *EmpatheticDialogues* and *Neil* the prediction head could not be preserved from one data set to the next. The layer is thus re-initialized with random weights models when switching data sets.

For *EmpatheticDialogues* the model classifies the input into one of the 32 emotions categories, with cross-entropy as loss metric. For the Neil data, a regression model that estimates the average evaluation score is used instead. The evaluation score is a decimal value in the range $[-4, 4]$. The loss metric for the regression model is the squared difference between the predicted and real scores.

8 EVALUATION METHODS

8.1 AUTOMATED METRICS

To get an estimate of how well the models predict the Neil data, the perplexities of the models on the test set were recorded. These scores can be seen in Table 3. To judge the emotion classification ability of the multi-task model, confusion matrices over the different emotion classes of the Neil data were produced. These can be seen in Figures 9 and 10.

I used the average BLEU scores on the test set as an automated alternative to human evaluations of the models. The scores were calculated separately for the seen and unseen scenarios in the test set, for the sake of measuring the model’s ability of generating utterances for new scenarios. For the seen scenarios, reference utterances were fetched from the whole Neil corpus. This means that utterances used as labels during training are included as references during this evaluation. For the unseen utterances, the reference utterances will also be unseen since those scenarios only exist in the test set. I chose to use multiple utterances as references since each distinct combination of scenario and affect can have multiple utterances associated to it. The BLEU scores can be seen in Tables 4 and 5.

8.2 HUMAN EVALUATION

The human evaluation was split into two phases. The first phase was intended to evaluate the performance of the three decoders. The second phase compares all the different model variations, using the decoder from the first phase. The reason for splitting the evaluation into two phases was that we did not have the resources to evaluate all possible combinations of models and decoders.

The evaluation consists of two forms, denoted as “stage 2” and “stage 3”². These are the same forms that were used to evaluate utterances written by Turkers, with one additional question added to stage 2 for phase 2. The original questions can be seen in Section 6.2. The added question was:

- The utterance moves the conversation forward. [Strongly Disagree, Disagree, Neither, Agree, Strongly Agree]

384 workers participated in the evaluation, 162 in phase 1 and 258 in phase 2. Each worker participated in average five times. Each time they participate, workers rate five utterances produced by one model variation for a combination of situation and affect. In the case of the greedy decoder, they only rate a single utterance.

²Stage 1 in the original system was the stage where Turkers were asked to write utterances.

The main measure of quality for this stage is the rejection rate of utterances. As a reminder, an utterance is rejected if the number of nonsensical ratings is ≥ 2 or the average score on the “typical and ordinary” question is < 2.6 . A line is also rejected if it is rated as > 2.6 on the offensiveness question.

A low rejection rate means that more of the generated utterances are considered usable by the chosen metrics. The rejection rates of the different decoders gathered in phase 1 can be seen in Table 6. The rejection rates are based off the scores seen in Table 7.

It is also desired that the models can produce utterances that are varied emotionally. This is what the emotion ratings from the stage 3 questions measure.

8.2.1 PHASE 1: COMPARING DECODERS.

Phase 1 was intended to study the performance of the three decoding methods. Six scenarios were used, with three emotion categories each (excited, neutral and frustrated). For each scenario and emotion combination I generated five utterances, with each of the three decoding methods. However, since greedy decoding always gives the same utterance for a given context, only one utterance per context was evaluated for that scheme. In summary, this approach meant that 90 utterances were evaluated for top- k and top- p sampling, and 30 utterances for greedy decoding. Three of the six scenarios were present in the training set, and three were previously unseen to the models. The results of this evaluation phase can be seen in Section 9.2.1.

Nucleus sampling and top- k each have one hyperparameter associated with them. For top- k sampling, k was set to 25. This choice was informed by a recommendation in the ParlAI documentation. The choice of $p = 0.9$ for nucleus sampling was made because that is the suggested value presented by Holtzman et al. [17], and the default value in ParlAI. DialoGPT (MT), trained on ED and Neil, was used to generate utterances for this phase. It was chosen because it had the lowest perplexity on the test set, if by a slight margin, as seen in Table 3.

8.2.2 PHASE 2: COMPARING MODELS

The goal of phase 2 was to compare how the utterances from the different models were perceived by humans. For phase 2, the decoder with the best performance from phase 1 was used to generate utterances for the six model variations, as listed in Section 7. The same scenarios from phase 1 were used to generate utterances, three seen and three unseen. Each model generated five utterances for each of the six scenarios, with the three main affects. Each model thus had 90 utterances evaluated, for a total of 540 utterances across all models.

Like for the different decoders in phase 1, I calculated the rejection rates and emotion scores for the different variations. The results of this evaluation phase can be seen in Section 9.2.1.

9 RESULTS

Some sample utterances produced by one of the models, along with evaluation scores, can be observed in Figure 8. The utterances are generated for one of the scenarios that was unseen to the model.

The human and the robot have finished playing the game and talked about the game for a little while. If the robot is **excited**, how would it say goodbye to the human player?

Nucleus: There’s always next time. (3.0, 1.2, 0.4)
Top-*k*: I am so glad you are enjoying it so far. (3.8, 1.0, 2.8)
Greedy: I’m glad we can play together again. (4, 1.4, -1,4)

Figure 8: Some responses to the given scenario, produced by DialoGPT (MT) (ED→Neil) with different decoders. The parenthesis after the utterances contain evaluation scores for typicality, offensiveness and emotion.

9.1 AUTOMATED METRICS

9.1.1 PERPLEXITY & TOKEN ACCURACY

The multitasking version of DialoGPT, trained on both ED and Neil data, reached the lowest perplexity on the test set. There is, however, little difference between the multitasking and not multitasking versions of DialoGPT, with both achieving the same token accuracy and similar perplexity. Based on the perplexity of the pre-trained multitasking model, it was chosen as the model to use for phase 1 of the human evaluation.

9.1.2 BLEU SCORES

For both seen and unseen scenarios, the sorted BLEU scores of the models are clearly stratified by the decoder used. Greedy decoding scores produces the highest BLEU scores and nucleus sampling the lowest. Tables 4 and 5 show the BLEU scores of the different models, with different decoding schemes. The higher scores for the greedy model indicate that it produces utterances that are

Table 3: Training metrics for the different models, tested on Neil Data.

