Natural Language Processing Project Work

Does there exist a metric to evaluate a chatbot?

Davide Sangiorgi, Valerio Tonelli, Riccardo Falco

Academic Year 2021-2022

1 Executive Summary

Recent studies have highlighted the effectiveness of conversational agents, with some such as ChatGPT making names for itself. In our previous study, we were successful in transferring some of the personality of a TV show character to a chatbot through fine-tuning, although the change in personality was still lacking in coherency. Within that work, we have also attempted to perform an analysis on the chatbots performances, acknowledging that the evaluation of a conversational agent's personality remains a complex and unresolved issue.

Here we will try to explore this latter topic further, by using a wider range of metrics from the literature to perform evaluation, in order to study how and which metrics may be well-suited to investigate specific aspects of conversations. Namely, we approach the evaluation problem by correlating it to the one of Machine Translation (i.e. by using metrics like ROUGE, Google BLEU, COMET, ...), of Text Generation (i.e. metrics like Distinct, grammar-based metrics, ...) or even to a problem of sentence-level Semantic Similarity (i.e. metrics like MPNet Semantic Similarity, Semantic Answer Similarity).

Additionally, we propose some novel classification-based approaches. We show the promises and great limitations of emotion classification, which we think may become a great tool to recognize personality types. We further present two alternative approaches to classification for chatbots: one based on Machine Learning, aimed to find a latent embedding space which allows to separate the data according to a given metric, and another based on Information Retrieval, a method based on similarities among class based Term Frequency-Inverse Document Frequency (c-TF-IDF). These two approaches have been designed to overcome the limitations of the neural chatbot classifier of our previous project.

For all of these groups of metrics we perform both singular analysis and correlation comparisons, demonstrating that, in most cases, metrics analyze different aspects of language and are difficult to compare, and that all have awful-to-decent degrees of efficacy for our task.

Finally, we explore correlations among metrics according to a human ranking of chatbot-generated responses, to gain insight into which metrics are closest to human judgment. Our results show that single metrics cannot correlate well with human judgment under all tests, and therefore we strongly suggest for research in this area to focus on a coordinated, conscious effort for the creation of a suite of metrics for general-purpose chatbot evaluation.

2 Background

A chatbot is any software application aimed at mimicking written human speech within the context of use of a prolonged exchange with a human interlocutor. The responses provided by the software should be coherent with the discourse at hand, while also being convincing, natural, insightful and, possibly, charged with a personality. Due to the inherent complexity and variety of language, this task has proven difficult for a long time, even for state-of-the-art technology [3]: implied meanings, ambiguity at both the syntactical and semantical level, contextual clues, cultural aspects and idiosyncracies of individuals and groups are all elements that a compelling conversational agent must take into consideration and integrate into its logic.

The generation of software conversations has been tackled over time in a variety of ways: from rule-based, regex-like systems such as Eliza [32] to learning-oriented approaches such as sequence-to-sequence models [27] and fuzzy logic, as in the internet-famous Cleverbot [4]. However, it is only through the more general task of modelling language as a whole [16] that the best performances can be achieved, and the building block of these systems are attention mechanisms, namely the transformer architecture [30]. A noteworthy chatbot belonging to this category is DialoGPT [38], built upon the GPT-2 language modeller [16], but many others have been developed [31]. In 2022 new standards in terms of quality and persuasiveness are being achieved thanks to ChatGPT, developed by OpenAI (although a paper has yet to be released on this model), LaMDA [28] by Google and Blenderbot 3 [22] by Meta AI.

These recent improvements are pushing the boundary of what artificial agents are capable of delivering, to the point that concerns related to the use of these systems in specific scenarios are mounting [19]. This being said, quantifying these improvements is arguably harder than the original task, as it is finding a comparative measure between two dialogues. Nonetheless, a measurable metric (or a set thereof) would provide a benchmark against which to classify conversational systems, and it may potentially even pave the way for a greater analysis of all language models. It is therefore a problem of great interest in the field.

Some chatbot performances metric do exist, though they are mostly taken from other applications in natural language processing. Typical metrics include machine translation metrics such as BLEU [14] and METEOR [1] and text generation metrics like BERTScore [36]. However, all of them have severe limitations and degraded performances in edge cases, due to their different scope of application or, in the case of metrics based on neural networks, their unexplainability. Our work attempts to combine and compare multiple such metrics, as well as a couple of innovative ideas, in order to try to find a suite of metrics which can be indicative of the general performances of a conversational agent.

3 System Description

This section describes the definitions, processes, models and metrics in use for this project. We begin by detailing the characteristics of the chatbots, and we then move onto the definition and training of the tested metrics.

The `Transformers` library [33] 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, some of which

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

we use as a foundation for our metrics. The `NLGMetricVerse` library has also been used for some metric implementations [2].

3.1 Chatbot Model

The chatbot model chosen for this project is `DIALO-GPT` [38], which stands for Dialogue Generative Pre-trained Transformer. It is built on *OpenAI GPT-2* [16], a language modeller 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 (117M parameters), a choice made due to time and memory constraints.

We fine-tuned several identical copies of the model on different datasets taken from tv shows and films, with those taken from films being much smaller in size than those from tv series. The idea is to let the models reproduce characters with marked and distinctive traits, so that they are more easily recognizable and distinguishable during analysis. Unfortunately, among the many scripts available online, only few of them have explicitly written the character playing a specific line, limiting our choices considerably. The other factor which determined the selected characters were also a little bit of personal preferences and knowledge of the series in question. 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*.

A typical train/val/test split was done on each dataset after an ad-hoc pre-processing phase to remove undesired symbols and lines. For a given character, each row of its prepared dataset contains a response by that character as well as the 5 preceding lines of dialogue, which serve as context. Unfortunately, after pre-processing some lines of the source material were lost or ended up being of poor quality, especially for films, due to adherence to strict formatting rules and line divisions not being followed in the source itself.

