Building a Chat-bot Based on Text-to-Text Transformer

Anonymous Author(s)

ABSTRACT

This report presented how to build a chatbot using T5 model. T5, a text-to-text deep learning model, had a excellent performance on natural language understanding and generation, but was also highly GPU-consumed. Transfer learning was used to help to overcome the short of GPU resources and fine-tune a pre-trained model in our project.

After being fine-tuned on given datasets, our chatbot was able to answer some commonsense problems, such as *'When did Minecraft season 2 episode 2 release for PC, Mac, PS4, Xbox 360, Xbox One, iOS, and Windows 10?'*, *'When is DJ Khaled's album Grateful coming out?'* and so on.

KEYWORDS

Chat-bot, Text-to-Text, Transformer, Natural Language Processing, T5

1 MOTIVATION

We would like to build a chatbot that is able to answer commonsense questions. A chatbot is defined as a conversational model which has communication ability to interact with human [12]. It can be used in areas such as customer service, social media, payment management, sending information, making appointments and so on. Currently existing chatbots (like Elizabot [19], Mitsuku [9], Cleverbot [18] and Watson [6]) are already shown to reduce labor costs and provide rapid feedback for all requests, which brings much benefit to human life.

A literature survey was conducted and it was found that models based on the Transformer's encoder [5] and decoder [13] achieved state-of-the-art results in many natural language processing (NLP) tasks, which demonstrated their powerful language understanding and generating abilities accordingly. T5 (text-to-text transfer transformer [14]) is a general generation model which is pre-trained on C4 (Colossal Clean Crawled Corpus [14]) and is designed to be flexible enough to be fine-tuned on downstream tasks. This meets our need and inspires us to explore whether the pre-train and fine-tune processes can also be used to build a customized chatbot. T5 will be the core model of our project.

Several candidate datasets were collected. After consideration in terms of their quantity, format and domain , AmbigQA [11] and WikiQA [20] were picked as our datasets to fine-tune our chatbot model. Together these datasets provide around 18,000 well-formatted question answer pairs that are open-domain. A question in the datasets will be something like "Who plays the doctor in dexter season 1?", and the answer will be "Tony Goldwyn" or "Goldwyn". The questions and answers will be fed into the encoder and decoder separately, but the encoder will be trained in an auto-encoder manner while the decoder will be trained in an auto-regressive manner. After fine-tuning, the text-to-text model will be able to generate answers based on training data.

We are dividing this project into three major tasks: data cleaning and preparation, model building, and application building. We will first integrate the datasets into one unified dataset as our input to the model. This task involves designing a format for the data that suits our needs and also filtering out the corrupted data. After that we will begin building the model using the method mentioned above. Notice that we are expecting this step to be computationally expensive. A mitigation plan would be to pick a smaller subset of the overall dataset that is focused on a specific domain and train the model on it. Once our model is functioning correctly, we will create a website to demonstrate it's capabilities. A user interface will be provided so that questions can be typed. These questions will be sent to the backend, where our model functions, and return the answer to the frontend. This will provide excellent interactivity and perfectly demonstrate the capability of our model. We saw this as an important step, since models similar to ours are hard to run and demonstrate on different machines; However, the accessibility of a website is nearly universal.

This report is organized as follows. In section 2, others' work in this area will be discussed. In section 3, we will present the design and implementation of the chatbot model, explain the functions of each module. In section 4, analysis of the project will be presented. In section 5, we discuss the legal considerations of the issues relevant to our project. In section 6, we discuss the ethical considerations of the issues relevant to our project. In section 7, we will present the current state of the project, possible future work, and make a conclusion.