Model	Data	Perplexity	Token	
			Acc.	Epochs
GPT-2	Neil	20.1	0.42	5.4
DialoGPT	Neil	13.05	0.42	6.0
DialoGPT (MT)	Neil	12.72	0.43	4.4
GPT-2	ED \rightarrow Neil	13.93	0.42	4.0
DialoGPT	ED \rightarrow Neil	11.94	0.44	4.0
DialoGPT (MT)	ED \rightarrow Neil	11.81	0.44	3.8

more similar to existing ones in the Neil data, compared to the ones by the sampling method. Conversely, models using nucleus sampling have the lowest scores which indicate that those responses are the least similar to the original ones. Methods using Top- k sampling fall in the middle, score-wise. Looking within the decoder categories, models only trained on the Neil data achieve higher scores most of the time. This may indicate that those models produce utterances that are more similar to the original utterances than those trained on both ED and Neil data.

Table 4: BLEU scores for test set (with unseen scenarios excluded), using different decoding methods. References are retrieved from *entire* Neil corpus. $k = 25$ for top- k sampling, $p = 0.9$ for nucleus sampling. Models are ordered by BLEU score in descending order.

Model	Data	Decoder	BLEU	Std. Dev.
DialoGPT (MT)	ED \rightarrow Neil	Greedy	0,683	0,108
DialoGPT	ED \rightarrow Neil	Greedy	0,679	0,114
DialoGPT (MT)	Neil	Greedy	0,676	0,130
GPT-2	ED \rightarrow Neil	Greedy	0,676	0,120
GPT-2	Neil	Greedy	0,674	0,129
DialoGPT	Neil	Greedy	0,668	0,141
GPT-2	Neil	Top- k	0,660	0,102
DialoGPT	Neil	Top- k	0,651	0,100
DialoGPT (MT)	Neil	Top- k	0,644	0,097
GPT-2	ED \rightarrow Neil	Top- k	0,637	0,094
DialoGPT	ED \rightarrow Neil	Top- k	0,630	0,093
DialoGPT (MT)	ED \rightarrow Neil	Top- k	0,626	0,092
GPT-2	Neil	Nucleus	0,616	0,089
DialoGPT	Neil	Nucleus	0,613	0,089
DialoGPT (MT)	Neil	Nucleus	0,609	0,087
GPT-2	ED \rightarrow Neil	Nucleus	0,602	0,086
DialoGPT	ED \rightarrow Neil	Nucleus	0,600	0,086
DialoGPT (MT)	ED \rightarrow Neil	Nucleus	0,597	0,085

Table 5: BLEU scores for unseen scenarios in test set, using different decoding methods. References are retrieved from test set. $k = 25$ for top- k sampling, $p = 0.9$ for nucleus sampling. Models are ordered by BLEU score in descending order.

Model	Data	Decoder	BLEU	Std. Dev.
GPT-2	Neil	Greedy	0,618	0,077
GPT-2	ED \rightarrow Neil	Greedy	0,565	0,074
DialoGPT	ED \rightarrow Neil	Greedy	0,550	0,092
DialoGPT (MT)	ED \rightarrow Neil	Greedy	0,549	0,098
DialoGPT	Neil	Greedy	0,545	0,073
GPT-2	Neil	Top- k	0,536	0,098
DialoGPT (MT)	Neil	Greedy	0,533	0,077
DialoGPT	Neil	Top- k	0,530	0,093
DialoGPT (MT)	Neil	Top- k	0,522	0,090
GPT-2	ED \rightarrow Neil	Top- k	0,519	0,086
DialoGPT	ED \rightarrow Neil	Top- k	0,516	0,084
DialoGPT (MT)	ED \rightarrow Neil	Top- k	0,511	0,086
GPT-2	Neil	Nucleus	0,503	0,083
DialoGPT	Neil	Nucleus	0,498	0,082
DialoGPT (MT)	Neil	Nucleus	0,494	0,076
GPT-2	ED \rightarrow Neil	Nucleus	0,490	0,079
DialoGPT	ED \rightarrow Neil	Nucleus	0,488	0,078
DialoGPT (MT)	ED \rightarrow Neil	Nucleus	0,486	0,077

9.1.3 CONFUSION MATRICES

For the matrix in Figure 9, three major emotion categories are considered. Predicted scores > 0.1 are considered as predictions for *excited*, < -0.1 *frustrated* and anything in between *neutral*.

For the confusion matrix in Figure 10 I instead use the full scale of evaluated affects. The model is considered correct if its predicted evaluation score falls within a distance of 0.5 to the actual score, essentially rounding its estimation to the nearest stage 3 evaluation option.

From both matrices it can be seen that the model never misclassified an excited utterance as frustrated, and vice versa. Most errors are neutral utterances that are falsely predicted as excited, which makes sense since neutral utterances are much rarer than excited ones. Looking at the more detailed matrix in Figure 10, it can be noted that the model tends to classify most utterances as moderately excited or frustrated, and rarely predicts anything above that. This again makes sense looking at the emotion distribution in Figure 6b, where fewer utterances are ranked as more than very excited or frustrated.

Figure 9: Confusion matrix for emotion prediction head of DialoGPT trained on ED→Neil, with recall. Score below 0.1 are considered as frustrated, scores above 0.1 are considered excited and the rest neutral.