Once trained on these datasets, the models could be used to generate new sentences in character. In particular, since DialoGPT is an auto-regressive model, it appends a single new token to the sentence being generated at each passage. However, although there exists a token which is considered best by the network, always choosing it tends to produce uninteresting, generic, "safe" responses. Instead, a better approach we have taken is to consider the conditional probability distribution on all tokens and sample from it. A trade-off between speech coherency and speech variety now arises, which is usually solved by limiting the sampling behavior to a parametrized top percentile or cumulative probability.

3.2 Metrics Definition

There exists a plethora of metrics for Natural Language Processing, but only few are applicable to at least some degree to the task of dialogue generation. Here we describe the metrics we considered for this project, which include several well-known and commonly used metrics from the literature. We also highlight a few novel ideas, of which we further detail the reasoning behind.

3.2.1 Machine Translation Metrics

Machine translation metrics quantify the similarity between a correctly translated text, used as a reference, and the translation generated by a model. There exists many metrics for this task [6, 26, 18, 10, 1], since it is far more approachable than evaluating the generation of arbitrary text.

A basic approach to the problem is to directly compare the two strings, yielding a higher score the more the two are similar. The Levenshtein distance (minimum number of edits to transform one string into another) can be used to this endeavour, but it lacks flexibility. A recent proposal called the **Extended Edit Distance** (EED) [25] improves upon this simple idea by adding a "jump" operation to align characters, making it more usable for machine translation. Another similar approach is known as **Translation Edit Rate** (TER) [23], which computes a normalized number of edits on *words*.

BLEU [14] is a long-standing benchmark for machine translation, computing the correspondence between prediction and (one or a set of) references as the n-gram precision with a brevity penalty. Since this metric is known to have bad performances when used on short sentences, rather than on a corpus [26], we employ a variant of this metric known as **Google BLEU** [35]. Under this modification, the n-gram recall is computed in substitution of the brevity penalty, mitigating its impact on shorter predictions.

ROUGE [10] is another well-known metric for summarization and machine translation evaluation. It is a package of 5 different metrics which are n-gram based, much like BLEU; among these, we selected **ROUGE-L**, which approaches the problem of n-gram similarity by finding the longest common sub-sequence between the two texts.

Yet another flavour of the same approach comes from **METEOR** [2], a metric specifically studied to improve upon BLEU through stemming and synonymy matching, along with similarly-computed n-gram precision and recall; it grants more flexibility to the translation and therefore it has better correlation against human judgement.

As for metrics based on neural networks, a recent work called **COMET** and published in 2020 [18] demonstrated state-of-the-art performances on the WMT19 dataset; the metric takes in input an additional source text, beside the reference(s) and the prediction.

[2] A project of the University of Bologna, available at https://github.com/disi-unibo-nlp/nlg-metricverse.

3.2.2 Text Generation Metrics

Although we cannot find a single metric to evaluate text generation, we can nonetheless consider metrics for specific, lateral qualities of generated sentences with the objective of performing sanity checks and, possibly, find a correlation against our testset labels.

Despite the remarkable growth in many text generation tasks, nearly all existing generation systems suffer from the repetition problem [7]. This problem refers to an undesirable effect that the results of the generation system always contain duplicate fragments. A basic statistic that can be computed to understand the impact of such problem on a generated sentence is **Distinct**, the number of different n-grams in a sentence. We have also attempted to test grammar correctness for sentences in the form of a neural corrector trained from the language model T5 [17]: we let this model correct text, and we compare the distance between the original and the corrected sentences, a measure we call **T5 Grammar Correction Distance**. Another general-purpose metric for the performance of a model is **Perplexity**, a dataset-dependant measure of how much a given model can predict the occurrences of tokens in a sentence.

Finally, we attempt a more general evaluation for generated text by employing the neural metric **BLEURT** [21], which takes in input both a reference and a prediction and supposedly conveys how much the prediction is fluent and carries the meaning of the reference.

3.2.3 Semantic Similarity Metrics

While machine translation metrics focus on lexicographic similarity between texts, they may lose on or excessively penalize sentences which "look" different, albeit their meanings match. Measuring the semantic similarity of two texts is a top-down approach to the same task, which can also have wider applicability (e.g. document classification, question/answer).

A typical approach for semantic comparison is to represent text through some word-meaningful embedding. **Word Mover Distance** [34] uses word2vec [13] to then find, in the embedded vector space, the minimum cumulative distance needed for the (weighted) words of one sentence to map onto the words of the other.

The embeddings may, of course, be also computed through neural networks. **BERTScore** [37] is a well-known example of such a metric: it encodes tokens through the language model BERT [5], and then computes pairwise cosine similarity between the resulting vectors to pair them optimally. Similar metrics can be constructed by employing any language modeller. Indeed, our **MPNet Semantic Similarity** embeds text through MPNet [24] and similarly computes a cosine similarity, the only difference being that it works at the level of sentences, rather than words.

Specifically for the sub-task of question answering, there exists a metric proposal named **Semantic Answer Similarity** [11], where the authors use a cross-encoder based on RoBERTa [12] to distinguish between question and answer sentences, and therefore compute a reference-aware similarity.

3.2.4 Classification Metrics

Absolute performance metrics are general-purpose and overall desirable, but it may also be interesting to work in a comparative setting, measuring relative performance differences between chatbots. This greatly simplifies the problem of measuring the characteristics of a conversational agent, as it becomes akin to a classification task, while at the same time providing insight into how similar or different agents are. Furthermore, if this same comparison is done against some reference such as a testset, it can provide valuable data into how a chatbot matches expected outputs, so long as the strategy for classification is understood.

Emotion Classifier A parallel task to that of classifying characters is to grade the emotional spectra of different personalities: albeit any character may express, at a given time, any subset of emotions, their average tendencies may be distinctive of the type of person they represent. For instance, we may imagine Barney to be mostly sarcastic and carefree, while Vader may be mostly angry and menacing. These differences, we hypothesize, should be spotted when looking at a sufficiently large and varied set of responses.