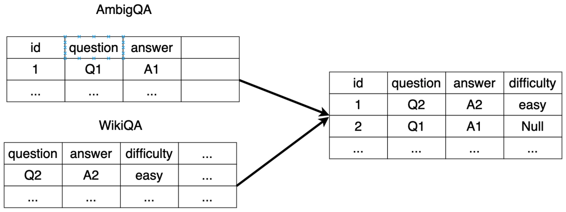
2 RELATED WORK

After transformer [17] was proposed by Google, this kind of deep learning architecture has been widely used in natural language processing tasks such as text classification, machine translation, document summarization and question answering. Chatbots are widely used because they can increase users' experience when interacting on applications and websites. Also, chatbots decrease labor cost and increase the feedback efficiency.

Meena [1] is a multi-turn open-domain chatbot based on the Evolved Transformer and trained end-to-end on conversation data. In [2], an online education tool with continuous interaction and personalization using bi-directional encoder representations from Transformers (Bert) was proposed. In its architecture, Bert combines the question generator module, the answer recording module, the answer analysis module and the conversational flow control module to increase the database of questions and responses. In [4], a Transformer-based mental health chatbot with topic extraction module was proposed.

3 DESIGN AND IMPLEMENTATION

The first step of our project is to prepare the datasets. As mentioned above, we have selected two datasets for our model. At this stage we are only using the AmbigQA dataset to build a primitive prototype for testing and verification purposes. In phase 3, we will attempt



AmbigQA

id	question	answer	
1	Q1	A1	
...	...	...	

WikiQA

question	answer	difficulty	...
Q2	A2	easy	...
...	...	...	...

id	question	answer	difficulty
1	Q2	A2	easy
2	Q1	A1	Null
...	...	...	...

Figure 1: Merging two datasets

to integrate both AmbigQA dataset and WikiQA dataset into one dataset to make the model more robust. Notice that these two datasets have different structures. We will select "question" and "answer" columns to build a unified dataset. WikiQA dataset also comes with a "difficulty" column which can be very useful to our performance analysis later, so it will be kept in the unified dataset. Since AmbigQA does not provide this information, we will put null value to the corresponding records.

The architecture of our chatbot is mainly made up of an encoder-decoder system like T5, which works entirely on the principle of self-attention and cross-attention. Compared with Recurrent Neural Network (RNN) and Convolutional Neural Network, this kind of model performs better in natural language processing tasks. Inside the model, encoders and decoders are stacked to increase the depth of the model, which can improve the performance of the model. Encoders will process self-attention and feed-forward inside themselves and output the hidden states of input data. Decoders will also first process self-attention on their inputs, but after that, the hidden state of input data will be fed into decoders to process cross-attention between encoders input and decoders input to determine which token should be generated. The tokenizer is used to encode the inputs into numerical data that model can process and decode the outputs of model decoders into English sentence. Content embedding module maps the numerical data into higher dimensional vector space to make the model more robust. Positional embedding module adds position information into the numerical data. Because self-attention process in encoders and decoders could not acquire any position information of text data, this module is very important for performance improvement.

Training process is shown in figure 2. During training stage, the inputs of the encoders are questions and the inputs of the decoders are answers. Decoders should output the answers auto-regressively, but because auto-regressively generation will be time-consuming, inputs of decoders will be processed into a triangle format like figure 3. In figure 3, we assume padding-token was 0 to make the figure easier to understand.

Inference process is shown in figure 4. During inference stage, the inputs of the encoders are the same as those in training stage. However, the inputs of the decoders will all be empty strings and the decoders will generate outputs auto-regressively. Auto-regressive generation means every token generates based on the tokens before it.

We implemented our model in Python using Pytorch. The training loss was calculated using cross-entropy between targets and decoders outputs. Because of the lack of GPU resources, we could only set the batch size small, but we overcame it by using accumulation gradients to update the parameters after using several batches of data. We fine-tuned our model on AmbigQA dataset to make it able to answer some commonsense questions. After 20 epochs, the loss dropped from 7 to 0.3.