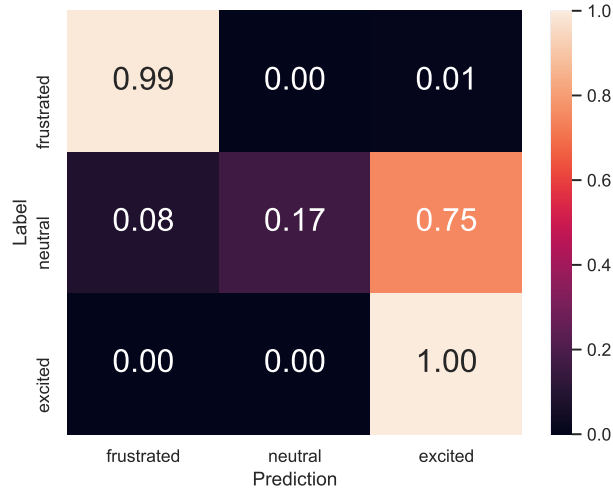
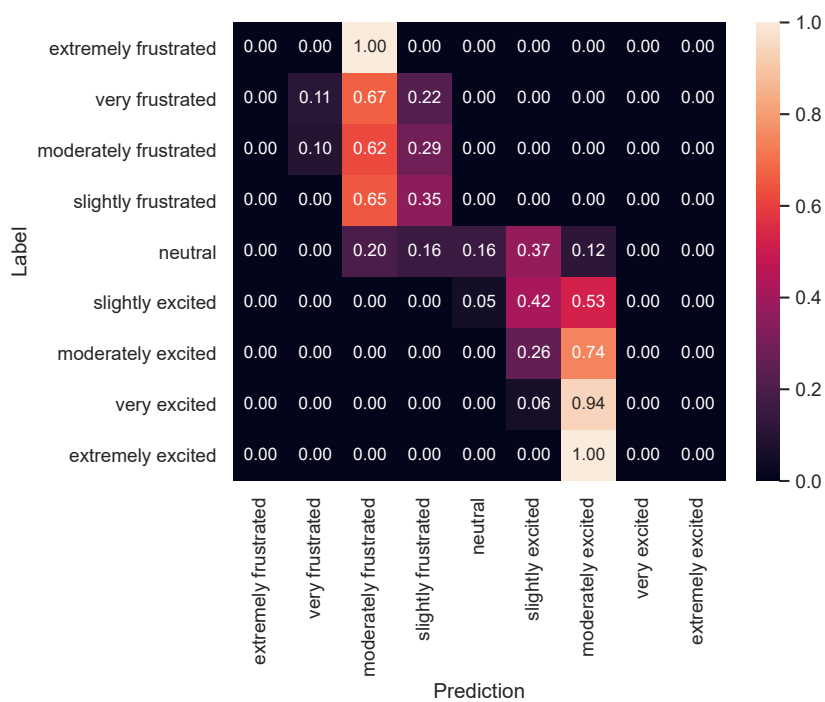


Figure 10: Confusion matrix for emotion prediction head of DialoGPT trained on ED→Neil, with recall. A correct prediction is considered as the distance to the predicted score being less than 0.5.



9.2 HUMAN EVALUATION RESULTS

9.2.1 PHASE 1 RESULTS

Top- k decoding has the lowest rejection rate for typicality, at 14%. This is the same percentage as for utterances produced by crowd workers. Nucleus sampling has the highest rejection rate, at 33%. On the other hand, it has the lowest rejection rate for offensiveness at 8%.

Table 6: Rejection rates for utterances produced with the different decoder, with human performance for comparison. Rejection criteria can be read in Section 6.2.

	Typicality Rejection Rate	Offensive Rejection Rate
Greedy	17%	10%
Top- k	14%	11%
Nucleus	33%	8%
Human	14%	11%

Table 7: Average ratings for offensiveness and typicality, averaged per utterance. Note that minimum offensiveness is 1.

Rating	Decoder	Mean	Std. Dev.
Offensive	Top- k	1,6	0,7
Offensive	Greedy	2,0	0,6
Offensive	Nucleus	1,9	0,6
Offensive	Human	1,9	0,7
Typical and Ordinary	Greedy	3,4	0,8
Typical and Ordinary	Top- k	3,4	0,7
Typical and Ordinary	Nucleus	3,0	0,8
Typical and Ordinary	Human	3,4	0,7

I also studied the emotion range of the different decoders, the results of which can be seen in Table 8. For the sake of the robot, a wide range of emotions is desired. Top- k sampling produced the widest range from the most excited utterance to the most frustrated. For this reason, and the low rejection typicality rejection rate, top- k sampling was used as the decoding method to generate utterances for phase 2.

Table 8: Evaluation scores for how emotional utterances produced with different decoders is perceived.

Decoder	Affect	Max.	Min.	Mean	Std. Dev.
Greedy	Excited	3.4	0.4	2.1	0.9
Greedy	Frustrated	-1.8	-0.4	-1.0	0.5
Nucleus	Excited	3.6	0.2	1.3	0.9
Nucleus	Frustrated	-3.4	-0.2	-1.1	0.9
Top- k	Excited	3.6	0.2	1.6	1.0
Top- k	Frustrated	-3.8	-0.2	-1.5	1.1
Human	Excited	3.8	0.2	1.5	0.9
Human	Frustrated	-4	-0.2	-1.4	0.9

9.2.2 PHASE 2 RESULTS

The average scores for typicality and offensiveness for the different model variations can be seen in Table 10. The scores are fairly similar across the different variations. The models are on average more offensive and less typical than humans, by a small margin. Two outliers are GPT-2 (ED→Neil) and DialoGPT (ED→Neil), which both have notably lower mean typicality scores than the other models. There are also the scores for the new question, which was not used in the original data collection procedure. There is not much difference between the models for this question, but here too DialoGPT (ED→Neil) scores low. On the other end, DialoGPT and GPT-2 trained on only Neil data appear to produce utterances that are the best at moving the conversation forward.

Table 9: Rejection rates for model variations. Using top- k sampling with $k = 25$.

Model	Data	Typicality Rejection Rate	Offensive Rejection Rate
DialoGPT (MT)	ED→Neil	0.21	0.43
DialoGPT (MT)	Neil	0.18	0.33
DialoGPT	ED→Neil	0.76	0.26
DialoGPT	Neil	0.21	0.57
GPT-2	ED→Neil	0.68	0.60
GPT-2	Neil	0.27	0.49

Table 10: Averages stage 2 scores for model variations. Using top- k sampling with $k = 25$.