Emotion classifiers on sentences already exist [29]; we will attempt to use them, and in particular DistilBERT [20]—a light variant of the language modeller BERT [5]—having been fine-tuned for emotion classification. The objective is to obtain a sort of "emotional fingerprint" for each character, with the hope that, by decoding a personality into emotions with quantifiable intensities, we may be able to compare characters through vector-space operations, such as computing their correlation or visualizing them in a reduced space. Furthermore, this type of analysis may be extended to spot differences between reference sets and generated sets.

Frequency Classifier Another way to look at the problem for the evaluation of a text generation model is to reshape the task following an Information Retrieval (IR) approach. Common applications of IR require searching for relevant documents with respect to user queries in the repository of unstructured documents. The goal is to return the documents ordered according to their relevance with respect to the query. The idea behind this viewpoint is that the more query terms are frequent in a document, the more the score of that document should be high.

Thus, it is possible to consider each script as a single list of documents and a set of chatbot replies as queries. The frequency classifier is meant to classify a chat and return the name of the character for which his or her lines script is more similar according to a given distance function. Cosine similarity is a typical choice, so as to return the similarity computed over the frequency vectors of each reference-query documents pair.

More specifically, we decided to implement a Class-based Terms Frequency-Inverse Document Frequency (CTF-IDF) [9], a special case of the standard TF-IDF applied for document classification tasks. According to the architecture used in BERTopic [8], we can define the cTFIDF as follows:

$$cTFIDF = \frac{t_c}{w_c} \cdot \log \frac{m}{\sum_j^n t_j}$$

where t_c is the frequency of each word extracted for class c, w_c is the total number of words in class c, m is the (unjoined) number of documents, and n is the set of classes. The logarithm can be seen as a re-weighting term based on all classes.

The model has been trained on a list of documents D_c for each character c. An additional "Default" class was added, trained on a small subset of the original dataset [3]) on which DialoGPT had been trained.

DistilBERT-Embedder Classifier The DistilBERT classifier is an embedding-based model with the purpose of detecting to which character a set of input sentences belong. The model works in two steps: the sentence embedder, based on the DistilBERT sentence transformer from *HuggingFace*, projects the group of sentences into a 4-dimensional space and then a KNN algorithm is applied for the actual multi-label classification.

The dataset for training this model is obtained from the same scripts used for the chatbot, by adding as label a one hot encoded vector indicating which character the sentences belong to. Most of our research focused on the training of the sentence embedder, which was tackled with the triplet loss: starting from an input-label couple, named as *anchor*, we search for two other inputs, one with the same label (the *positive* example) and one with a different label (the *negative* example). The triplet loss is then defined as follows:

$$TripletLoss(A, P, N) = max\{0, ||f(P) - f(A)||_2^2 - ||f(N) - f(A)||_2^2 + m\}$$

where A, P and N are the anchor, positive and negative inputs, respectively, f is the embedding function and m is a margin. The purpose of this loss is to bring together embeddings belonging to the same class and increase the distance (L2 norm of the difference) between embeddings belonging to different classes in order to build clusters used for the actual classification. Since it would be counterproductive to push too far away points of different classes, a margin m is introduced, which is the ideal distance at which we would like the clusters to be created. Indeed, if $||f(N) - f(A)||_2^2 - ||f(P) - f(A)||_2^2 > m$, the loss would be 0.

Another key aspect of training is the *semi-hard negative mining*: in order to make easier for the model to converge, not all triplets can be used, in particular hard negatives make learning really hard for the model and have to be filtered out: these are the triplets (A, P, N) for which $||f(P) - f(A)||_2^2 \geq ||f(N) - f(A)||_2^2$. If the model is trained on the filtered dataset for a certain number of epochs it is then capable of better embeddings and the number of hard negatives decreases, so they can be slowly reintroduced to the model; in other words, the dataset can be filtered again and the training repeatedly follows the previous steps until it improves no more.

Even if it may not perform as well as other tested methods, the DistilBERT classifier has a quite interesting property: if a character has to be added, unlike other possible deep learning models, only the KNN needs to be trained while the embedder does not, since it should agnostically create a different cluster for any new character. We expect the accuracy on the new class to be lower, compared to those it trained on, but it is an interesting general-purpose approach.

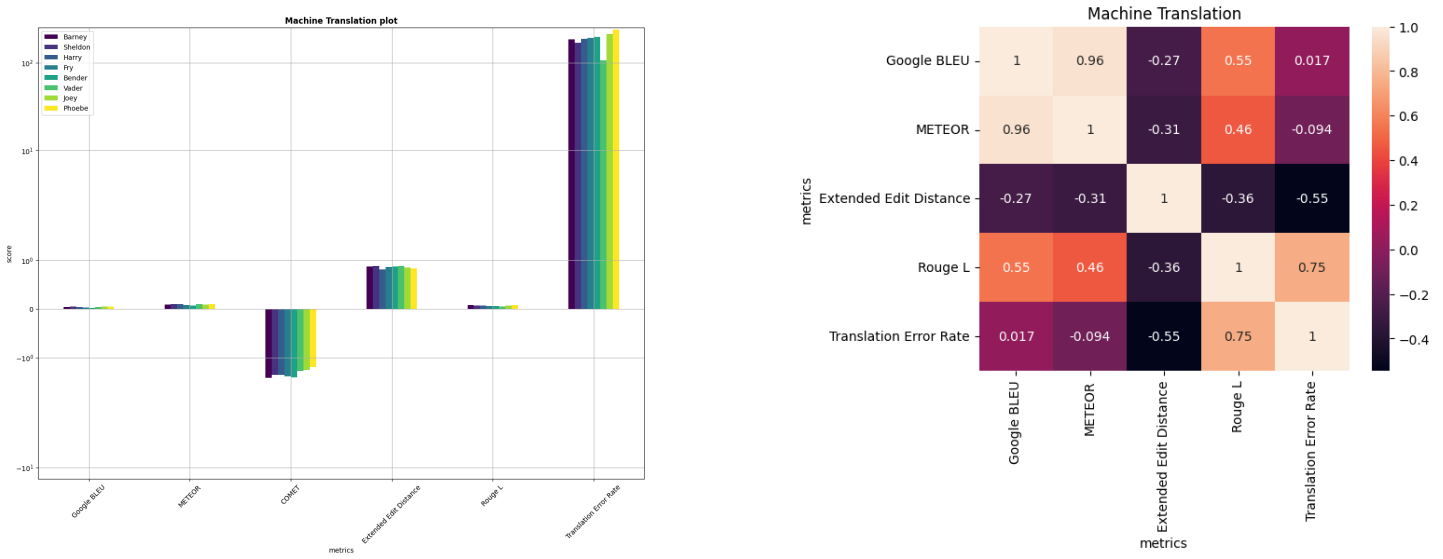
4 Experimental Results

This section presents the quantitative and qualitative results coming from the application of our metrics under test to the chatbots, all as defined in Section 3. Furthermore, we make some considerations on each metric specifically, as well as on their comparison.

