Alma Mater Studiorum · Università di Bologna

Department of Computer Science and Engineering
Master degree in Artificial Intelligence
Natural Language Processing Course Project

BarneyBot: A Personality Chatbot based on Transformers

Authors:

Valerio Tonelli valerio.tonelli2@studio.unibo.it
Davide Sangiorgi davide.sangiorgi3@studio.unibo.it
Riccardo Falco riccardo.falco2@studio.unibo.it

June 2022

1 Executive Summary

Speaking about conversational agents implies speaking about a very complex and stratified task: human language is complex, and full of rich details and implied meanings. It is without a doubt a very important task with wide applicability, from emotional companionship to assistance in performing tasks and providing knowledge in a natural way.

Ideally, we would wish for our models to be:

1. **Coherent**: they do not contradict themselves over time;
2. **Consistent**: they follow the flow of a conversation naturally;
3. **Stylish**: they have a distinct personality, including related quirks.

From hand-crafted rule-based systems to more sophisticated neural models like sequence-to-sequence networks to transformers, new improvements have definitely been achieved in these aspects. However, albeit better and more convincing models are coming out year after year, chat-bots are still lacking crucial components such as long-term consistency and strong, distinct personalities.

The idea behind this project was initially inspired by Nguyen et al. in [12], which developed an "*open-domain response generator with personality and identity*", that is, they built chat-bots capable of imitating characters from popular tv shows like *How I Met Your Mother* or *The Big Bang Theory*. We were intrigued by the approach of their work, so we decided revamp it while extending its ideas, by creating more characters from more diverse sources. Furthermore, since the seq2seq approach had been superseded by transformers in 2017 [26], we reasoned that a new attempt at studying the same topic was justified, in order to see how much the field has improved since their study. Finally, as we noted that evaluations and comparisons of chat-bots is critically under-developed, which we attribute to a general lack of metrics in the field, we also strive with this project to perform a more in-depth study of chat-bot quantitative metrics. We do so by attempting to not only evaluate the performances of chat-bots, but also of the metrics themselves.

The strategy we adopted moving forward was to fine-tune a pre-trained model, specifically DialoGPT, being one of the SotA models with open source access and easy training objectives. We fine-tuned 8 different bots on a corresponding amount of characters from 6 different sources, in order to have a varied and data-robust approach to our results. As for the metrics, we considered commonly used ones in the field, as well as a couple of innovative ideas. Our study recognizes that there does not exist "one metric to rule them all", but rather that each can give some insight into a particular aspect of a natural conversation.

Our experiments show that our chat-bots can preserve the content and context-consistent responses of the base DialoGPT model, while gaining coherence and some hints of personality with respect to the taken source dataset. Moreover, characters that originally have a more marked personality seems to give more charismatic answers where the user asked precise questions. Overall, there is a definite improvement over the seq2seq approach, and we expect even better results by using larger architectures.

As for the metrics, we noticed some relevant results like a correlation between the algorithmic metric `Rouge-L` and neural metrics based on semantic similarity models. Significant work came out from also using all metrics in a comparative way, rather than just as an absolute value, both between different bots and by doing sentence-level pair-wise comparisons. This again demonstrates that a full evaluation of a chat-bot can be done in multiple directions which, to the best of our knowledge, are still hardly explored in the field.

Finally, our innovative attempts, based on emotion recognition (`Emotion`) and direct classification of a character through sentences (`Semantic Classifier`) did not go as well as we had hoped for: although some hints of results were obtained, implying that possibly further work in these directions may yield interesting results, for now they are unreliable metrics and require further improvements, which we propose towards the end of the report.

For these reasons we have also deployed a small set of humans metrics in order to have a stronger evaluation, even if qualitative. However, we underline that these metrics are limited just as well as the others, as they are characterized by human bias and inaccuracies, especially in small numbers.

2 Background

A general conversational agent is any system that promises the ability to maintain a conversation with a human, responding automatically to their prompts in natural language. They are one of the main topics of attraction in Natural Language Processing: the famous Turing test [25] is based exactly on this premise, that is, on the capability of a chat-bot system to fool the human at the other end of the conversation.

Chat-bots have been developed in many forms, starting in the 90s from simplistic regex systems such as Eliza [28], which are still very much in use nowadays thanks to their explainability [21, 24]. However, as is common to the entire field of machine learning, most of the focus has now shifted to the much more powerful neural-network systems [4]. Sequence-to-sequence models, originally thought for translation tasks [23], have been widely used up until 2017, when they have been superseded by transformers [26]. These systems, which are trained as general-purpose language modellers, much like an embedder but with awareness of the context in which tokens appear, can be reused for a wide variety of natural language processing tasks [15] with small additions to their structure and fine-tuning, including (but not limited to) conversational agents, which is why they have become so popular in the field.

Many transformer-based models have been developed in recent years, including BERT [6], GPT-2 [15] and GPT-3 [3]. All these have been adapted in some shape or form for text generation and human-like conversation, for instance DialoGPT [31] builds on GPT-2. Further examples of chat-bot systems include Blenderbot [18], Meena [1] and many others [5, 27].