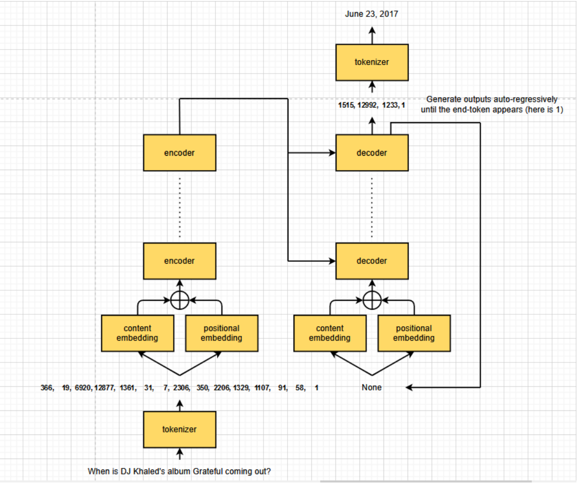
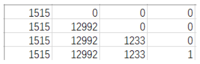


Figure 2: Diagram of training stage



1515	0	0	0
1515	12992	0	0
1515	12992	1233	0
1515	12992	1233	1

Figure 3: Inputs of decoders in training stage

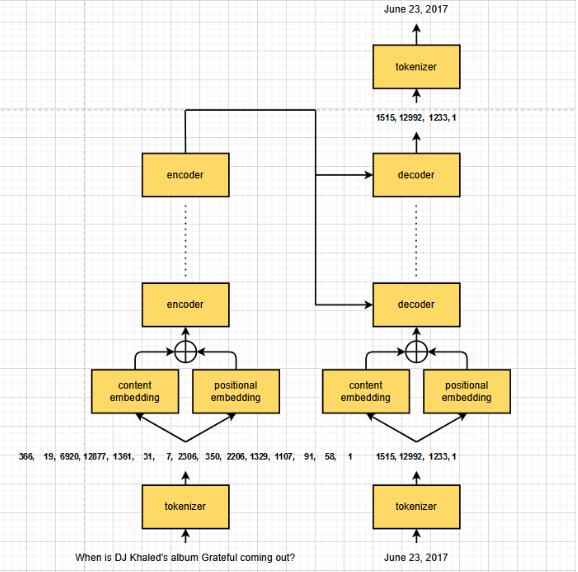


Figure 4: Diagram of inference stage

4 ANALYSIS

In this section we will demonstrate the performance of our model in both quantitative and descriptive manners and analyze the result to find room for potential future improvement. We will first define our metric for the evaluation of model performance, then proceed to give some examples of question-answer pairs, and finally analyze the performance in comparison with other currently available models for potential improvement.

As the quality of answers to questions is highly subjective, especially in the cases of questions that are open-ended, it is hard to find a universal standard that we can hold our answers against. Hence, we decided to appropriate the idea from the IBM question answering bot, Watson [6], to build a customized metric with our modification. For any question and answer chatbot, the key is whether the bot can provided useful and correct information for the question asked. Hence, the most important factor here is the information contained in the content. Answers that contain critical and useful information will have a higher score in this segment. Moreover, in the cases where the bot is unable to deliver a perfect answer for the question, an answer that is more relevant to the field being asked by the question is obviously more useful than an answer that is completely irrelevant. Therefore, our second factor is the relevance between the fields that the answer and the question are about. Finally, even the answer contains all the essential information needed for a question, due to the communication purpose of a chatbot, the answer should be well organized and structured in terms of both wording and grammar for the best experience of users. In conclusion, our customized metric contains of three factors: information quality, answer relevance, and answer organization, ranked by the priority. The final score should be a weighted sum of all three factors, where the scores of each factor will be manually graded. The formula below shows the calculation of this score.

$$S(a) = w_1 \cdot S_{qual}(a) + w_2 \cdot S_{rel}(a) + w_3 \cdot S_{org}(a)$$

where a is an generated answer, and $w_1 > w_2 > w_3$

Notice that this metric is customized to our own standard and is most suitable to compare our models that are trained with different settings to select the best result. For a lateral comparison with other published models, we will have to adopt their metric, such as the precision from IBM Watson [6]. Other than the evaluations mentioned above, we are also planning to use data analysis to dive deep into our model performance, such as analyzing which areas or types of question it answers best/worst, to help us get a more complete understanding of the capabilities of the T5 model and our training method.