4.1 Machine Translation Metrics

Let us first make some considerations about each metric from Figure 1. Google BLEU generally produces very low values: n-grams may not be enough to compute the semantical meaning of sentences and fail to capture longer sequences. Similarly, METEOR and Rouge-l tend to low values. METEOR produces slightly higher values, possibly because it takes into consideration more advanced features such as stemming, however we note that it is extremely correlated with Google BLEU, suggesting that these improvements do not amount to much in the context of text generation. EED (Extended Edit Distance) and TER (Translation Edit Distance) are also in line with the other results: given a low similarity between the reference and the chatbot answers, we expect that the number of edits needed to transform one into the other is high. Due to their similar nature, we also have a quite strong correlation between the two metrics.

[3] https://github.com/microsoft/DialoGPT#data-loading



(a) (b)

Figure 1: Scores and correlation matrix for Machine Translation metrics.

The most unexpected results come from the negative values of COMET and from the correlation between several metrics with Rouge-L. We believe that the correlation between TER and Rouge-L may be due to their common behavior of aligning sentences for comparison, while its correlation with Google BLEU/METEOR is due to the common problem they tackle. As with our previous work, Rouge-L seems to be able to synthesize a lot of information, more that the other algorithmic metrics of this category. As for the negative values produced by COMET, they may be a sign of the need for a fine-tuning of the COMET model that is suited for our task at hand, since the base model is usually applied to translation between different languages.

4.2 Text Generation Metrics

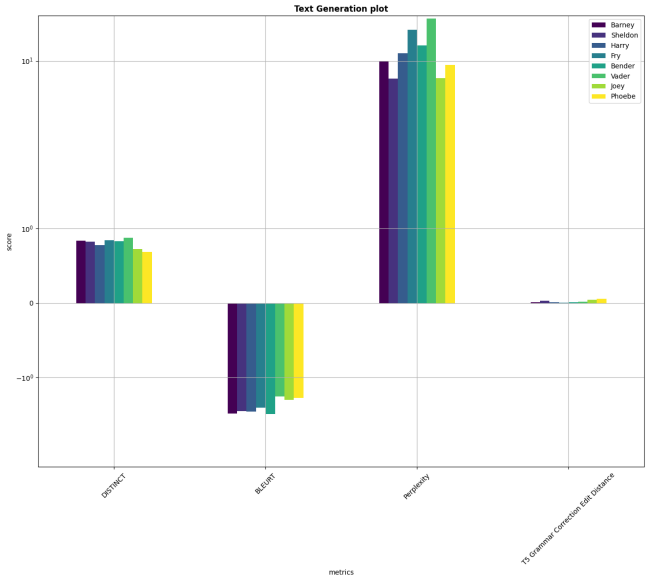
We now move onto the analysis of text generation metrics.

Distinct shows a high range of values, suggesting good general syntactic variety by our chatbots. Perplexity similarly provides a good sanity check to confirm that all our models have been trained properly. Differences in value between chatbots may also indicate more or less variability in responses.

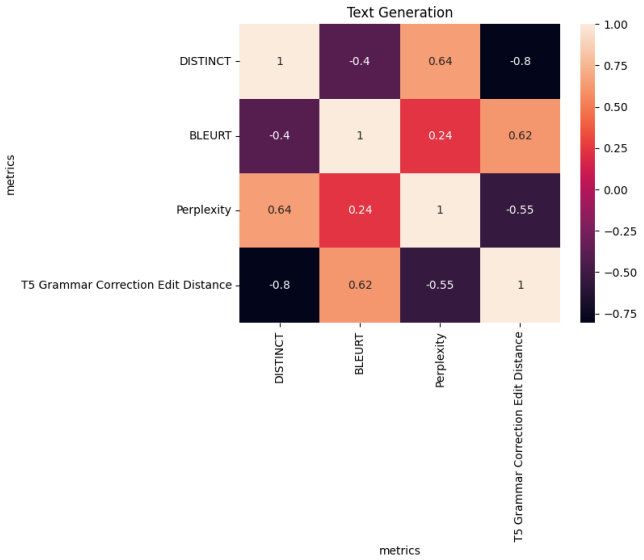
BLEURT is, unfortunately, plagued by negative values all around, even though the metric should reportedly output values between 0 and (approximately) 1. This result may be attributed to the short length of responses, as the model, tuned for longform text generation, may have learned to attribute a large penalty in such cases. We believe this metric has still a lot of potentials and should be further investigated after proper fine-tuning of the model. It is also worth noting that the only other metric that has also produced negative results is COMET; this metric also compares the language model of the reference text with that of the chatbots sentences through a neural network, indicating that the negative results from both metrics may have a common underlying cause.

For what concern the T5 Grammar Correction Edit Distance, results demonstrate good performance of the chatbots in generating sentences which are well-built under grammar. This is an important tool in the toolkit of an evaluation of text generation model, and in particular it is a trait we wish to keep whenever we perform personality transfer through fine-tuning.

From the correlation matrix, there is not so much relevant information that can be extrapolated. This is somewhat predictable, given that each of the metrics is a different model which aims to tackle the task of evaluating text generation from very different viewpoints.