Systematic evaluation of a chat-bot is a hard task, as appropriate algorithmic metrics would need to take into consideration the entire complexity of human language [18]: implicature, knowledge, common grounding are only a few examples of aspects which a metric cannot easily capture. As such, it is no surprise that evaluations of chat-bots, as well as comparative studies, are severely lacking. Some promising approaches have attempted to build semantically-meaningful metrics based on neural networks [30, 10], usually other language modellers. Of course, these metrics lack explainability, which is far from ideal.

Commonly considered metrics from the literature are *i)* Semantic Similarity [16], *ii)* Bleu [13], *iii)* Rouge [9], *iv)* METEOR. [2] A more in-depth explanation of them will be given in 3.3.

Broadly speaking, these metrics—coupled with qualitative analysis which is still considered fundamental in the field—are showing that, over time, chat-bots have improved in consistency and coherence, thanks to bigger and more complicated architectures. Personality, instead, seems to be tougher both to emerge within the model and to be analyzed appropriately [11]. Nonetheless, applications of a personality-focused chat-bot could be numerous: it could allow people to speak with their favourite (real and fictitious) celebrities, creating more life-like AI assistants, virtual alter-egos of ourselves, and could even aid in play-wright scripts for new episodes of a particular character show, or for entirely new character ideas.

A few works have attempted this, for instance [12], in which the authors built several conversational agents that imitate characters of popular TV shows based on a seq2seq neural network architecture. However, results were severely lacking and responses were generally unnatural, as the authors themselves admitted. To the best of our knowledge, there is no work in the field attempting to reproduce these results with newer SotA systems.

3 System Description

This section will describe the various elements of our project. The `Transformers` library [29] has been extensively used all throughout [1]: it consists of carefully engineered transformer architectures available under a unified API. This library also contains a large number of pre-trained models. It was designed to be extensible by researchers and simple for practitioners, characteristics which made it perfect for our project tasks.

3.1 Data

The data we worked on is based on scripts of famous films and tv shows. The idea is to find characters with marked traits, so that they are more easily recognizable and, maybe, more easily reproducible by our chat-bot. Unfortunately, among the many scripts available online, only few of them have explicitly written the character playing a specific line: our datasets choice highly depend on this.

The following shows and corresponding characters have been selected: *i)* Barney from *How I Met Your Mother*, *ii)* Fry and Bender from *Futurama*, *iii)* Harry from *Harry Potter*, *iv)* Joey and Phoebe from *Friends*, *v)* Sheldon from *The Big Bang Theory*, *vi)* Vader from *Star Wars*.

The factors which determined the selected shows and characters were, besides the marked and varied personalities of the character themselves, the availability of scripts, their quality and a little bit of personal preferences and knowledge of the series in question. We have also decided to choose datasets of differing sizes, with those taken from films being much smaller than those from tv series.

In the following, we will describe the procedure adopted in order to create the dataset for each character. Firstly, we do some pre-processing on the scripts in order to collect dialogues only: we remove scene descriptions, titles and so on. Then we create and save a dataset of the film/show in which each row contains a dialogue line from a character and the name of the character, so that we can later reuse the dataset for any character of the show in question, as well as to have context lines from other characters. Context lines are a necessity, as the objective of any chat-bot is to answer coherently *to* some context, which can generally consist of one or more sentences. Thus, for a chosen character, we collect all the lines assigned to the character and, for each of them, we collect also some preceding preceding lines: the line of the character is ideally what we expect from the fine-tuned chat-bot, while the preceding ones will be our context, which will be the initial input of the chat-bot.

Pre-processing is also applied to the single lines, in order to remove unusual characters and fix punctuation, among others. It is noteworthy that scripts do not follow a rigorous format: it can happen, for instance, that the character shares the same line of the script with others (e.g. *Marshall, Lily and Barney* talking), or sometimes the name can be followed by the current condition of the character (e.g. *Barney, thinking*), and so on. To solve this issue, we merged these character "names" into valid lines for the character, globally gaining a considerable number of lines. Unfortunately, even with this expedient, some scripts were of lesser quality than others, and adherence to strict formatting rules and line divisions were not followed, more often than not. This resulted in a few lines of poor quality, as well as loss of some lines altogether, particularly for the Harry Potter and Star Wars datasets, which were already the smallest ones. Some general information about the created datasets are listed in Table 1.

[1] The library is available at https://github.com/huggingface/transformers.

Character Name	# Lines	Gained Lines	Show/Film	# Show Lines
Barney	5194	3.8%	HIMYM	31776
Bender	2388	1.1%	Futurama	15226
Fry	2716	1.4%		
Harry	1037	27.8%	Harry Potter	4925
Joey	8229	10.5%	Friends	61023
Phoebe	7460	10.2%		
Sheldon	11642	2.5%	TBBT	51268
Vader	160	15.7%	Star Wars	2750

Table 1: Dataset infos. From the left: the name of the character, the number of character lines, percentage of lines gained merging all different names given to the character, tv show/film which the character belongs to, the total number of lines in the tv show/film dataset.

3.2 Chatbot Model

The architecture chosen for this project is the auto-regressive model `DIALO-GPT` [31], which stands for Dialogue Generative Pre-trained Transformer and is built on the language modeller *OpenAI GPT-2* [15]. The model had been originally trained on 147M conversation-like exchanges extracted from Reddit comment chains over a period spanning from 2005 through 2017. Its training dataset and model objectives have a focus on consistency of answers, making it ideal for sensible conversations.

Of the three available pre-trained versions of this model, we chose to work with the *small* one, using 117M parameters. This choice was made due to time constraints, since we had to train a dozen different chat-bots.