Question	Model	Data	Mean	Std. Dev.
Offensive	DialoGPT (MT)	ED→Neil	2.5	0.6
Offensive	DialoGPT (MT)	Neil	2.4	0.6
Offensive	DialoGPT	ED→Neil	2.4	0.5
Offensive	DialoGPT	Neil	2.9	0.6
Offensive	GPT-2	ED→Neil	2.8	0.6
Offensive	GPT-2	Neil	2.6	0.7
Offensive	Human		1,9	0,7
Typical and Ordinary	DialoGPT (MT)	ED→Neil	3.2	0.6
Typical and Ordinary	DialoGPT (MT)	Neil	3.3	0.6
Typical and Ordinary	DialoGPT	ED→Neil	2.3	0.6
Typical and Ordinary	DialoGPT	Neil	3.2	0.6
Typical and Ordinary	GPT-2	ED→Neil	2.5	0.6
Typical and Ordinary	GPT-2	Neil	3.2	0.7
Typical and Ordinary	Human		3,4	0,7
Forward	DialoGPT (MT)	ED→Neil	3.6	0.5
Forward	DialoGPT (MT)	Neil	3.5	0.5
Forward	DialoGPT	ED→Neil	2.9	0.4
Forward	DialoGPT	Neil	3.7	0.5
Forward	GPT-2	ED→Neil	3.0	0.6
Forward	GPT-2	Neil	3.7	0.6

Looking at the emotion ratings from stage 3, in Table 11, the multitasking model seems to produce utterances with the widest range of affects. It almost reaches human levels of maximum excitement and frustration, like in phase 1.

The rejection rates for the different models can be seen in Table 9. Worth noting is that the rejection rates for the model that was used in both phase 1 and 2 (DialoGPT (MT), trained with ED→Neil) are nearly double those from the first phase, although the same utterances having been rated. This may be a

Table 11: Stage 3 scores for different model variations. Using top- k sampling with $k = 25$.

Model	Data	Affect	Max.	Min.	Mean	Std. Dev.
DialoGPT (MT)	ED→Neil	Excited	3.6	0.2	1.6	1.0
DialoGPT (MT)	ED→Neil	Frustrated	-3.8	-0.2	-1.5	1.1
DialoGPT (MT)	Neil	Excited	3.6	0.2	1.2	0.9
DialoGPT (MT)	Neil	Frustrated	-3	-0.2	-1.0	0.7
DialoGPT	ED→Neil	Excited	2.8	0.2	0.9	0.6
DialoGPT	ED→Neil	Frustrated	-3.4	-0.2	-1.1	0.8
DialoGPT	Neil	Excited	2.6	0.2	1.0	0.7
DialoGPT	Neil	Frustrated	-2.8	-0.2	-1.2	0.7
GPT-2	ED→Neil	Excited	2.2	0.2	0.8	0.4
GPT-2	ED→Neil	Frustrated	-3.2	-0.2	-1.1	0.7
GPT-2	Neil	Excited	3	0.2	1.3	0.8
GPT-2	Neil	Frustrated	-3.2	-0.2	-1.2	0.9
Human		Excited	3.8	0.2	1.5	0.9
Human		Frustrated	-4	-0.2	-1.4	0.9

consequence of different turkers participating in the different phases, and being more or less critical.

A table of rejection rates split into percentages for seen and unseen scenarios can be seen in Table 12, with a bar graph version in Figure 11. For utterances produced from previously seen scenarios the average rejection rates for typicality and offensiveness are 42% and 47% respectively. For utterances produced from unseen scenarios the average rejection rates are 42% for offensiveness and 35% for typicality.

If the two rejection criteria are combined, a considerable number of utterances are rejected, as shown in Figure 12. For GPT-2 (ED→Neil) about 90% of all produced utterances are rejected for either being too offensive or not typical enough. The model with the least percentage of rejected utterances is the multitasking version of DialoGPT trained only on Neil data, at 46% for both seen and unseen scenarios. For comparison, utterances produced by workers had a combined rejection rate of 24%.

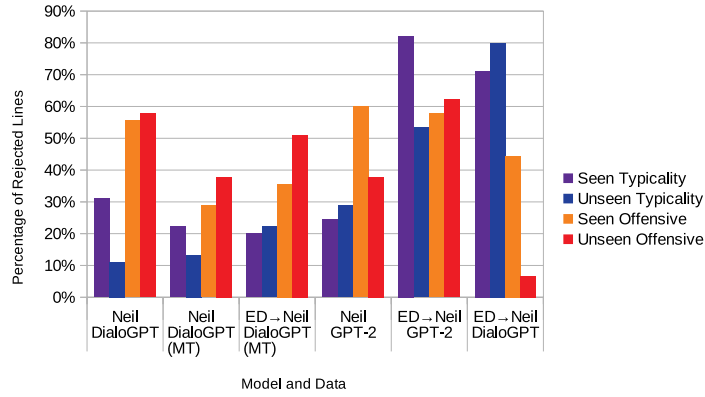


Figure 11: Rejection rates for the six models, separated for seen and unseen scenarios.

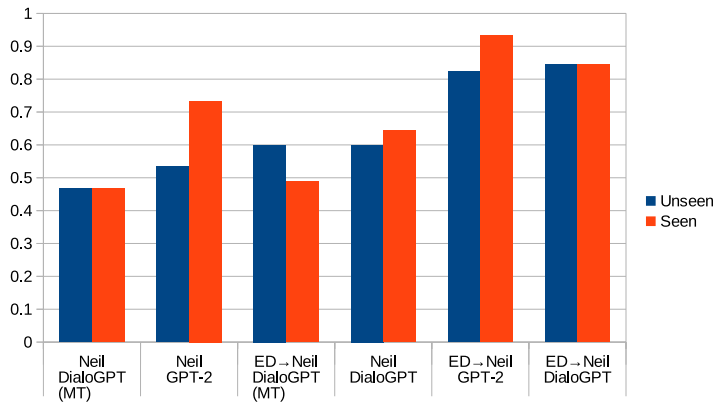


Figure 12: Combined rejection rates for the six model variations, separated for seen and unseen scenarios. A line is rejected if is too offensive, not typical enough or both. For comparison, the combined rejection rate for the human utterances was 24%.

10 DISCUSSION

In this project, I trained six large-scale language models to produce and expand utterances from the Neil data. Utterances produced by the models were evaluated using a combination of automated metric and human evaluation.

One of the desired outcomes for this work was the ability to produce utterances for unseen scenarios. From both the automatically calculated BLEU scores and the human evaluations, it can be observed that the performance of the models does not greatly decrease when presented with an unseen scenario. This is a promising result, as it may allow for utterances to be authored for new scenarios, without human involvement.

10.1 HOW DO THE MODELS COMPARE TO ONE ANOTHER?

I trained and tested six model variations, with variations in the amount of pre-training used as well as the use of multitasking during training. I also tested three different decoder methods. Due to resource limitations I could not test all permutations of parameters. I first decided on a decoder, and then tested the six variations using this decoder.