Since our model is relatively primitive at this stage, we will not include or analyze its result as yet. A more detailed demonstration of its result, along with the analysis of its result, will be included in Phase 3.

5 LEGAL CONSIDERATIONS

As Chatbots become as ubiquitous as online companies, an increasing amount of laws and regulations must be kept in mind when releasing a bot to the public. A chatbot handles a variety of tasks human employees used to handle such as: answering questions, fixing accounts, handling payments, and processing complaints. The data and advice in these conversations can be highly sensitive and highly influential to the human on the other end of the conversation. With this much stake, there are new and existing laws to keep in mind.

Data is at the center of all Cognitive Computing projects and in a Chatbot project, data is constantly being entered and processed by the system. With the myriad of data privacy laws being enforced in law, this Chatbot project and others similar to it are not exempt from handling the data properly. Notably, hosted projects must be complaint to the Data Protection Act (DPA) and General Data Protection Regulation (GDPR) [16]. When processing a message from a human, the chatbot should not be actively storing and learning off of this data without human intervention. Circumstances such as telling a Chatbot a set of Personal Identifiable Information (PII) should not then allow the Chatbot to be queried for that PII. If an organization wants to record the conversations of the Chatbot, the company must include this in their End-user License Agreement (EULA) and provide a means to control and delete that information from the user who created it [8, 15].

This concept is important throughout the entirety of the data handling process with a Chatbot. User inputted data must be carefully taken care of and screened before being used to improve the bot. However, if the bot does reveal PII somehow or provides a disservice that leads to bodily harm or financial losses, is the Chatbot a legal entity that can be pursued in the court of law? Though the Chatbot functions similar to an employee in most use case, the Chatbot is not it's own entity, but instead is classified as a product in most regions. Therefore, the legal ramifications of a Chatbots disservice fall under acts similar to the Product Liability Act [7]. In these laws, the responsibility of the disservice falls to the manufacturer of the product, so the organization that created and hosts the Chatbot. With this in mind, our group and other organizations must be very careful releasing and hosting Chatbots onto the internet without a proper licensing and legal procedure in place.

There are also new Bot specific laws in place and being drafted that our group kept in mind. One such law is the California B.O.T. law [3]. This law states that an account not being operated as by a human (bot), must disclose that it is indeed not a human, but instead it is a bot. This project is open and discloses through the interface that the output is from a bot.

Lastly, our project is built on a variety of Open Source software systems and Public datasets. Each of these provides a license of it's own that our project must abide by. As stated, we have chosen all Open Source and flexible systems and datasets that allow us to combine their projects and data in a legal manner.

6 ETHICAL CONSIDERATIONS

In our project, ethical considerations are very similar to legal considerations. We will use the same structure to discuss the ethical considerations in our project as well as the general chatbot development.

The main pillars of the ACM Code of Ethics that we were concerned about were: 1.2 Avoid Harm, 1.3 Be Honest and Trustworthy, 1.4 Do not discriminate, 1.5 Respect Privacy, and 1.6 Honor Confidentiality. As seen, there are many points in which a Chatbot can ethically misbehave and as it's creators, we must take proper care to ensure that the Chatbot is ethical.

As discussed in Section 5, respecting the privacy and confidentiality of a conversation have already been thought out and implemented. However, pillars 1.2, 1.3, and 1.4 are still not ensured. However, knowing that the performance and responses of our Chatbot are directly influenced by the dataset we train it on give us great leverage over the Chatbots behavior.