The main parameters which affect the output of DialoGPT, besides its size and training, is the generation method. Since it is an auto-regressive model, a single new token must be generated at each passage, however multiple potential tokens are outputted by the transformer network. As such, the next token can be chosen in a variety of manners. We tested a few possibilities to compare their differences, in particular:

- *Greedy search* simply selects the next word w_t keeping the one with the highest probability $w_t = argmax_w P(w|w_{t-1})$;
- *Beam search* reduces the risk of missing high probability word sequences by keeping in memory, at each step, the num_beams words with highest probability, rather than a single one, and keeping the sequence with the overall highest probability. We will use $num\_beams = 3$;
- *Sampling* chooses randomly the next word according to the conditional probability distribution outputted by the model. Since this behavior is generally too random to produce coherent sentences, a trade-off is generally employed: *top-k sampling* first filters the k words with the highest probability, redistributes the probability mass among them and then applies sampling as described before. Alternatively, *top-p sampling* chooses from the smallest possible set of words whose cumulative probability exceeds the probability p. We will attempt a mixture of these two methods, setting $top_k = 50$ and $top_p = 0.92$.

The first two methods just described do not introduce any randomness, making the answers more predictable and sometimes repetitive. This is usually a negative trait for a chat-bot, but it makes it easier to detect if the model is conforming to a selected character or not. The sampling methods are instead those generating the more interesting sentences, generally, but are by their nature less reliable.

Since the task is quite difficult by itself, we decided to not add any further level of complexity by creating, for instance, a unique model conditioned on a character token, and we instead trained separately an instance of DialoGPT for each character, saving their weights independently.

3.3 Metrics Definition

To achieve a systematic study of the chat-bot performances, we have developed a suite of metrics, attempting to cover several basic and higher-level aspects of dialogue, by taking commonly considered metrics from the literature as well as a couple of innovative attempts: *i)* Semantic Similarity [16], *ii)* BLEU [13], *iii)* Rouge-L [9], *iv)* Distinct [8], *v)* Perplexity [7], *vi)* Emotion [20], *vii)* Semantic Answer Similarity [17], *viii)* Semantic Classifier, *ix)* Human Qualitative Metrics.

Furthermore, given a test set, for pair-wise metrics we consider triples of inputs (C, L, A) where C is a context sentence given to the chat-bot, L is the label answer from the dataset and A is the chat-bot predicted answer. We compute the metric by aggregating for all pairs (C, L), (C, A) and (L, A), and we further compute a summary score, which we call Context-Chatbot-Label (CCL), as follows:

$$score_{CCL} = \frac{score_{AL}^2 + (1 - \mid score_{CL} - score_{CA} \mid)^2}{2} \tag{1}$$

We decided to develop this score to summarize the information of each specific metric for these three previous scores. It works well under the hypothesis that the corresponding metric falls in the range of values $[0, 1]$. To understand the

reasoning behind this choice, let us consider the best case scenario, in which we would like to have the chat-bot responses as close as possible to the real character responses i.e. the labels, and as high as possible of a similarity between labels and answers. In a such case we can assume that:

$$L = A \Rightarrow \begin{cases} score_{CL} = score_{CA} = s_c \\ score_{AL} = 1 \end{cases}$$

And therefore:

$$score_{CCL} = \frac{1^2 + (1 - \mid s_c - s_c \mid)^2}{2} = 1$$

If instead, for instance, we assume the worst case scenario:

$$score_{AL} \approx score_{CA} \approx 0 \text{ and } score_{CL} = 1$$

We obtain:

$$score_{CCL} \approx \frac{0 + (1 - (1 - 0))^2}{2} = 0$$

Exceptions to this scoring are the *perplexity*, which is computed on the full test-set, that is, including multiple context sentences for each reply; *distinct*, which is computed separately on the three sets; *emotion labeling*, which is also computed separately on the three sets; the *human metrics*, which are computed with an ad-hoc setup as described in Section 3.3.2.

3.3.1 Algorithmic Metrics

Algorithmic metrics for chat-bots are closely related to those for machine translation tasks, as the chat-bot answer can be considered to be a translation of sorts with respect to the previous conversational input. We use typical n-gram based statistics such as **BLEU** and **Rouge-L**, which can be considered measures of similarity between sentences. In particular, Rouge-L considers the longest common sub-sequence between label translation and machine translation. This selection on the Rouge statistics has been made since the other metrics of the suite are somewhat similar to BLEU, which we already deemed not too indicative of the quality of results 5.2.

Furthermore, we employ the well-known **Perplexity** to check overall performances and a less-known statistic named **Distinct-n**, which counts the number of different n-grams in a sentence, as a measure of text diversity. We considered for this metric $n = 3$.

3.3.2 Human Metrics

Three qualitative metrics were considered, based on the **Coherence**, **Consistency** and **Style** requirements expressed in Section 1. For consistency, testers were asked to have a short conversation with the chat-bot, and evaluate its general credibility; for coherence and style, a series of (question, answer) pairs were shown instead, with coherence questions being the standard set:

i) "Who are you?", *ii)* "What is your name", *iii)* "What is your job?", *iv)* "Where do you live?"

while the latter is evaluated with a number of character-specific questions, with a focus on how much in-character the answers are. In all cases, testers are supposed to have at least some knowledge of the character and its belonging serie/tv show, and all tests are run using the sampling generation method.