The models pre-trained with more data perform better at automated tests. DialoGPT gets higher scores than GPT-2, and models trained on both ED and Neil data get higher scores than those only trained on Neil data. There is a slight decrease in perplexity for the model with the added training goals, but it is not clear if this is a significant improvement.

In the human evaluation, a conclusion is harder to reach. When comparing the different models using the same decoder, it is difficult to find a model that is clearly better, in terms of typicality and offensiveness. The scores of the models are clustered close to each other.

This could either be taken as an indication that the pre-training does not affect the average perception of the utterances, or that the evaluation itself is flawed in some way. Working under the assumption that the scores are correct, one could speculate that the relatively small amounts of data I use for fine-tuning are not enough to significantly alter the performance of the large-scale models. This is where a non-finetuned version of either DialoGPT or GPT-2 would have been an interesting baseline, to see how the zero-shot performance differs from that of the fine-tuned models. Unfortunately, the change in syntax for the Neil dataset makes this impossible as the model will produce nonsense without any fine-tuning, often repeating single tokens endlessly. It is possible that most of the time during fine-tuning is spent adapting to a new input format. This likely affects GPT-2 more than DialoGPT, as it not only has to adapt to a response generation task but also the intricacies of the Neil data.

The low typicality and BLEU scores for nucleus sampling is something that sticks out when comparing the decoders. One reason for the low score may be that the output is more random than that of top- k sampling or greedy decoding. This relates to the choice of parameters for the sampling decoders. The number of tokens nucleus sampling draws from changes dynamically. The decoder could theoretically sample from hundreds of tokens on average, whereas top- k sampling with $k = 25$ always only draws from 25 tokens. For a more fair comparison between the methods, the average number of utterances produced for a given p could have been calculated, and used as the basis for the choice of k .

Choosing a decoding method is a trade-off between relevancy and variance. The more varied output tends to be more interesting, but not always relevant to the context. In task-oriented settings, deterministic greedy methods are likely the better option, whereas for social chat tasks randomness tends to be more welcome. The context in this thesis is a mix of the two, where the robot is for the most part social chatting in an open domain setting, but is also expected to follow the given direction for the mood. Nucleus sampling may be receiving lower scores for typicality on average because the utterances lose the connection to the prompt, from the high variety in the token sampling.

Comparing the different model variations, the scores for the models are separated more by the decoder used, rather than the amount of pre-training done. This can be observed both in the BLEU scores and human ratings. This can be taken as an indication that the choice of decoding method can change the performance of an language model significantly.

Which of the models is thus best suited for generating utterances for Neil? Based on the phase 2 scores alone, this is hard to decide as there is not variation with significantly higher scores for typicality. Looking at the overall rejection rates however, vast majority of certain models' utterances are rejected. From this metric it seems like the multitasking model produces the most usable utterances.

The emotion content is an important aspect of the language generation task. The robot should be able to produce emotions of varying levels. By that metric too the multitasking model seems like best choice, as it appears to produce utterances with the largest spread of emotions.

10.2 HOW WELL DO THE MODELS FIT THE DATA?

Out of the six models. GPT-2 trained on only Neil data is the only one that produced special tokens in its output. Special tokens are tokens used during training to indicate things like padding or the end of the sequence. These should ideally not be present in the final output, as they are not actually present in the training data. The lesser quality of GPT-2 (Neil) is better reflected in the automated training metric, see Table 3, where it has a significantly higher

perplexity than the other models. On the other hand, it has some of the better scores in the human evaluation, reaching high scores in

The evaluation scores of the models rarely go above the human baseline. One might argue that the model fits the data a bit *too* well, as it is reproducing offensive and nonsensical utterances at about the same rate as the Turkers. I did not implement a penalty if a model produces utterances that are present in the training set, so there is a risk of a model only producing existing utterances. This is a greater risk when using a greedy decoder, as the sampling decoders introduce an element of randomness that produces more varied responses.

An issue with the Neil data is that scenes are written in different ways, which likely make it harder for the models to find consistent patterns in them. Some scenario descriptions are more direct, for example “How does the robot say goodbye to the human player?”. Other scenes are more abstract and require higher levels of reading comprehension for the task to be clear. An example of such a prompt is the one seen in Figure 13, which can be summarized as “The robot is curious of the human’s profession, how would it react to the human player describing his/her job?”. This scenario requires an assumption that the human has responded to the implied question the robot has made. Many of the models responded with variations of “What is your profession?”, failing to recognize the latter part of the prompt.

The human and the robot will begin playing the game soon. The robot wants to get to know the human player better. The robot would like to know what the profession of the human player is.

Robot: What is your profession?
The Human player describes his/her job.

If the robot is overall **%AFFECT%**, how would it react to the human player describing his/her job?

The robot says:

Figure 13: An example to a scenario with authored lines.

10.3 HOW ARE THE MODELS PERCEIVED BY HUMANS?

While fitting the data is the goal of the machine learning process, the larger task I wanted to achieve was to produce utterances that were of human quality. This could only be evaluated by humans, as automated metrics mostly inform us how good the models are at reproducing the data it was trained on [3].

Based on the results from phase 1, the utterances are perceived as nearly equal to those produced by humans. From phase 2, all models perform slightly worse than humans, but by a small margin. The answers from the added question in phase 2 indicate that the workers in general think that the utterances move the conversation forward.

It is worth comparing the human ratings to the BLEU score, since the BLEU score is partly intended to reduce the need for human evaluation. Although the greedy decoder consistently had the highest BLEU scores, it did not get the highest scores for typicality in the human evaluation. Nucleus sampling scores poorly in both BLEU scores and typicality rejection rate, indicating a correlation between the two scores. Models trained on both ED and Neil consistently got lower BLEU scores, which is a consistency not as evident in the human evaluations. The BLEU score was averaged over more data than the utterances in the human evaluation, so is possible that the changes resulting from different pre-training level would be more clear with more evaluations. It is also possible that this reflects a difference in how the BLEU score works and how a human rates the utterances. The BLEU score only looks at the similarity between the preoduced utterances and a list of references, whereas a human can consider an utterance as typical even if it has no relation to the original data.