The dataset should be full of relevant and ethical, that is, it should not contain information that may cause damage to people's lives or reputations. Examples of this would be private information such as patient's record or leaked data from an organization, of which chatbots should be clear. The bot can only know what it has learned and by boxing the bot into it's use case can be beneficial for ethical considerations even if that means the bot cannot understand and respond to any input properly.

These precautions are easier said then done and can only be ensured through testing and monitoring of the system. Even large organizations such as Microsoft have had ethical issues with online bots before. Take their bot named "Tay" for example [10]. Tay was trained and then given access to have have conversations over Twitter. In under twenty-four hours, the bot was tweeting racially insensitive text and incorrect facts stitched together by a small understanding of various topics. Most of these issues arose from the bot being trained on datasets from forums and conversations found throughout the internet, and these insensitive things were being said in the training set. This caused damage to the reputation of Microsoft and hurt to many online users simply trying to have a discussion with the bot. This prompted Microsoft to disable the bot and apologize for the entire experiment.

7 CONCLUSIONS

In this report we have provided details about our T5 model based chatbot. We started from a literature survey that confirms the state-of-the-art performance of T5. Then we demonstrated our design of the chatbot and provided details of our implementation including dataset selection and model fine-tuning. We also defined a customized metric for model performance evaluation and comparison. Legal and ethical considerations of this project were covered in discussion regarding both specifically our project as well as the chatbot development in general. Currently we have learned about the structure, training process, and capabilities of the T5 model. In phase 3 we will be focused on putting T5 into real application and analyze its performance for potential insights and improvements.

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. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977* (2020). https://arxiv.org/abs/2001.09977v3

[2] Richeeka Bathija, Pranav Agarwal, Rakshith Somanna, and G B Pallavi. 2020. Guided Interactive Learning through Chatbot using Bi-directional Encoder Representations from Transformers (BERT). In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. 82–87. https://doi.org/10.1109/ICIMIA48430.2020.9074905

[3] The People State Of California. 2018. Senate Bill No. 1001. (2018). https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB101

[4] Reuben Crasto, Lance Dias, Dominic Miranda, and Deepali Kayande. 2021. CareBot: A Mental Health ChatBot. In *2021 2nd International Conference for Emerging Technology (INCET)*. 1–5. https://doi.org/10.1109/INCET51464.2021.9456326

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine* 31, 3 (Jul. 2010), 59–79. https://doi.org/10.1609/aimag.v31i3.2303

[7] FindLaw. 2021. Product Liability Law. (2021). https://www.findlaw.com/injury/product-liability/product-liability-law.html

[8] Martin Hasal, Jana Nowaková, Khalifa Ahmed Saghair, Hussam Abdulla, Václav Snášel, and Lidia Ogiela. 2021. Chatbots: Security, privacy, data protection, and social aspects. *Concurrency and Computation: Practice and Experience* 33, 19 (2021), e6426. https://doi.org/10.1002/cpe.6426 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6426

[9] Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 928–939. https://aclanthology.org/C14-1088

[10] Peter Lee. 2016. Product Liability Law. (2016). https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/

[11] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *EMNLP*. https://nlp.cs.washington.edu/ambigqa/

[12] Mohammad Nuruzzaman and Omar Khadeer Hussain. 2018. A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. 54–61. https://doi.org/10.1109/ICEBE.2018.00019

[13] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[15] Andrea Trost. 2020. Chatbots – Key Legal Issues. (2020). https://www.mll-news.com/chatbots-key-legal-issues/?lang=en

[16] European Union. 2021. General Data Privacy Regulation. (2021). https://gdpr-info.eu

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010. https://dl.acm.org/doi/10.5555/3295222.3295349

[18] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *ArXiv* abs/1506.05869 (2015). https://arxiv.org/abs/1506.05869

[19] Joseph Weizenbaum. 1966. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. https://doi.org/10.1145/365153.365168

[20] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Lisbon, Portugal. https://www.microsoft.com/en-us/download/details.aspx?id=52419