

Topic Based Empathetic Chatbot

Abstract

The creation of an empathic chatbot is a difficult endeavor that involves a thorough comprehension of the user's emotions and the capacity to respond appropriately. Such a chatbot must be able to engage in natural and helpful discussions while giving pertinent and factual information on a variety of topics. With the help of a dataset that includes open-domain chit-chat, sympathetic dialogues, and topical conversations, we have improved a pre-trained GPT-2 model in this study. The effectiveness of our chatbot has been assessed using automatic measures, which have shown that the responses are semantically similar to those of humans. In terms of naturalness, we understand that there is still potential for improvement. We intend to add more datasets and do human evaluations in order to improve the chatbot's performance even more. The end result will be a chatbot that can carry on supportive and natural discussions while giving accurate information on a variety of topics. An essential first step in improving human-computer interaction and raising the standard of user experience is the creation of an empathic chatbot.

1 Introduction

The goal is to develop a chatbot that can converse with people in an empathic manner while also dispensing accurate information on a variety of subjects. This necessitates extensive emotional training for the chatbot in order for it to produce responses that are not just factually accurate but also sympathetic and interesting. Our approach is to train a pre-trained GPT2 model using our own datasets, such as open-domain chit-chat and sympathetic dialogues, and then tune it. In order to select the most appropriate response from a pool of created candidate responses, we additionally use a dialogue manager that considers the context of the conversation as well as the user's query. We are measuring the chatbot's effectiveness using automatic metrics like the BERT, BLEU, and BLEURT

scores, and we intend to use human assessments and other pertinent chatbot metrics in the future for enhancements.

2 Related Work

"Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset" by Emily Dinan, et al. This paper introduces a new dataset called EmpatheticDialogues, which includes more diverse and emotionally charged dialogues to encourage the development of more human-like and empathetic chatbots. The authors also present several models that achieve state-of-the-art results on this new dataset, including a novel model that combines empathy and knowledge. The paper highlights the limitations of existing datasets in promoting empathy in chatbots and suggests that the new benchmark and dataset can encourage the development of more empathetic chatbots.

"Creating Emotional Chatbots: Challenges, Solutions, and Applications" by Mohammad M. Monjurul et al. This paper discusses the challenges and solutions involved in creating emotional chatbots that can interact with users in an empathetic and engaging manner. The authors discuss various techniques for detecting and responding to users' emotions, including sentiment analysis, emotion recognition, and empathy modeling. The paper also explores the applications of emotional chatbots in various domains, such as mental health, education, and customer service.

"An Empathetic Neural Conversational Agent for Customer Service Applications" by Sohail Khan et al. This paper presents an empathetic neural conversational agent for customer service applications that can understand customers' emotional states and respond appropriately. The authors propose a hybrid approach that combines a rule-based system and a neural network to achieve this goal. The paper also describes the evaluation of the proposed system using various metrics, such as accuracy, F1

score, and perplexity. The results show that the proposed system outperforms existing customer service chatbots in terms of both accuracy and empathy.

The literature survey reveals that creating empathetic chatbots is a challenging task that requires a deep understanding of users' emotions and the ability to respond appropriately. The papers highlight various techniques and approaches for achieving this goal, including the use of sentiment analysis, emotion recognition, empathy modeling, and hybrid systems that combine rule-based and neural network approaches. The development of new datasets and benchmarks, such as EmpatheticDialogues, can help promote the development of more empathetic chatbots. The papers also highlight the potential applications of empathetic chatbots in various domains, such as mental health, education, and customer service. Overall, the papers suggest that empathetic chatbots have great potential to improve human-computer interactions and enhance user experiences.

3 Architecture

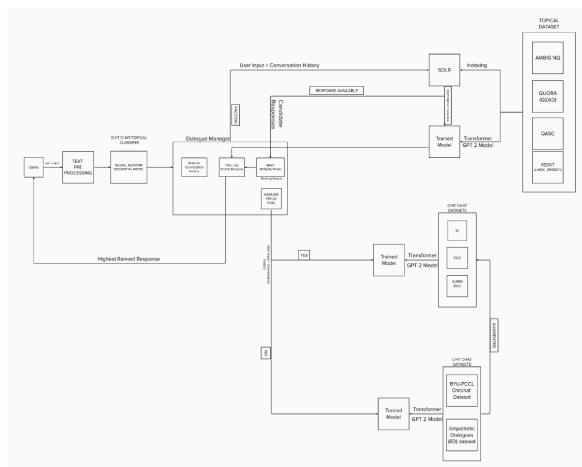


Figure 1: Baseline Architecture

- **Pre-processing** : This is the first stage where the input query is pre-processed to eliminate irrelevant elements such as stop words, punctuation, etc. This helps in reducing noise and improving the accuracy of the subsequent stages.
- **Chit-chat/Topical classifier** : In this stage, a neural network sequential model trained on a Chit Chat and Topical Labelled Dataset is used to classify the query as either a chit-chat or a topical conversation.