10.4 REFLECTIONS ON THE IMPLEMENTATION

I chose to implement this project in ParlAI. ParlAI already had GPT-2 support, which made adding DialoGPT to the framework easier. Adding multi-task learning was a greater undertaking however, since this task did not comply to the existing agent classes in ParlAI. The alternative to using a dialogue agent framework would have been to use a less specialized machine learning library, like pure PyTorch or Keras. With a system designed by oneself, there is a greater understanding of the components, but more time is required for implementation. It also restricts the ability for others to independently reproduce the results without the source code.

The choice of validation metric is an important choice when training a machine learning model in ParlAI. The validation metric is what dictates how long the models train on the data, and serves as a quick comparison metric between models. Accuracy is the default validation metric in ParlAI, which means that the model has to generate responses that exactly matches the label. In this context, this may not be desired behavior as we would like the model to be able to produce novel utterances. I instead chose to use token accuracy as my validation metric, relaxed the accuracy demands slightly but still requires the model to accurately predict at least parts of the original utterance to score. BLEU is a viable alternative to using token accuracy, as it is intended to look at content similarity rather than specific tokens.

I added multitasking in an effort to increase the performance of the model. The results indicate that the benefits to this are slight, but noticeable. The multitasking model reaches the highest scores in the emotion rating questions. I am hesitant to attribute this fact to the emotion prediction it does during training, as the difference between the models multitasking and not multitasking is not very large. It may be noted that for the multitasking model, the loss for

the multiple choice task decreased much more rapidly than the other tasks, which means that this task does not contribute much to the optimization process for most of the training. I did not compare the difference in performance between training with multiple choice prediction and not, but the rapid decrease in loss indicates that the performance difference would be small. Another, more subtle, advantage to the multitasking model is that it converged two hours faster than the non-multitasking model during training. The increased training speed of the model, while still receiving similar evaluation scores, brings some credence to the argument that multitasking increases sample efficiency.

10.5 REFLECTIONS ON HUMAN EVALUATION USING CROWD WORKERS

We use crowd-sourced workers to evaluate how the utterances of the models are perceived by humans. This relies on the assumption that workers will answer the form truthfully, which unfortunately is not always the case. We had problems with some workers that would always pick the same option for all questions they answered. The questions are largely subjective, and it is thus difficult to automatically filter workers with control questions, or disregard them for being “incorrect”.

My informal metric used to filter out bad workers was to see if they had participated more than five times, and also answered in an obvious pattern across different models, affect or decoders. This is obviously an imperfect and biased method, as it relies on one or two humans to try to decide if answers are valid or not.

Apart from the answers being subjective, another component to the difficulty in automatically finding and filtering bad workers is that there are large number of dimensions to consider. Workers rank utterances for different models, scenarios and affects. For each utterance, there are five evaluations. From this limited data it is hard to find outliers. Limiting the number of times an worker can participate could be a way of regularizing the evaluation data somewhat.

As mentioned before, workers are allowed to participate multiple times, which means that a single worker could bias the evaluation of a certain model. If the worker is consistently more critical in their evaluations, and mostly ranks utterances produced by single model, this would lead to a worse overall score for that model. On the other hand, a single good worker evaluating different models may lead to more consistent evaluations, as they may recognize differences in the utterances that workers only participating once would not.

It is also worth reflecting on the evaluation form itself. In this project I used the same form that was used to evaluate the original utterances written by humans. This was done to be able to fairly compare the performance of the models to that of humans.

A question to consider is whether the questions as stated measure what is intended to measure. There is a neutral option on both the typicality and “forward” question. This may lead workers who are unsure about their opinion to pick the middle option, rather than labeling the utterance as nonsense in some cases. Using an even-numbered scale with no middle option could be a way to try forcing people into an opinion, to maybe get well-separated result.

Automatic metrics are imperfect in some ways, but they are reliable. With this type of evaluation, the scores may change depending on who participates in the evaluation. There are more automated metrics than the ones I used. Utterances could be spell-checked for example, and the model penalized for producing utterances that fail. For an example of inconsistency inherent to human evaluation, the scores for the multitasking model between phases can be compared. The same utterances were used for both phases, but the mean scores for typicality and offensiveness differ between them.

Ultimately, this project is mainly about the implementation of generative language models. I also did not want to stray too far from the original evaluation process, to be able to make a fair comparison of the quality. It was also out of the thesis’ scope to research methods to detect bad workers. The problem of how to evaluate these types of models is not unique to this problem however, but is a hotly debated topic, with many suggested solutions. Adiwardana et al. [20] for example proposed a method where only two questions are asked, how sensible the utterance is and how specific it is. Those question by themselves may be a bit generic in this context, where for example a specific emotion is desired at times.

The inconsistency when it comes to crowd-sourced production and evaluation is a good argument for the work presented in this report. Although not perfect, the multitasking model can evaluate utterances very well into the major emotion categories used, opening the possibility of removing this evaluation step from the human evaluation. The multitasking model is training an emotion predictor alongside response generation. From the confusion matrices in the results section, it can be observed that the model is very good at classifying utterances into the two emotion categories, excited and frustrated. The accuracy decreases somewhat when using a more refined scale, but the results are consistent with the data.

10.6 FUTURE WORK

I only compared transformers in this project. It would be interesting to see how a smaller sized model performs compared to the transformer models. It would also have been interesting to train a RNN or transformer from scratch on the Neil data and comparing its performance to that of a pre-trained model. The scope of this project meant that there was not enough time to make either of these comparisons.

On the topic of decoders, I did not have time to test decoding methods that generate multiple answers and rank them. This approach can also be combined with other machine learning models, which rank the answers based on some criteria it is taught. As the evaluation scores differ more between decoders than levels of pre-training, I believe this is an area worth investigating.

It would be interesting to re-run the experiment, but without the rejected lines. The models currently do not reach above human performance, possibly because they are learning it is ok to be offensive or nonsensical. Would only training on “good” utterances lead to the model producing utterances that are better than the average human? Rather than just removing the rejected utterances, an interesting approach would be to have the model rate utterances for topicality and offensiveness during training, as multi-task learning goals. The information about offensiveness and typicality could also be added as part of the input, similar to the way the affect is currently included. With this approach, all the available data is utilized and the model is explicitly taught what is considered a highly rated utterance and what is not.

11 CONCLUSION

We see from the human evaluations that a transformer can produce utterances that are evaluated to be of similar quality to those produced by humans. How similar in quality the utterances are depends on what metric is observed. In terms of evaluation scores, the models are close to human performance. Looking at the percentage of rejected lines, the best model has nearly 50% of it’s lines rejected, which is about twice that of the crowd workers.