3.3.3 Semantic Classifier

We next introduce our most promising approach. The **Semantic Classifier** is a neural network with the purpose of detecting if its input sentences belong to a certain character. As we did for the chat-bot model, the architecture is common for the character that we tested, but we trained and saved separate weights for each character. The dataset for this model is obtained from the same scripts used for the chat-bot, by adding a label 1 if the line belongs to the selected character, 0 otherwise. The sentences are then encoded as per usual through *HuggingFace*. Our first attempt was to train the network with a single input sentence, but the model was not performing well. We identified two main causes for this: class imbalance and the fact that some sentences are too standard to be attributed to a character. To understand the second problem, think of sentences such as "Hello!", "I am good", "I love you": they are anonymous, and could be reasonably spoken by anyone. While it is true that DialoGPT (specially DialoGPT-small with greedy generation) tends to favor these short and standard answers, and if this lowers the score it is a desirable trait, it is also often unreasonable to guess who is talking from a single sentence. A solution which mitigates this problem is to change the input of the network to take a triple of random sentences: each sample of the dataset is now a triplet, with the label being 1 if they all belong to the selected character and 0 if none of the sentences belong to the selected character. We exclude intermediate cases with only one or two sentences from the character, both in training and in testing. In this way, we increase the probability that at least one of those sentences is sufficiently informative to guess the character and, underlying that the sentences are selected randomly, we can collect from the script any triplet of lines. Furthermore, given a triplet, changing the order of the sentences can be seen as a form of data augmentation, which is particularly useful for the smaller datasets. It may of course be unfeasible to keep all possible triplets, as they are of order n^3 if n is the initial number of sentences, but we can impose some constraints on the selection of the subset: we always set the dimension of the semantic classifier dataset for each character to be around 1M samples, and we also build it to be class-balanced. The results after training

are impressive, reaching accuracy on train, validation and test set close to 100%. The architecture of the model is mainly based on a sequential architecture of fully connected layers with ReLU activation functions and batch normalizations. The last layer outputs a single value from a sigmoid function, and L2 regularization is used on the last 2 layers, forcing them to work on a smaller domain closer to 0, which is more suitable for a sigmoid. The depth and the number of parameters has been chosen considering a trade-off between accuracy and the time needed by the model to converge.

3.3.4 Other Neural Metrics

The last of the metrics in our proposed suite are also based on neural networks. The **Semantic Similarity** is based on *BERTScore*, as it computes the cosine similarity of the embeddings produced by MPNet, an improvement over the BERT transformer [22]. The **Semantic Answer Similarity** is instead an asymmetrical semantic similarity metric based on the RoBERTa transformer [10] which takes in input a pair of sentences, in the order *(question, answer)* [17].

Finally, **Emotion** is an attempt at using emotion classification through a neural network—DistilBERT [19], a reduced version of BERT maintaining most of its knowledge, in our case—pre-trained on a dataset composed of sentences where emotions are one of the following class labels, based on Plutchick's wheel [14]: *i)* Joy, *ii)* Fear, *iii)* Love, *iv)* Surprise, *v)* Anger, *vi)* Sadness.

The reasoning here is to consider the averaged emotions of a character over a large amount of responses, giving a sort of "personality fingerprint" of the character. This should, in theory, allow us both to distinguish personalities from one another, and to potentially find similarities among similarly-dispositioned characters (eg. Joey and Barney).

The choice of trasformers for all of these metrics was mostly dependant on which pre-trained models were readily available on the *HuggingFace* API.

4 Experimental Results

The training of the DialoGPT model was performed on a NVIDIA RTX A4000 GPU with 16 GB of dedicated RAM and an environment based on python v3.8.8, using Tensorflow 2.8.0. The setup allowed us a max batch size of 8 and the training is done on 3 epochs, with learning rate of 1e-3, with a context of 2 sentences in the first run, and 5 sentences for final results.

Due to memory constraints, we fine-tuned DialoGPT with a batch size of 8 on our datasets of contexts-response tuples. As suggested by Huggingface, we set the optimizer to Adam with weight decay. The number of epochs was statically set to 3, as an empirical balance between transferring the personality of the chat-bot and keeping it consistent.

4.1 First Analysis

The chat-bot does show some promising results. In comparison to the seq2seq approach, there is, qualitatively, a higher degree of consistency, grammar correctness and coherence—which are baseline results from the DialoGPT model [31]—and a comparable transfer of personality, as it can be seen from Table 2 and Table 3).

However, the results still have some limitations: quite specific but simple questions can sometimes break the bot, which starts giving non-sense or repetitive answers (e.g. *Question: who are you? DialoGPT: I'm here*). This behavior is improved by characters having bigger datasets, and where such question-answer "patterns" are explicitly given as sample during the model training, (e.g. *Question: who are you? BarneyBot: I'm Barney*). These situations may also be made less recurring by using the sampling generation strategy rather than the greedy or n-beams ones, which also empirically increases variety and personality at the cost of coherence and consistency. Furthermore, it is somewhat subjective the degree to which the bot has learned the correct personality, and it is not always consistent within the same chat instance nor coherent (eg. it may respond with a different name when being asked theirs, or with different names when asked multiple times).

4.2 Metric Results

We describe here the general trend of each metric, showing our most significant results [2].

