Project Report: Conversational model

Bipul Mani Pokhrel

Abstract This project focuses on developing an advanced chatbot using Google's Text-to-Text Transfer Transformer (T5) model. The idea is to enhance the chatbot's ability to understand and generate human-like responses across various conversational scenarios using the T5 model. Key phases include fine-tuning the T5 model on a specialized dataset and evaluating the chatbot via various evaluation metrics and the loss. The goal is to create a chatbot, which is capable of exceling in response accuracy, contextual understanding, and user engagement, demonstrating the T5 model's potential as a conversational AI.

1 Introduction

In the realm of artificial intelligence, or to be precise deep learning the development of chatbots to play a significant role in the interaction between technology and human. Chatbots have become irreplaceable in various sectors, providing efficient customer service, personalized assistance, and playing a central role in automating communication processes. The evolution of chatbot technology has been significantly accelerated by advancements in Natural Language Processing (NLP), a domain where machines are trained to understand and respond to human language. Part of these advancements are through neural-based models. Thanks to the recent Transformer Architecture (Vaswani et al., 2023) Fig. 2, which introduced the concept of Self-Attention mechanism, models have been scaled up in the size, which results in rising performance on various benchmarks (GLUE, SuperGlue, ...).

We focus on Google's Text-to-Text Transfer Transformer (T5) (Raffel et al., 2023), which marks a transformative shift in NLP. The uniqueness in this model is that it stands out for its approach to handling language tasks, where all forms of text-based problems are treated as a text-to-text conversion process, unlike previous architecture such as BERT (Devlin et al., 2019). This simplification allows for a more unified and efficient method in tackling NLP tasks, making T5 an ideal choice for developing chatbots.

The primary objective of this project is to use the capabilities of the T5 model to develop a chatbot that is capable of capture a high degree of linguistic understanding and adaptability. The chatbot aims to not only respond accurately to user queries but also to maintain context and coherence over the course of a conversation.

To achieve this, the project involves fine-tuning the T5 model on a dataset (*3K Conversations Dataset for ChatBot*). This process involves training the model to recognize various patterns in human conversation, adapting its responses to suit different tones, and ensuring that it can handle unexpected queries with a degree of cleverness.

Moreover we will in the later sections provide a series of evaluations of the chatbot's performance. These evaluations are crucial in iteratively refining the chatbot, ensuring it meets the desired standards of functionality.

2 Background

The field of Natural Language Processing (NLP) has been one of the most progressing fields in the development of AI-driven communication tools. One of the cornerstones in the recent times are Chatbots such as ChatGPT form OpenAI. NLP encompasses a range of techniques and algorithms that enable machines to understand, interpret, and respond to human language. NLP had lot of stages. This ranges from rule-based systems to the current machine learning based system, each contributing to more and more sophisticated and human-like interactions in AI systems.

In the beginning chatbot development consisted of developing rule-based systems, where responses were generated based on a set of predefined rules. These chatbots, while innovative for their time, were limited in their ability to handle unstructured or complex queries as they lacked the flexibility and understanding necessary for more pronounced conversations.

The introduction of machine learning in NLP brought a significant shift. Machine learning models, especially those based on statistical methods, allowed chatbots to learn from large datasets, enabling them to respond more dynamically to a variety of inputs. However, these models still struggled with understanding context and maintaining coherence over longer conversations.

The emergence of deep learning further revolutionized NLP. Deep learning models, particularly those based on neural networks, offered improvements in language understanding and generation. These models could process and generate language in a way that was more aligned with human-like communication, offering greater flexibility and adaptability in responses.

Google's T5 model represents one of the more recent advancements. Its unique text-to-text approach, where every NLP task is treated as a conversion from one form of text to another, simplifies and unifies the process of handling diverse language tasks. Unlike traditional models that require different architectures for different tasks(e.g. BERT for Classification, NER, Question Asnwering), T5 uses a single model architecture to perform a variety of NLP tasks, making it highly versatile and efficient.

The application of the T5 model in chatbot technology is particularly promising. Its ability to understand context and generate coherent, contextually relevant responses presents a significant advancement in creating chatbots that can engage in meaningful and seamless conversations with users. This capability is crucial in environments where chatbots are expected to provide accurate information and maintain engagement.