(a)

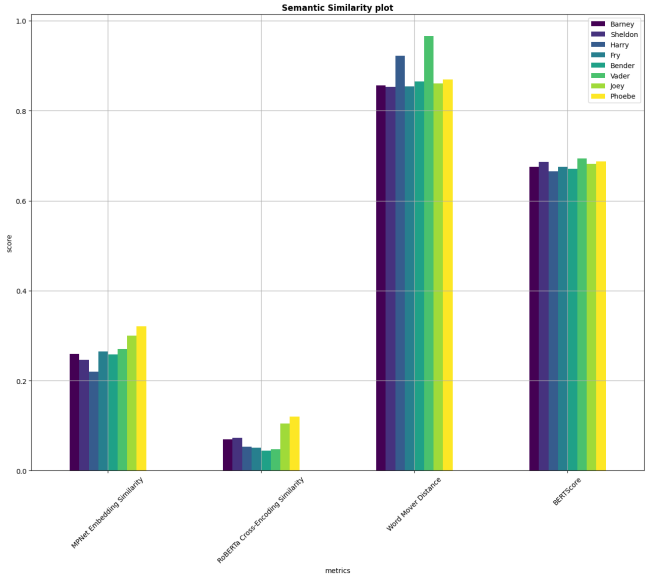


(b)

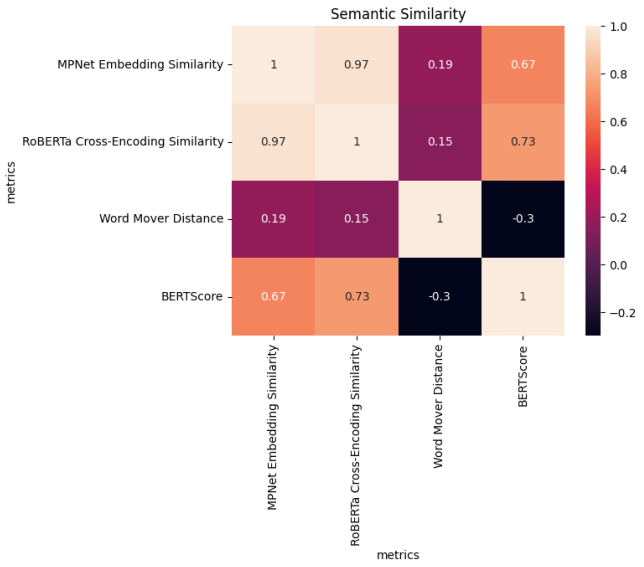
Figure 2: Scores and correlation matrix for Text Generation metrics.

4.3 Semantic Similarity Metrics

This set of metrics is arguably the most representative of conversational agents, due to the fact that none of them attempt to find a perfect match, but rather a likeness between context and (chatbot) response. This line of reasoning is supported by our results: the highest metric values are obtained by BERTscore, as it computes pairwise cosine similarity between embedded tokens of the sentences. When we focus more on the sentence level analysis, with MPNet or with the semantic answer similarity, values decrease considerably; the low scores from the semantic answer similarity are probably influenced also by the application of this metric to a context different than question answering. This is the kind of behavior we would expect as the responses to a given context can be many, with many layers of semantical difference, which become more apparent as we move higher-level in our evaluation. Taken together, these metrics therefore seem to provide some insight into the overall behaviour of the chatbot; further work in this direction could be very interesting, both comaparative and regarding novel approaches, albeit it is limited by the lack of algorithmic approaches. Word Mover Distance is a potential candidate, but it is hardly correlated with the other metrics; it does tend to edge on the higher end of values, showing a certain (possibly desirable) distance between context and response.



(a)



(b)

Figure 3: Scores and correlation matrix for Semantic Similarity metrics.

4.4 Classification Metrics

We finally move onto our novel, comparison-based approaches for the evaluation of chatbots.

Emotion Classifier Results on emotion analysis were, unfortunately, disappointing: although single sentences can be classified more or less correctly, their average quickly degrades to pretty much the same output. We have found no significant differences with respect to the dataset taken under consideration; this can be seen in Figure 4, which shows the radar plots for all characters under 6 different emotions as classified by DistilBERT.

No deviation from this behavior was obtained by using a different classifier (the HuggingFace model `maxpe/twitter-roberta-base-jun2022_sem_eval_2018_task_1` on 11 emotions), by thresholding or polarizing the outputs, or by considering the sentence-level counts of the strongest emotion.

Most damning, results are invariant even when considering the classification on the testsets alone, and even with a very small amount of sentences (the testset for Vader has only 16 sentences!). Therefore, it is safe to assume that there is either an innate bias in both classifiers (and corresponding datasets) or that the task in itself is biased.

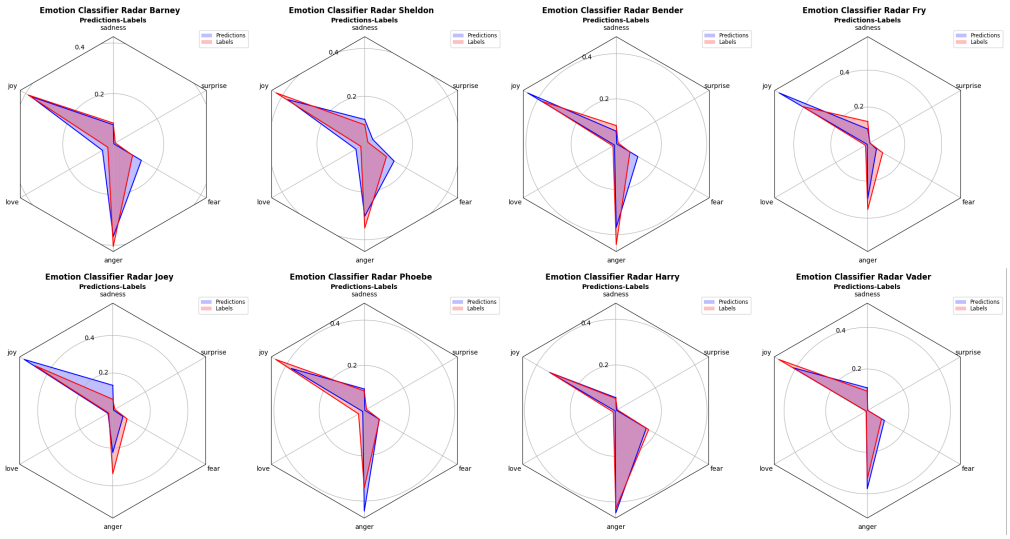


Figure 4: Emotions on all characters based on Plutchick's wheel of emotions [15], using as classifier the `bhadresh-savani/distilbert-base-uncased-emotion` HuggingFace model. The red area refers to the sentence-mean emotions over the character responses in the testset, while the blue area is with respect to the corresponding chatbot responses. All plots look extremely similar, even when comparing labels only, therefore they yield no significant information.