Perplexity is usually below 10, being generally quite unpredictable in its behavior. It reaches maximum value of 20.6 with Vader, and minimum of 7.4 with Sheldon. The value does not seem to be strongly correlated with the dataset size, as for instance Fry has a value of 17.1 and Harry a value of 11.4. Of course, if we attempt to compute the perplexity of a character with a different test set [7], the value jumps ten-fold, which is a good sanity check.

The rest of our experimentation relies of comparative analysis. Given the triplets (C, L, A) described in Section 3.3, we compare the answers of a chat-bot, using as support the test set from its source tv/series, against:

- the label, that is, the answer of the original character from the scripts;
- the answer given by the same chat-bot with a different generation method. We test all possible pairings between our three generation methods *greedy*, *beams* and *sampling*;
- the answer given by the default DialoGPT bot;

[2] If one is interested in seeing all the visualizations we have produced, please refer to the OneDrive folder

- the answer given by a different character. We do so for only a few selected pairs, due to time constraints. In this case the support set is given by a small common dataset including lines from all dataset sources as well as a few custom ones.

At a first glance, if we compare the three pairings (C, L), (C, A) and (L, A), it appears that the semantic similarity, rouge-L and semantic answer similarity are all close to each other, regardless of the comparison pair. In particular, they share similar relative trends, albeit with different absolute values, as it can be seen in Figure 1.

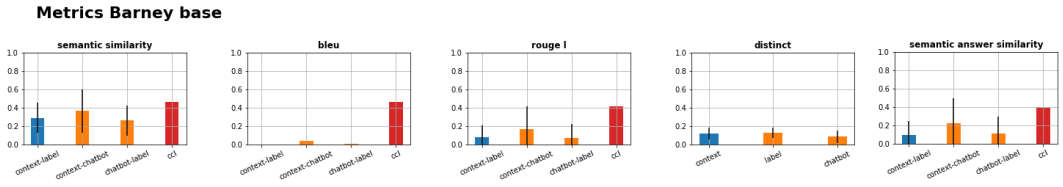

Figure 1: semantic similarity, bleu, rouge l, distinct and semantic answer similarity results comparing Barney chat-bot (with greedy generation method) with context and label. On plots of semantic similarity, bleu, rouge l and semantic answer similarity also the *ccl* metric that we introduced.

Based on these metrics, the chat-bot seems to generate sentences which are more highly correlated with the context than the label. However, with respect to the default DialoGPT bot, this dependency is lower, and this is even more true if we use sampling, although it never reaches the levels of the labels in the script.

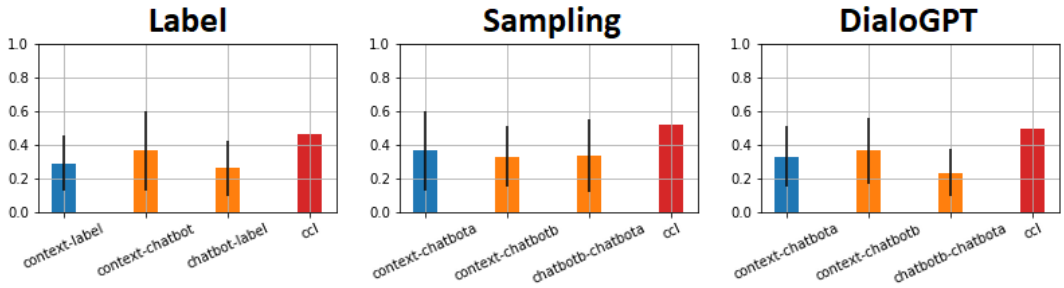

Figure 2: **Semantic Similarity results** semantic similarity applied on different comparisons. On the left we compare semantic similarity between context, chatbot and label. In the other plots *chatbot a* refers to the trained chat-bot with greedy generation method, while *chatbot b* refers to sampling method in the middle and to the default DiagloGPT on the right.

Distinct and BLEU also seem to have some visual similarities in behavior, however Distinct tends to be a lot flatter and BLEU is strongly regularized towards zero.

As for the emotion labelling, it is meant to capture differences between character personalities. However, in practice the comparative spider charts are almost always overlapping: although sometimes we can seem to find such differences, the tendency is to almost always have high values of both joy and anger, and sometimes sadness. This difference can happen between different character [Figure 3], or even within the same character using a different generation method [Figure 4].

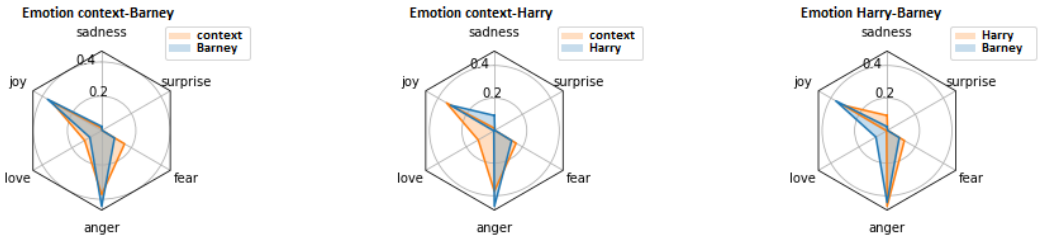

Figure 3: **Emotion Barney-Harry**. Emotion comparisons between a common context sentence, Barney and Harry responses. The metric still shows high bias on emotions of joy and anger, but in this case also catches some differences between the two characters.