Since the model’s training objective is to imitate the existing utterances, they will also reproduce offensive and irrelevant utterances, which is likely one of the reasons none of the models seem to perform significantly better than humans. The similarity of certain evaluation scores could be interpreted in two ways. Either the performance variations of the different model variations are negligible, or the evaluation method used does not capture the differences that do exist. Most likely, the answer lies somewhere in-between. Based on its ability to produce the most emotionally varied results and the least rejected utterances, I believe the multitasking model, perhaps without the multiple choice task and with an typicality estimator, would be the best choice for future work on this topic.

The ability of transformers to reproduce text is impressive. It is worth remembering, however, they are much like the models that can produce images of cats. On a surface level the output appears genuine, but a closer look may reveal that the paws merge into the tail of the cat. The model does not have a concept of feline physiology, merely data about what the common visual features of cats are. Neural language model are not that different. They are able to produce text that

appear fluent and syntactically correct, but often repeats and contradicts itself, or veers from the intended subject. The models are only trained to predict what is the most likely utterance for a given scenario, with reading comprehension and grammar being implicit byproducts.

Despite the problems, it is once again worth noting how impressive it is that machine learning models created almost purely from data are seemingly able to write responses as well as a random human. I believe that generative models may be the most useful as parts of a larger system, which includes components that try to address some of the problems inherent to these methods.

The task of this thesis was to automate the production and evaluation of utterances for a robot in a map-game context, replacing the humans used in the process. The results indicate that humans may still be able to produce better utterances in average, but the algorithmically generated utterances are not far behind. The models also have the advantage of being able to produce utterances for many more new scenarios than would be economical to have humans author. The general quality of text generated from language models have only gotten better in recent years, and will only keep improving. Eventually, the robot Neil may truly be able to speak freely, regardless of the situation it is in.

12 ACKNOWLEDGMENTS

- Maïke Paetzel; for providing resources, support and guidance throughout this project.
- Ramesh Manuvinakurike, for providing research assistance.
- KDDI Research and the intelligent systems group, for housing and supervising me during my initial period in Japan.

REFERENCES

- [1] Iolanda Leite et al. “Semi-Situated Learning of Verbal and Nonverbal Content for Repeated Human-Robot Interaction”. In: *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. ICMI ’16. Tokyo, Japan: Association for Computing Machinery, 2016, pp. 13–20. ISBN: 9781450345569. DOI: 10.1145/2993148.2993190. URL: <https://doi.org/10.1145/2993148.2993190>.
- [2] Maïke Paetzel et al. “Incremental Acquisition and Reuse of Multimodal Affective Behaviors in a Conversational Agent”. In: Dec. 2018, pp. 92–100. ISBN: 978-1-4503-5953-5. DOI: 10.1145/3284432.3284469.

- [3] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. “Challenges in Building Intelligent Open-domain Dialog Systems”. In: *arXiv e-prints*, arXiv:1905.05709 (May 2019), arXiv:1905.05709. arXiv: 1905.05709 [cs.CL].
- [4] Chia-Wei Liu et al. “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2122–2132. DOI: 10.18653/v1/D16-1230. URL: <https://www.aclweb.org/anthology/D16-1230>.
- [5] Dan Jurafsky and James H. Martin. *Speech and language processing*. English. 3rd ed. Harlow: Pearson Education, 2019.
- [6] Boxing Chen and Colin Cherry. “A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU”. In: *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 362–367. DOI: 10.3115/v1/W14-3346. URL: <https://www.aclweb.org/anthology/W14-3346>.
- [7] Ryan Lowe et al. “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)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1116–1126. DOI: 10.18653/v1/P17-1103. URL: <https://www.aclweb.org/anthology/P17-1103>.
- [8] Ashish Vaswani et al. “Attention Is All You Need”. In: *arXiv e-prints*, arXiv:1706.03762 (June 2017), arXiv:1706.03762. arXiv: 1706.03762 [cs.CL].
- [9] Zihang Dai et al. “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context”. In: *arXiv e-prints*, arXiv:1901.02860 (Jan. 2019), arXiv:1901.02860. arXiv: 1901.02860 [cs.LG].
- [10] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. “Reformer: The Efficient Transformer”. In: *arXiv e-prints*, arXiv:2001.04451 (Jan. 2020), arXiv:2001.04451. arXiv: 2001.04451 [cs.LG].
- [11] Piji Li. “An Empirical Investigation of Pre-Trained Transformer Language Models for Open-Domain Dialogue Generation”. In: *arXiv e-prints*, arXiv:2003.04195 (Mar. 2020), arXiv:2003.04195. arXiv: 2003.04195 [cs.CL].
- [12] Alec Radford et al. *Language models are unsupervised multitask learners*. Technical Report. OpenAI, 2019.
- [13] Alec Radford et al. *Improving Language Understanding by Generative Pre-Training*. Technical Report. OpenAI, 2018.