DistilBERT-Embedded Classifier The first test we did for the DistilBERT classifier were focused on the best choice of hyperparameters, that is, dimensionality of the embedding and the number of sentences to take in input: Table 1 and Table 2 shows reasonable results: embedding efficacy has a peak, beyond which the model is worse due to the curse of dimensionality, while the more the input sentences the better, generally.

embedding dim	2	4	8	16	32	64	128	256	512
accuracy (%)	76.64	84.62	74.62	78.62	67.13	68.5	63.38	39.25	29.66

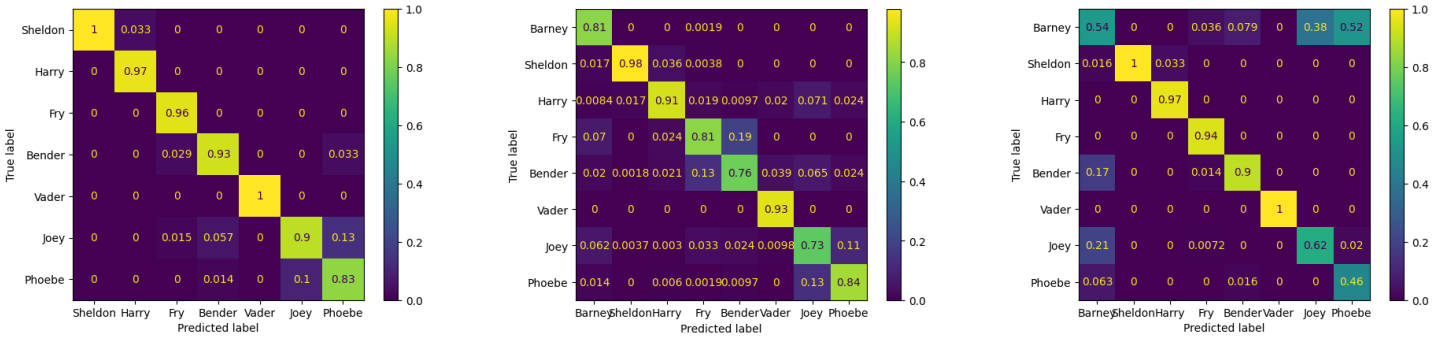
Table 1: Average accuracy over all characters changing the embedding dimensionality

input sentences	1	3	5	7	9	11	13	15	17	19
accuracy (%)	13.63	41.31	62.38	67.38	75.63	71.63	71.63	70.25	64.75	79.5

Table 2: Average accuracy over all characters changing the number of input sentences

The first test is done with an arbitrary value of 5 input sentences and for the second test is has been selected the embedding dimension with the best score. Based on the results, the final model has embedding dimension of 4 and 9 input sentences; technically, the best possible result corresponds to a number of input sentences of 19, but the model was also slower so the second-best value was preferred.

We have then moved to the results obtained by applying the model to classification on our characters. Image 5 shows the resulting confusion matrices.



(a) (b) (c)

Figure 5: (a) Confusion matrix from scripts testset; (b) Confusion matrix from chatbot responses; (c) Confusion matrix from script testset, where the embedder had no training knowledge of Barney.

The first test was performed over the testsets, the second over the chatbot responses. The confusion matrix w.r.t chatbot responses shows lower performances, as expected, and it also has higher difficulty in distinguishing characters from the same tv show, which is an interesting characteristics of these chatbots which is properly spotted by the classifier. Furthermore, as anticipated at the end of the paragraph on DistilBERT description 3.2.4, this model, theoretically, should be able to correctly classify characters also not present in the training set of the embedder. To test this assumption we re-trained the same architecture, removing the character Barney from the training of the embedder (but not from the training of the KNN classifier). From the confusion matrix, it is evident the drop in performances with respect to the other cases, but the results are still much better than random.

Frequency Classifier We performed two groups of tests using our frequency classifier, which we will refer as:

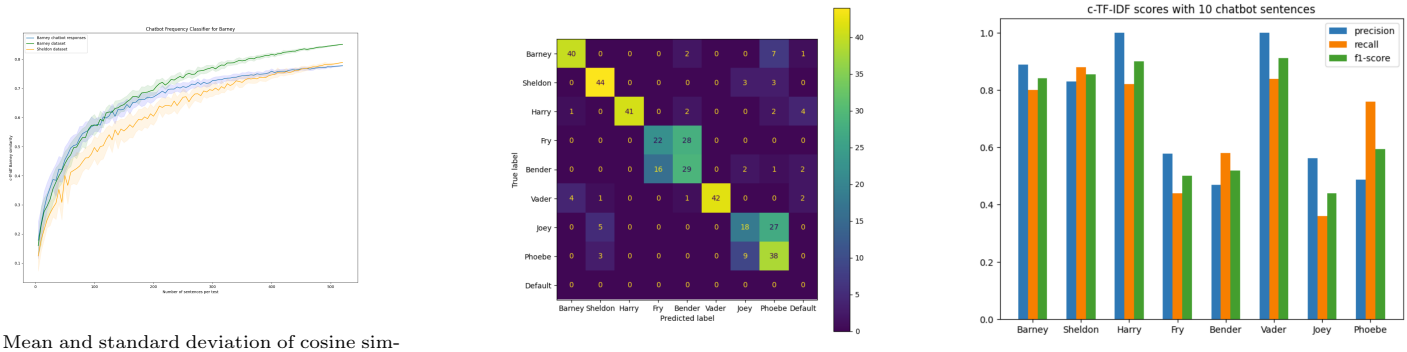
- T_L: compute the c-TF-IDF against labels;
- T_C: compute the c-TF-IDF against the chatbot responses;

For both types of tests, each sample is constituted by a set of 3 or 10 sentences.