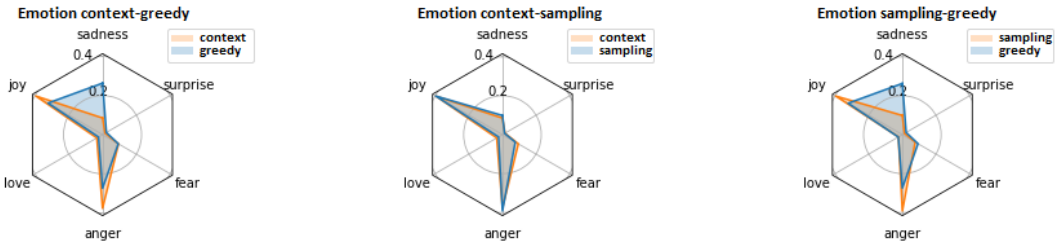

Figure 4: **Emotion Barney greedy-sampling**. Emotion comparisons between a context sentence and two generation methods (greedy and sampling) for the character Barney. The metric still shows high bias on emotions of joy and anger and unexpectedly shows relevant differences between the two generation methods, more than what we would have between some pairs of characters.

We have also attempted to compute correlations between two emotion vectors, for instance between different characters. It is usually quite high in all comparisons we made, and it is almost 1 when considering the same bot with different generation methods.

Finally, the semantic classifier is performing tremendously well on the test set taken from the original dataset, but suffers from really high variance when working on sentences generated from the chat-bot and also seems to be strongly affected by some degree of noise such that, if we compare two characters c_a and c_b on classifier C_a, it sometimes gives a higher average score to the wrong character c_b. The same can be said about the comparison between a character and the default DialoGPT model. As such, it is really difficult to make reliable statements using this metric, even if it often follows what we expected like in the following plots as in Figure 5.

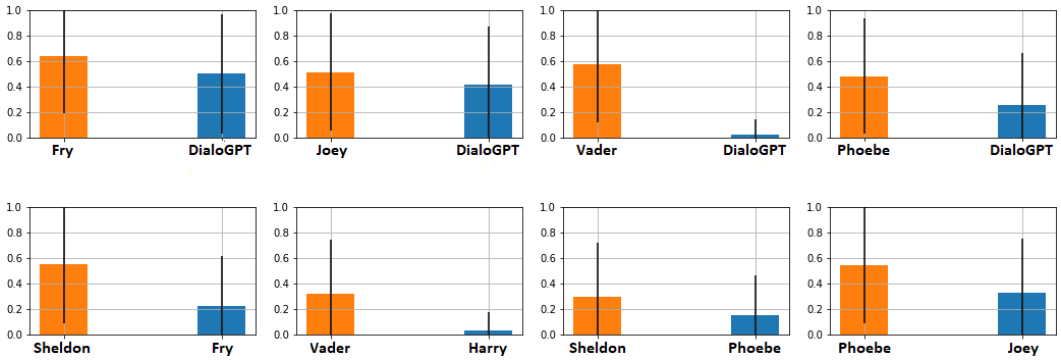

Figure 5: **Semantic Classifier positive results**. Semantic classifier applied to compare chat-bot performances; on the vertical axis the confidence value from the classifier, on the left of each chart there is the positive label, on the right the negative one. The figure shows in the first half comparisons w.r.t. the default DialoGPT to check improvements, in the second half comparisons between different characters to check differentiation between them.

5 Analysis of Results

We further discuss the results of our experimentation and metrics, which was mostly a relative analysis of values, rather than of their absolute values, since we lack a strong baseline against, say, a simplistic rule-based or dummy chat-bot. We note that the evaluation process is both on the chat-bot *using* the metrics and *for* the metrics themselves, which is possible thanks to the large amount of comparisons at disposal.

5.1 General Results

Regardless of the chosen metric, greedy and n-beams generation methods are practically equivalent, which is not surprising since they are both deterministic and similar in nature; the same reasoning can be applied to explain the difference between those two and the sampling methods. However, the metrics do not show a clear winner, and this is corroborated in practice by the intuition that the deterministic methods are more consistent while the sampling-based ones are more varied. It should be noted that this may be a problem specific to the small model, as larger models may have more answering options and concepts to pick from and less uncertainty, counteracting the noise introduced by the sampling process.

Comparisons between default and fine-tuned models, and even between different characters, do not show consistent results. This also seems to hold true relative to specific improvements: while the 5-context version of the chat-bot seems to perform, anecdotally, slightly better with respect to the 2-context one, this is not shown by any of the metrics in a consistent manner. In short, the metrics are for the most part unable to distinguish the chat-bots, unless they are vastly different. Therefore, in the following we will focus on the analysis of the single chat-bot performances, and of the metric themselves.

5.2 Quantitative Analysis

As a measure of overall performances, the perplexity values of our chat-bots tend to be comparable to other DialoGPT-small bots present in the HuggingFace library of conversational agents [3], for instance the chat-bot `AK270802/DialoGPT-small-harrypotter` has a perplexity of around 11.6. This corroborates the idea that switching to a larger version of DialoGPT would produce significant improvements, as those bots from the same list have a perplexity on average slightly above 1.