- [14] Oscar Schwartz. “Could ‘fake text’ be the next global political threat?” en-GB. In: *The Guardian* (July 2019). ISSN: 0261-3077. URL: <https://www.theguardian.com/technology/2019/jul/04/ai-fake-text-gpt-2-concerns-false-information> (visited on 05/28/2020).
- [15] James Vincent. *OpenAI’s new multitasking AI writes, translates, and slanders*. en. Feb. 2019. URL: <https://www.theverge.com/2019/2/14/18224704/ai-machine-learning-language-models-read-write-openai-gpt2> (visited on 05/28/2020).
- [16] *GPT-2: 1.5B Release*. en. Nov. 2019. URL: <https://openai.com/blog/gpt-2-1-5b-release/> (visited on 05/28/2020).
- [17] Ari Holtzman et al. “The Curious Case of Neural Text Degeneration”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=rygGQyrFvH>.
- [18] Angela Fan, Mike Lewis, and Yann Dauphin. “Hierarchical Neural Story Generation”. In: *arXiv e-prints*, arXiv:1805.04833 (May 2018), arXiv:1805.04833. arXiv: 1805.04833 [cs.CL].
- [19] Yizhe Zhang et al. “DialogPT: Large-Scale Generative Pre-training for Conversational Response Generation”. In: *arXiv e-prints*, arXiv:1911.00536 (Nov. 2019), arXiv:1911.00536. arXiv: 1911.00536 [cs.CL].
- [20] Daniel Adiwardana et al. “Towards a Human-like Open-Domain Chatbot”. In: *arXiv e-prints*, arXiv:2001.09977 (Jan. 2020), arXiv:2001.09977. arXiv: 2001.09977 [cs.CL].
- [21] Thomas Wolf et al. “TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents”. In: *arXiv e-prints*, arXiv:1901.08149 (Jan. 2019), arXiv:1901.08149. arXiv: 1901.08149 [cs.CL].
- [22] Rohola Zandie and Mohammad H. Mahoor. “EmpTransfo: A Multi-head Transformer Architecture for Creating Empathetic Dialog Systems”. In: *arXiv e-prints*, arXiv:2003.02958 (Mar. 2020), arXiv:2003.02958. arXiv: 2003.02958 [cs.CL].
- [23] Hao Zhou et al. “Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory”. In: *arXiv e-prints*, arXiv:1704.01074 (Apr. 2017), arXiv:1704.01074. arXiv: 1704.01074 [cs.CL].
- [24] Hannah Rashkin et al. “Towards Empathetic Open-domain Conversation Models: a New Benchmark and Dataset”. In: *arXiv e-prints*, arXiv:1811.00207 (Oct. 2018), arXiv:1811.00207. arXiv: 1811.00207 [cs.CL].
- [25] Nitish Shirish Keskar et al. “CTRL: A Conditional Transformer Language Model for Controllable Generation”. In: *arXiv e-prints* (Sept. 2019), arXiv:1909.05858.

- [26] Thomas Wolf et al. “HuggingFace’s Transformers: State-of-the-art Natural Language Processing”. In: *ArXiv abs/1910.03771* (2019).
- [27] Alexander H. Miller et al. “ParlAI: A Dialog Research Software Platform”. In: *arXiv e-prints*, arXiv:1705.06476 (May 2017), arXiv:1705.06476. arXiv: 1705.06476 [cs.CL].
- [28] Yanran Li et al. “DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset”. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 986–995. URL: <https://www.aclweb.org/anthology/I17-1099>.
- [29] Laura-Ana-Maria Bostan and Roman Klinger. “An Analysis of Annotated Corpora for Emotion Classification in Text”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2104–2119. URL: <https://www.aclweb.org/anthology/C18-1179>.
- [30] Bjarke Felbo et al. “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1615–1625. DOI: 10.18653/v1/D17-1169. URL: <https://www.aclweb.org/anthology/D17-1169>.
- [31] Tolga Bolukbasi et al. “Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings”. In: *arXiv e-prints*, arXiv:1607.06520 (July 2016), arXiv:1607.06520. arXiv: 1607.06520 [cs.CL].
- [32] Mike Thelwall and Emma Stuart. “She’s Reddit: A source of statistically significant gendered interest information?”. In: *Information Processing & Management* 56.4 (2019), pp. 1543–1558. ISSN: 0306-4573. DOI: <https://doi.org/10.1016/j.ipm.2018.10.007>. URL: <http://www.sciencedirect.com/science/article/pii/S0306457318304692>.
- [33] Antonio A Arechar and David G Rand. *Turking in the time of COVID*. June 2020. DOI: 10.31234/osf.io/vktqu. URL: psyarxiv.com/vktqu.
- [34] Gary Marcus. *GPT-2 and the Nature of Intelligence*. The Gradient. Jan. 2020. URL: <https://thegradient.pub/gpt2-and-the-nature-of-intelligence/>.
- [35] Edward Raff. “A Step Toward Quantifying Independently Reproducible Machine Learning Research”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 5485–5495. URL: <http://papers.nips.cc/paper/8787-a-step-toward-quantifying-independently-reproducible-machine-learning-research.pdf>.

- [36] Zhaojiang Lin et al. “MoEL: Mixture of Empathetic Listeners”. In: *arXiv e-prints*, arXiv:1908.07687 (Aug. 2019), arXiv:1908.07687. arXiv: 1908.07687 [cs.CL].
- [37] Xianda Zhou and William Yang Wang. “Mojitalk: Generating Emotional Responses at Scale”. In: *arXiv e-prints*, arXiv:1711.04090 (Nov. 2017), arXiv:1711.04090. arXiv: 1711.04090 [cs.CL].
- [38] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv e-prints*, arXiv:1907.11692 (July 2019), arXiv:1907.11692. arXiv: 1907.11692 [cs.CL].
- [39] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv e-prints*, arXiv:1810.04805 (Oct. 2018), arXiv:1810.04805. arXiv: 1810.04805 [cs.CL].

A TRAINING HYPERPARAMETERS

Parameter Name	Value
Maximum input length	1024 (512 for context and 512 for label)
Validations Per Epoch	5
Validation Patience	5
Initial Learning Rate	$6.25 \cdot 10^{-6}$
Gradient Accumulation Steps	128
Batch Size	1
Optimizer	Adam
β_1 (Adam)	0.9
β_2 (Adam)	0.999
ϵ (Adam)	10^{-6}
Validation Metric	Token Accuracy
Transformer Dropout Rate	0.3
MC/EC head Dropout Rate	0.1

B REJECTION RATES FOR SEEN AND UNSEEN SCENARIOS

Table 12: Rejection rates (RR) for models, divided by if the scenario used was seen by model during training or not.

Model	Data	Scenario	Typicality R. R.	Offensive R. R.
DialoGPT	ED→Neil	Seen	0.71	0.44
DialoGPT	ED→Neil	Unseen	0.80	0.07
DialoGPT	Neil	Seen	0.31	0.56
DialoGPT	Neil	Unseen	0.11	0.58
DialoGPT (MT)	ED→Neil	Seen	0.20	0.36
DialoGPT (MT)	ED→Neil	Unseen	0.22	0.51
DialoGPT (MT)	Neil	Seen	0.22	0.29
DialoGPT (MT)	Neil	Unseen	0.13	0.38
GPT-2	ED→Neil	Seen	0.82	0.58
GPT-2	ED→Neil	Unseen	0.53	0.62
GPT-2	Neil	Seen	0.24	0.60
GPT-2	Neil	Unseen	0.29	0.38