3 Architecture of T5

The architecture of T5 is almost similar to that of the original Transformer model (Vaswani et al., 2023). Like

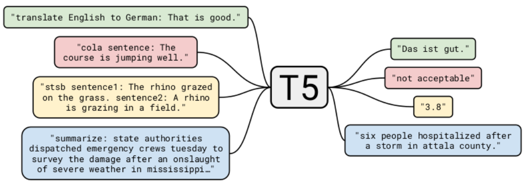


Figure 1: Overview of T5 (Raffel et al., 2023)

the Transformer model, T5 also consists of a stack of encoder and decoder blocks. T5 differs in two aspects from the transformer model. It uses relative positional encoding, in comparison to the sinusodial encoding in the original Transformer. The second difference is that, the instead the standard layer normalization, T5 only rescales the activations of the previous layer and doesn't add the bias.

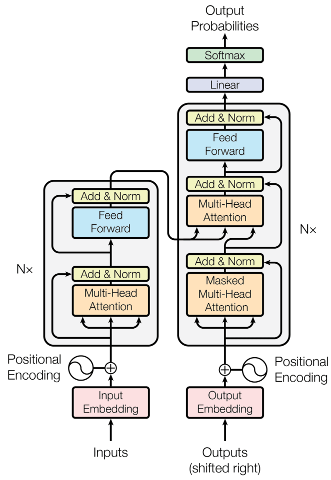


Figure 2: General Architecture of Transformer (Vaswani et al., 2023)

3.1 Pretraining

T5 is pretrained on the following datasets

- **C4** - 745GB
- **Unfiltered C4** - 6.1TB
- **RealNews-like** - 35GB
- **WebText-like** - 17GB
- **Wikipedia** - 16GB
- **Wikipedia + Toronto Books Corpus** - 20GB

with seven different auxiliary tasks, such as Masked Language Modelling for $2^{19} = 524,288$ steps. As the in-depth explanation of the pre-training procedure would be out of the scope of this report, we refer the reader to (Raffel et al., 2023).

4 Methodology

4.1 Dataset Preparation

Data Collection: We queried for some dataset in kaggle. Initially we were thinking on fine-tuning T5 on the AmbigQA [1] dataset. But as the answers on this dataset were rather short and the dataset itself was created to distinguish between ambiguity in questions,we chose the *3K Conversations Dataset for ChatBot*, which was more suited for our task. This dataset consists of 3724 conversations.

[1] https://nlp.cs.washington.edu/ambigqa/

Data Preprocessing: We used the opensource transformers library for tokenization, which comes with a predefined T5Tokenizaiton class and relevant methods.

4.2 Fine-Tuning

T5 Model Selection: We chose the T5-small (60M) and the T5 base (220M) variant for its balance between computational efficiency and performance.[2]
Parameter Tuning: The details of the Hyperparameter can be found in the appendix section.
Training and Validation: The T5 model was trained on the prepared dataset, with periodic validation checks to monitor its performance and prevent overfitting.

4.3 Evaluation

We use two different approaches to evaluate our model. The first method deals with comparing how similar the sentences are in terms of contextual embeddings. We convert the output response and the target sentence by first converting both into embeddings using the sentence transformers[3]. We later apply the cosine similarity, where the range goes from -1 to 1. Here -1 implies that the two vectors are negatively correlated and +1 implies that the vectors are positively correlated.

$$cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

We additionally use the ROUGE score to evaluate our model. ROUGE score is often used for language modeling tasks, such as summarization and machine translation. Rouge score can be more formally defined with the concepts of *PRECISION*, *RECALL* and *F1-SCORE*.

$$PRECISION = \frac{Overlapping\ number\ of\ n\text{-}grams}{Number\ of\ n\text{-}grams\ in\ the\ candidate}$$

$$RECALL = \frac{Overlapping\ number\ of\ n\text{-}grams}{Number\ of\ n\text{-}grams\ in\ the\ reference}$$

$$F1-SCORE = \frac{2 \times PRECISION \times RECALL}{PRECISION + RECALL}$$

One thing to note is that we represent *PRECISION* and *RECALL* with help of n-grams instead of *TP, FP, TN, FN*. We focus here on the harmonic mean (F1-SCORE), as this gives us a more balanced perspective between the PRECISION and RECALL.