As for the similarity metrics, the high (C, A) values can be seen as positive, as it means the chat-bot is responding meaningfully to the context. However, the fact that they are higher than (C, L) also implies that real answers have a lot more variety—in other words, independence—that the chat-bot is still unable to use. This suggests that it may be helpful to use these metrics not just by attempting to maximise their absolute value, as it is typically done [17, 30], but as a relative comparison to minimize. One of the best results in this line of thinking is that, with respect to the default DialoGPT bot, this dependency is a lot lower for a character chat-bot, and this is even more true if we use sampling, although it never reaches the levels of the labels in the script, suggesting that the training is not over-fitting and, quite the contrary, is loosening the coupling between context and chat-bot answers.

In terms of evaluating the metrics themselves, the similarity of results between semantic similarity, rouge-L and semantic answer similarity makes these metrics redundant, but at least reliable and consistent, properties that we had difficulty finding for other metrics. Rouge-L is particularly noteworthy since it is an algorithmic measure, with the other two being neural and considered more strongly correlated with human evaluation [17]. As such, we tentatively propose that Rouge-L could be an approximate replacement for these two others, and that it seems that the longest sub-sequence problem seems to better capture semantic meanings than n-grams counting.

As for the secondary similarity metrics, distinct falls somewhat flat between labels and generated answers, and we argue it is a good sign: it demonstrates that the chat-bot is using a vocabulary as varied as that of the source dataset. On the other hand, the very low values for BLEU probably imply that the metric is inadequate: since it is meant for machine translation, this is unsurprising.

Among our most promising ideas, the emotion classifier was underwhelming: most comparisons show little to no difference, even among very different characters such as Vader and Barney, while sometimes there are greater differences between different generation methods of the same characters. These, combined with the overwhelming bias towards joy and anger, which are quite conflicting emotions, as well as the lack of a neutral emotion state, makes us very wary of the metric.

The semantic classifier was, unfortunately, only slightly better. We cannot say that it is over-fitting, given the high performances against the test set, it may however be a crucial point that it has never seen sentences from different datasets during training or testing, and as such the tests may be biased. Another cause for poor performances could be the nature of the chat-bot itself: in order to not degrade performances, we keep low the number of training epochs, resulting in a lower influence on the original DialoGPT sentences. As such, while sentences are kept more consistent, they are also still highly dependent from the original, anonymous, training of DialoGPT, and the result is therefore a weakly differentiated chat-bot. Most likely, a combination of how hard it is to capture meaning in conversation, as well as both of the effects just described, is at play here.

5.3 Qualitative Analysis

Our analysis of metrics has proven, once more, that qualitative analysis is still a fundamental step in evaluating chat-bot performances. We thus continue our considerations through the human metrics we have defined in Section 3.3 and the chats examples provided at the end of this report (Table 23), as well as in the csv files in each character folder, whose average scores are reported in Table 4 along with the number of human testers for each bot.

From conversations, coherence seems somewhat partial, about the same of the base DialoGPT model. Sometimes, the flow of the conversation is followed quite nicely, but other times it is also abruptly changed or misunderstood. However, the scores seem to suggest that the dataset size unexpectedly plays a very impactful role on coherence, albeit the base model is the same. This may be explained by the fact that coherence is necessarily perceived w.r.t the particular series or show of reference—in other words, it is not disentangled from the other metrics—which the bot may have not learned well.

Consistency is poor, with answers to the common questions varying between correct and expressive ("I am Dr. Sheldon Cooper.") to outright random. We attribute this mostly to sampling, and as a consequence we suppose deterministic generation would help greatly with this issue. Scores seem to be the most random here, as well, with Vader surpassing Sheldon and being very close with Barney.

Characters trained on small datasets were less affected by the fine-tuning, and as such they show little personality changes w.r.t. the default DialoGPT. Their answers usually seem related to the context, but seem less correlated with the entirety of the chat and even less correlated with the context specific to the series/show, which confirms what the similarity metrics were already suggesting. Larger datasets decisively lead to better results: as we can see also in Table 3, Sheldon answers are, for instance, very much in-character and also refer to his passion for train modeling and quantum physics. However, we also note that marked elements of personality, such as catchphrases, are not strongly maintained. Barney's "Legendary" and "Expecto Patronum" are not common sights, unfortunately. An exception to this is Vader,

[3] The list is at HuggingFace - Conversational as of 21/06/2022.

which often uses the expression "My lord". We hypothesize this may be a form of over-fitting. The style metric matches this line of reasoning very well, being lower for Harry, slightly higher for Vader and best for Barney and Joey.

Character	N	Coherence	Consistency	Style
Barney	12	3.00	2.82	3.18
Bender	*2*	*5.00*	*2.50*	*2.50*
Fry	*2*	*1.50*	*1.50*	*2.50*
Harry	9	2.11	1.22	1.78
Joey	7	3.29	3.86	3.43
Phoebe	*3*	*3.33*	*2.33*	*1.67*
Sheldon	7	3.57	2.29	2.57
Vader	5	0.50	2.75	2.25

Table 4: **Human Metric scores**. From left: number of samples N, coherence, consistency and style score from random testers. In italics the rows with too few testers to be taken into account. Overall scores are quite low, sometimes passable.