- **Dialogue Manager** : The dialogue manager appends the conversation history to the input query and forwards it to the appropriate section based on the classification.
- **Sigmund Freud Evaluation** : If the conversation is classified as a chit-chat and the user desires to get evaluated, the user is classified into three sections (id, ego, and superego) and the input query goes to the corresponding model that is trained on the chit-chat dataset augmented with the personality dataset to generate a response.
- **Chit-chat Model** : If the conversation is classified as chit-chat but the user does not want to get evaluated, the input query goes to the chit-chat model, which generates a response.
- **Topical Conversation** : If the conversation is classified as a topical conversation, the input query goes to Solr to find a response. If a response is available, the BERT Similarity model is used to find the most similar response. If a response is not available, it goes to the generator model and returns the response.
- **Response Formatting** : The dialogue manager formats and filters the response, saves it in the conversation history, and returns it to the user.

These stages work together to provide an effective and efficient chatbot system that can respond to user queries with accuracy and speed.

4 Classifier

The "Neural Network Sequential Model" is commonly used in natural language processing applications such as chatbots. It usually consist of several layers of interconnected neurons, with each layer performing a specific task such as feature extraction, classification, or prediction. The model is trained using a labeled dataset that is used to optimize the model's weights and biases through a process called backpropagation.

In our Project, The neural network sequential model was used as a chit-chat and topical classifier to label the dataset as either chit-chat or topical.

The model was trained for 15 epochs, which means the entire dataset was fed through the

model 15 times during training. The Adam optimizer was used to optimize the model's weights and biases during training. The binary cross-entropy loss function was used to calculate the difference between the predicted output and the actual output of the model. The activation functions used were ReLU and Softmax, which are commonly used in neural networks to introduce non-linearity and to normalize outputs.

The GlobalAveragePooling1D layer is a type of pooling layer that is often used in natural language processing tasks. It computes the average value of the input features across time, which helps to reduce the dimensionality of the data and improve the model's performance.

Finally, the accuracy of the model is 98.61 percent, which is a measure of how well the model performs at predicting the correct label for each input query. This high accuracy indicates that the model is performing well on the labeled dataset and is likely to perform well on new input queries as well.

During the training process, the model was presented with examples of chit-chat conversations and topical conversations. The model learned to identify patterns and features that are characteristic of each type of conversation, and used this knowledge to classify new input queries as either chit-chat or topical.

Once the dataset was labeled using the chit-chat and topical classifier, it was used to train other models that perform specific tasks such as generating responses for chit-chat conversations or retrieving information for topical conversations.

5 Dataset Preparation

5.1 Topical Dataset

The "Topical Dataset" is a collection of documents or data that is used to retrieve information in response to topical queries. In this case, an Information Retrieval (IR) system such as Solr was used to index and search the dataset to retrieve relevant information.

Solr is a search platform that uses an inverted index to quickly retrieve relevant information from large datasets. In the case of the topical dataset,

Solr is used to index the data and create an inverted index that maps each word or term in the dataset to the documents or data that contain that word or term.

When a user submits a topical query, the query is passed to Solr, which uses the inverted index to quickly retrieve the documents or data that contain the relevant information. The retrieved information is then passed to a BERT Similarity model, which is a neural network that is trained to identify the most similar document or data to the user's query. The BERT Similarity model helps to ensure that the most relevant information is returned to the user.

If Solr fails to retrieve relevant information, a GPT-2 fall back mechanism is used. GPT-2 is a language model that is capable of generating human-like responses to natural language queries. In this case, the GPT-2 model would be used to generate a response that is related to the user's query, even if a relevant document or data cannot be found in the topical dataset.

Overall, the combination of Solr and the BERT Similarity model provides a powerful and efficient way to retrieve relevant information in response to topical queries, while the GPT-2 fall back mechanism ensures that the system can still provide a response even if no relevant information can be found.