[2] We couldn't test the T5-large(770M) variant as we ran into memory issues using it.
[3] https://www.sbert.net/docs/usage/semantic_textual_similarity.html

5 Results

We have plotted the training and validation loss for the 10 epochs in the following figures.
The results of the average cosine can be seen on Table

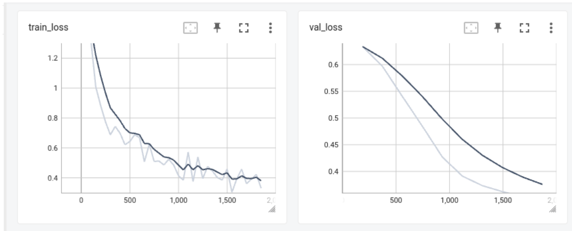


Figure 3: Training Validation Loss during Fine-tuning of T5-small

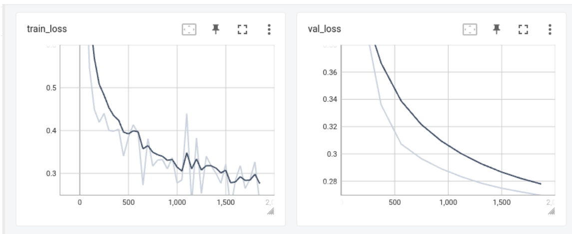


Figure 4: Training Validation Loss during Fine-tuning of T5-base

1. We can see that as we increase the size of the model, the average cosine value increases, which means that the prediction vectors and the target vectors get similar. Note that the scaling is not from 0 to 1, but from -1 to 1.
For the ROUGE score we see F1-Score on Table 2. Note that a value close to 0 indicates poor performance and a value close to 1 a relative good performance.

Model	Average-Cosine
T5-Small	0.22
T5-Base	0.27

Table 1: Average Cosine values of Test Set

	ROUGE 1	ROUGE 2	ROUGE L
T5-Small	0.11946	0.02652	0.11528
T5-Base	0.09148	0.01372	0.08790

Table 2: F1-Score according to ROUGE

6 Discussion

Although the average cosine similarity is slightly positive, there is still a space for improvement. This project can be thought of as a baseline. One should also be aware that we trained the model on a rather small dataset. If the memory constraints is not a problem one could train a bigger vairant of T5.

6.1 Challenges and Limitations

Handling Ambiguity and Complex Queries: The chatbot's performance in handling ambiguous and complex queries was less robust. This points to a need for further research in improving NLP models' comprehension of intricate and nuanced language.

6.2 Future Directions

Enhancing Error Handling: Future work should focus on enhancing error detection and handling mechanisms, particularly for complex and technical queries.
Expanding Domain-Specific Training: Tailoring the T5 model with domain-specific training could further improve its performance, especially in specialized fields.
Exploring User Personalization: Investigating ways to personalize interactions based on user history and preferences could elevate the chatbot's utility and user experience.
One particular problem was that we used open source data for this project, which had limited data points and our model was only able to hold a basic conversation. Future work should use a more nuanced dataset, or develop own dataset via crowdsourcing or other means.

References

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Appendix A

Hardware

`GPU: NVIDIA Tesla V100 SXM2 16 GB`

Software and Packages

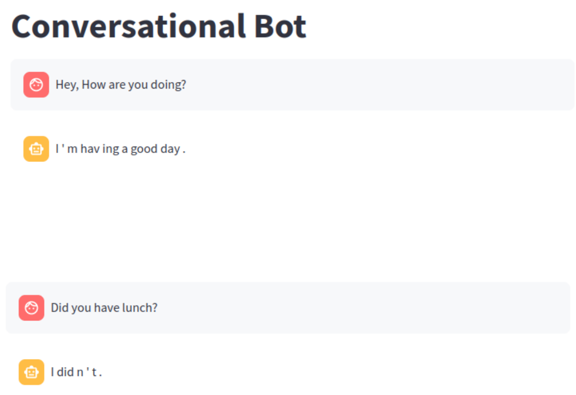
Operating System: The project was developed on a Rocky Linux 8.9 (Green Obsidian) environment.
Programming Language: Python 3.8 was chosen for its extensive support and libraries in data science and machine learning. We also used bash scripts to automatise some training loops.
Machine Learning Frameworks: We used PyTorch, PyTorch Lightning and the transformers library. Furthermore libraries such as pandas were also used. We also used jupyter-notebooks to run some tests.
T5 Model Implementation: The T5 model was implemented using the Transformers library by Hugging Face
Version Control and Collaboration Tools: Github[4]
API: The code that we published on Github also contains a simple UI, with which user can interact with the model. Some results are shown below in the subsection API.

Hyperparameter Settings

We used the `AdamW` optimizer with `linear scheduled warmup`.
`Training batch size: 16`
`Eval batch size: 8`
`Number of epochs: 10`
`Global seed: 100`
`Learning rate: 3e-5`
`Num Workers: 2`

API

In the following we listed two conversations with the model.



[4] The code can be founf here: https://github.com/eSharpMinor/T5ForConversation