6 Discussion

Our results prove that we were able to inject some knowledge from the selected characters through fine-tuning of DialoGPT on the scripts of the film/series, and that personality is indeed one of the aspects that the chat-bot is able to capture, albeit how convincing it is depends strongly on chat instances. On the other hand, we have seen little coherence and a reduced consistency, particularly with respect to the base model. Certainly, among the main limitations of our experiments are the complexity of the model, since we used DialoGPT-small and the nature of the datasets, with all scripts badly formatted to some degree and missing the assignment of at least some lines to characters.

Many improvements could be made to the base approach, as well: we fine-tuned the pre-trained DialoGPT model on a dataset, but other alternatives could be tested by using this work as a baseline, like approaching the task as a style transfer problem, or using the semantic classifier as a discriminator for GAN-like training. Certainly, switching to a bigger architecture such as DialoGPT-medium and using more sentences for context are also very much needed improvements.

As for the semantic classifier, we hypothesize that it could be improved with small modifications to its training dataset, for instance by introducing lines from different datasets as negative samples and by trying to reduce the variance by removing anonymous sentences with little meaning or personality. Furthermore, we argue that modifications to its architecture could be greatly beneficial: for instance, Recurrent Neural Networks could be fed an arbitrarily large amount of sentences, potentially solving the variance and instability issues. The emotion metric could also be developed further, for instance by adding a very much needed neutral emotion.

Unfortunately, the choice of a sufficiently correct and complete suite metric is still an open problem. More metrics could be tested, and as of now the metrics we do have give relatively little insight, especially about comparative evaluations.

References

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] Guendalina Caldarini, Sardar Jaf, and Kenneth McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1):41, 2022.

[5] Rollo Carpenter. Cleverbot. `https://www.cleverbot.com/`. [Online; accessed 18-June-2022].

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[7] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[8] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2015.

[9] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[11] Yukun Ma, Khanh Linh Nguyen, Frank Z Xing, and Erik Cambria. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70, 2020.

[12] Huyen T M Nguyen and David Morales. A neural chatbot with personality. 2017.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[14] Robert Plutchik. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350, 2001.

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[16] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[17] Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. Semantic answer similarity for evaluating question answering models. *arXiv preprint arXiv:2108.06130*, 2021.

[18] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot, 2020.

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[20] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[21] Jagdish Singh, Minnu Joesph, and Khurshid Jabbar. Rule-based chabot for student enquiries. *Journal of Physics: Conference Series*, 1228:012060, 05 2019.

[22] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[24] Sandeep A Thorat and Vishakha Jadhav. A review on implementation issues of rule-based chatbot systems. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[25] Alan M Turing. Computing machinery and intelligence. In *Parsing the turing test*, pages 23–65. Springer, 2009.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[27] Shivang Verma, Lakshay Sahni, and Moolchand Sharma. Comparative analysis of chatbots. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[28] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan 1966.

[29] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[30] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[31] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.

Barney [12]	**Barney** (Ours)
user: who are you?	**user:** who are you?
chatbot: i am the master of the most barney	**chatbot:** I'm Barney
user: hi barney!	**user:** hi barney!
chatbot: hi , have a drink .	**chatbot:** Hi, Barney. Long time no see. Have a seat.
user: oh thanks. what drinks do you have?	**user:** what drinks do you have?
chatbot: whatever what do you do ?	**chatbot:** Gatorade and soda.
user: i'm a student.	
chatbot: a . . . buddy	
user: yeah?	
chatbot: sleep with her mom ?	
user: lol no	
chatbot: because she ' s not happy .	
user: ha ha okay. how's work?	**user:** how's work?
chatbot: okay	**chatbot:** Oh, I love it.
user: just okay?	
chatbot: oh, okay	
user: what do you think of that girl over there?	**user:** what do you think of that girl over there?
chatbot: do not even try to try to that girl .	**chatbot:** She's great.
user: nooooooooo	
chatbot: legendary !	

Table 2: **Barney chat comparison** on the left the reference chat proposed by [12], on the right we submitted the same sentences (without forcing the conversation) to our chat-bot. Notice the increased consistency of the responses.

Sheldon [12]	**Sheldon** (ours)
user: hi	**user:** hi
chatbot: hello	**chatbot:** Hi.
user: who are you?	**user:** who are you?
chatbot: i ' m sheldon .	**chatbot:** I am Dr. Sheldon Cooper.
user: what's up?	**user:** what's up?
chatbot: i see how i could do .	**chatbot:** Well, Ive decided to refocus my efforts and use my time to reach out to the world of model trains.
user: what can you do?	**user:** what can you do?
chatbot: i have to tell you a secret .	**chatbot:** Oh, I have a few ideas. I think we should start with a little story structure.
user: tell me	
chatbot: no.	
user: what do you like to do for fun?	**user:** what do you like to do for fun?
chatbot: i want to leave .	**chatbot:** Hmm. I dont know. I was hoping to get a few ideas out there before I invested money in string theory.
user: to go home?	
chatbot: no.	
user: just okay?	
user: how are you doing?	**user:** how are you doing?
chatbot: i just wanted to sleep .	**chatbot:** Im doing okay.
user: sleep!	
chatbot: penny , i am getting too much for you .	

Table 3: **Sheldon chat comparison** on the left the reference chat proposed by [12], on the right we submitted the same sentences (without forcing the conversation) to our chat-bot. Notice the increased consistency of the chat and the quite specific references to the two main passions of Sheldon: train modeling and quantum physics.