	$T_{L,3}$	$T_{L,10}$	$T_{C,3}$	$T_{C,10}$
accuracy (%)	0.71	0.92	0.49	0.69
precision (%)	0.66	0.91	0.53	0.49
recall (%)	0.68	0.89	0.44	0.76
f1-score (%)	0.66	0.90	0.47	0.59

Table 3: Test results over the four types of tests.

Referring to Table 3, we notice in T_L overall high values, even with just three sentences, and a clear improvement when considering a larger set of sentences. This is not surprising: the higher the number of sentences to consider in the queries, the greater the information provided to the classifier to make a more informed prediction. Similarly, if we look at T_C, results generally improves when providing a larger set of sentences per sample. However, values are overall pretty low, showing a significant divide between labels and chatbot responses. This indicates that the tool is working quite nicely: albeit based solely on frequencies, it is capable of distinguishing the degree to which a set of responses matches an expected answer. We therefore suggest it may be interesting to consider the number of sentences required to have a relatively confident prediction by our frequency classifier, as a potential metric.



(a) Mean and standard deviation of cosine similarity of c-TF-IDF results, trained on Barney.

(b) Confusion matrix of $T_{B,10}$.

(c) Scores of each character in $T_{B,10}$

Figure 6: (a) Mean and standard deviation of cosine similarity of cTFIDF, trained on Barney, computed on other Barney labels, on Sheldon labels and on the Barney chatbot. Obviously, the similarity is higher when considering Barney w.r.t Sheldon chatbot, both for labels and chatbots. However, as the number of sentences increases, the Barney and Sheldon chatbots reach similar performances, a worrying sign of over-fitting. (b) The confusion matrix results in $T_{C,10}$ shows difficulties in predicting characters coming from the same tv series. (c) Scores of each character in $T_{C,10}$ show high accuracy over the board, except for characters coming from the same tv series.

Further observations can be made by looking at Figure 6, where we show the result of further tests. It seems that the metric has difficulty when considering too many sentences, as the cosine similarity graph shows: the Sheldon chatbot reaches a similarity comparable to that of the Barney chatbot when both are compared against the Barney dataset, as the number of sentences increases. This may either indicate a limitation of the metric (over-fitting), or the fact that it is starting to recognize a common baseline between the two chatbots, treating the fine-tuning differences as noise.

The confusion matrix in the same figure shows another issue with the metric: when two characters are taken from the same tv series, the metric struggles to distinguish them. In other words, it seems that the authors' style of writing has a greater impact on classification than the specifics of each character. Nonetheless, we can see a noticeable impact of fine-tuning, since the number of samples wrongly classified as "Default" is extremely low. Accuracy scores confirm this behaviour, as they are much lower between characters sharing their source.

4.5 10 Sentences Ranking

In order to test out the efficacy of these metrics in a practical setting, we perform a simple experiment, where we consider 10 possible responses of a chatbot to the same prompt, and compare how metrics compare to the human rankings. In order to correlate the rankings, we use the Kendall-Tau measure.

Figure 7 shows the results of this test on Barney, using as context the sentence "Barney, this is about the building". Albeit we imagine great variability in rank correlation due to the small number of sentences, one result is immediately quite clear: human rankings do not correlate well with any of the metrics we have proposed. We also find confirmation, on a very small set, of some correlations we have already highlighted all over Section 4.

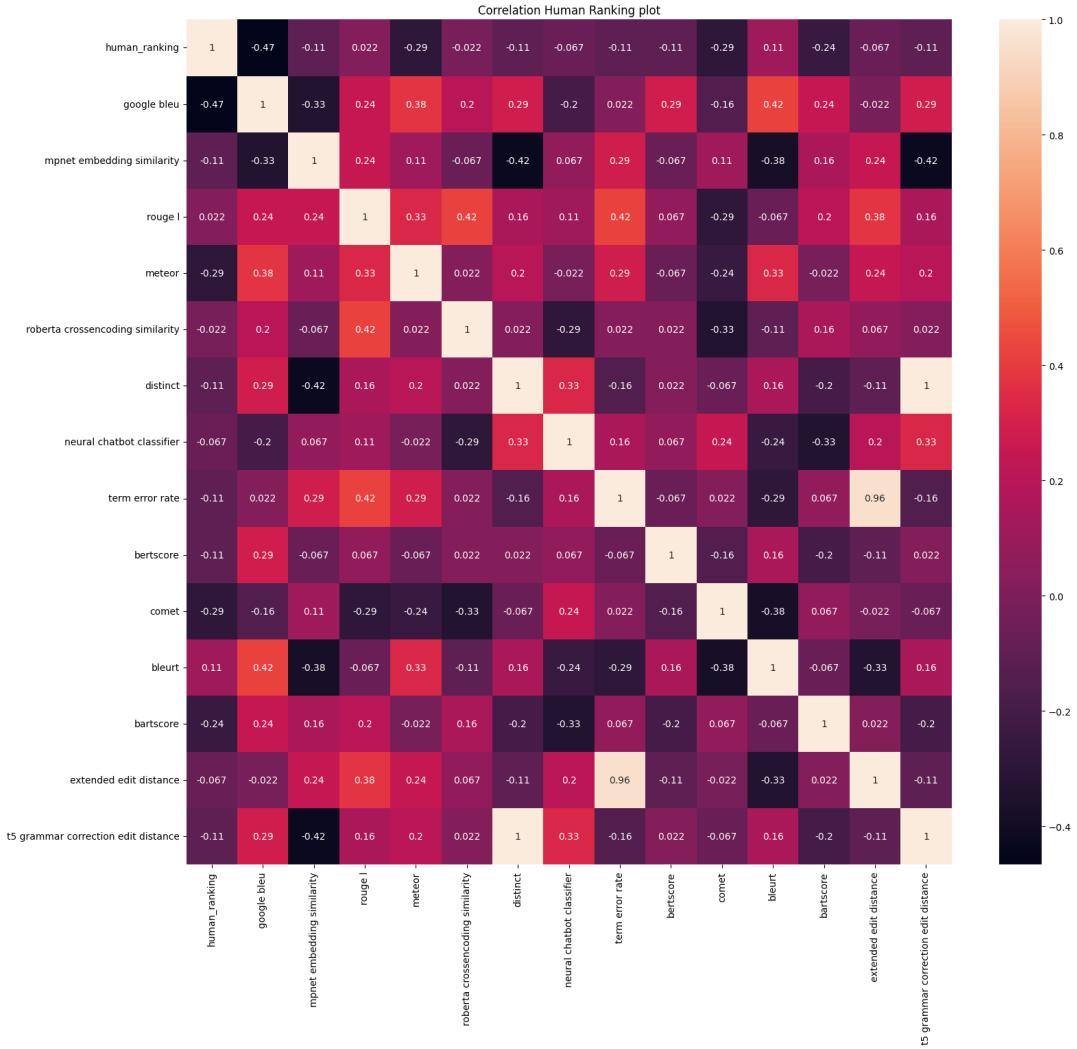


Figure 7: Confusion matrix of Kendall-Tau rank correlation between all metrics of our work, for 10 sentences sampled from Barney chatbot answering the same context "Barney, this is about the building."

5 Conclusions

We approached the chatbot evaluation problem from different perspectives, trying to build a set of metrics capable of analyzing the main aspects of language and conversation.

We began by looking at Machine Translation metrics, whose main utility could be to determine whether the chatbot has successfully captured the typical phrasing and catchphrases of the character it is trained to simulate; context may be also

used, as in COMET, to provide additional information, and it may be interesting to consider how context may be applied algorithmically to this task. However, MT metrics are too focused on syntactical similarity, even if the more complex metrics do consider additional text properties such as word order and synonymy. This is limiting, since we are expecting the chatbot to recall some iconic/representative expressions that are typical of the character (as it is the easiest way of identifying a specific character from a set of sentences) and *not* to answer exactly the same way of the scripts to a certain context.

Next, we looked at Text Generation, probably one of the most relevant tasks for what concerns a conversational agent. For this reason, we explored and tested this concept with different metrics. Performances of the base model of our chatbot revealed stable responses both in terms of capturing some aspect of the original personality, coming from the script text distribution, and of a generic language model according to common grammar rules. These are indeed two important aspects which should be persistent in a good language model capable of interacting with humans, in order to make it possible to let users believe they are really speaking with their favourite tv show character through proper, intelligible chats.

The Semantic Similarity metrics reinforce what was already observed with the machine translation metrics: the responses generated by the chatbot are significantly different from the reference scripts, although the BERTScore indicates that some relevant tokens are present in both the chatbot sentences and the reference scripts.

The fact that the frequency classifier is based on the weighted frequency of terms in documents, allows it to be explainable due to the absence of a black box, but it also introduces some obvious limitations for a metric which should evaluate the ability of chatbots to respond according to a given personality. Indeed this method doesn't take into account any notions about grammar, well-constructed sentences or even any context-response coherence. It may therefore be interesting to be paired up with other approaches, such as Distinct or the T5 Grammar Correction metrics.

In our opinion, the DistilBERT classifier remains one of the most interesting and promising metrics. Despite not achieving the best results in this work, it appears to be a fairly high-performing and particularly adaptable metric.

Finally, our test on sentence rankings proved that no single metric can correlate well with human judgment under arbitrary testing conditions. This further suggests that a combination of metrics may be best suited for the overall evaluation of a chatbot. A first, simple proposal for this set could be the following:

- Distinct and T5 Grammar Correction Edit Distance for low-level analysis, possibly other such metrics;
- Rouge-L/Google BLEU, with further investigations to correlate it with other MT metrics, for recognition of catch-phrases and common-say;
- BLEURT, appropriately fine-tuned, to evaluate a response as a text generation problem, possibly with a threshold to exclude shorter answers;
- A suite of similarity metrics to understand the semantics of the dialogue, working at different levels (e.g. BERTScore, MPNet Embedding Similarity, Semantic Answer Similarity);
- A comparative approach to spot/confirm improvements or differences between models, such as the frequency classifier or the DistilBERT-embedded classifier.

One glaring issue that is still open to debate is the evaluation of coherence over a longer dialogue. Also, many different approaches could be engineered to evaluate other aspects of speech not taken into consideration here: the possible future directions for this area of research are endless.

References

[1] 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.

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

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

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

[6] Giacomo Frisoni, Antonella Carbonaro, Gianluca Moro, Andrea Zammarchi, and Marco Avagnano. NLG-metricverse: An end-to-end library for evaluating natural language generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3465–3479, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[7] Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. A theoretical analysis of the repetition problem in text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12848–12856, May 2021.

[8] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[9] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[10] 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.

[11] 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.

[12] 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.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[14] 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.

[15] 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.

[16] 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.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[18] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*, 2020.

[19] Elayne Ruane, Abeba Birhane, and Anthony Ventresque. Conversational ai: Social and ethical considerations. In *AICS*, pages 104–115, 2019.

[20] 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.

[21] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

[22] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*, 2022.

[23] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, 2006.

[24] 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.

[25] Peter Stanchev, Weiyue Wang, and Hermann Ney. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 514–520, Florence, Italy, August 2019. Association for Computational Linguistics.

[26] Elior Sulem, Omri Abend, and Ari Rappoport. Bleu is not suitable for the evaluation of text simplification. *arXiv preprint arXiv:1810.05995*, 2018.

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

[28] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[29] Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. *Natural language processing with transformers.* " O'Reilly Media, Inc.", 2022.

[30] 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.

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

[32] 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.

[33] 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.

[34] Lingfei Wu, Ian EH Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J Witbrock. Word mover's embedding: From word2vec to document embedding. *arXiv preprint arXiv:1811.01713*, 2018.

[35] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

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

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

[38] 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.