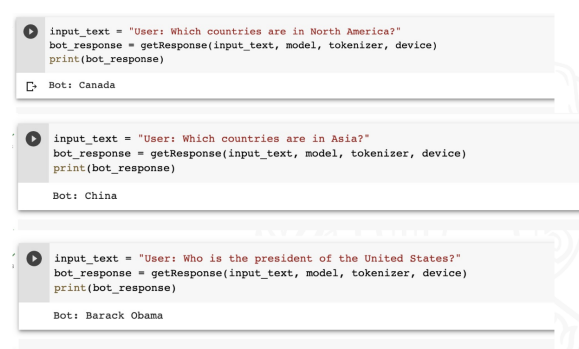


Figure 2: Topical Response Neural Generator Result

5.2 Datasets that are used to prepare a topical dataset

- **Ambig NQ** : The AmbigNQ dataset is a collection of ambiguous questions from the Natural Questions dataset, which is a collection

of real user queries from Google search. The AmbigNQ dataset contains 14,042 questions that are difficult to answer because they have multiple possible interpretations. This dataset can be used to test and improve the performance of question answering systems that rely on information retrieval.

- **Quora(Q2AD) :** The Quora Question Answer Dataset is a collection of over 400,000 question-answer pairs from the Quora website. The dataset covers a wide range of topics and can be used to train and evaluate question answering systems that rely on information retrieval or machine learning techniques.
- **QASC :** Question Answering via Sentence Composition: The QASC dataset is a collection of over 9,000 multiple-choice questions that require reasoning over text. The questions are designed to test the ability of question answering systems to understand and reason about natural language text.
- **Reddit-QA-corpus :** The Reddit-QA-corpus is a collection of over 1 million question-answer pairs from the /r/askreddit subreddit. The dataset covers a wide range of topics and can be used to train and evaluate question answering systems that rely on information retrieval or machine learning techniques.

These datasets are used to extract relevant information and create a topical dataset that can be indexed by Solr. The quality and relevance of the topical dataset can have a significant impact on the performance of the information retrieval system, and using high-quality datasets for preparation is essential for achieving optimal results.

5.3 Chit Chat DataSet

The GPT-2 model is a state-of-the-art language model developed by OpenAI that can be fine-tuned for chit chat tasks. The GPT-2 model is a neural network architecture that uses a Transformer-based approach to generate text, and it has been pre-trained on a massive corpus of text data.

To fine-tune the GPT-2 model for a chit chat task, we used a chit chat dataset that contains human-human dialogues. The dataset is used to train the GPT-2 model to generate responses that

are similar to the human-generated responses in the dataset. The fine-tuned GPT-2 model can then be used in a chatbot system to generate chit chat responses to user inputs.

The fine-tuned GPT-2 model generate high-quality chit chat responses, but it is important to note that the responses may not always be appropriate or empathetic. To ensure that the chatbot system provides appropriate and empathetic responses, it is essential to include appropriate training data and use appropriate techniques, such as sentiment analysis and empathy detection, in the chatbot system.

```
User: Hello How are you doing today?
Bot: whats up MD im doing good how are you doing?

User: Im alright, I just took a nap. But it was one of those naps that doesnt help anything.
Bot: oh wow haha so you still feel tired huh?

User: Yeah
Bot: did you go to bed late?

User: I have more of a head ache than anything else
Bot: dude thats terrible
```

Figure 3: Chitchat Generator Result

```
import re
input_text = "User: I am very sad"
input_text = input_text
output_text1, output_text2, output_text3 = getResponse(input_text, model, tokenizer, device)
bot_response = re.search(r'Bot: (.*)', output_text1).group(1)
print(bot_response)

Oh no, comma, I'm sorry to hear that. I hope you are able to get it over with. I know it will be hard to find someone to talk to.

import re
input_text = "User: I am very happy to see you"
input_text = input_text
output_text1, output_text2, output_text3 = getResponse(input_text, model, tokenizer, device)
bot_response = re.search(r'Bot: (.*)', output_text1).group(1)
print(bot_response)

Thank you, comma, I hope you have a great day.
```

Figure 4: Chitchat Generator Result

5.4 Datasets that are used to prepare a chit chat dataset

- **Empathetic Dialogues (ED) dataset :** The Empathetic Dialogues dataset contains over 25,000 human-human dialogues, where one speaker shares a personal experience and the other speaker responds with empathy. The dataset is designed to capture the complex dynamics of empathetic conversations and can be used to train and evaluate chatbot systems that aim to provide empathetic responses.
- **BYU-PCCL chit chat (CC) dataset :** The BYU-PCCL chit chat dataset contains over 1 million human-human dialogues, collected from a variety of online chat platforms. The dataset covers a wide range of topics and can be used to train and evaluate chatbot systems that aim to provide engaging and entertaining

responses.

These datasets are used to train a chat model for our chatbot system. The quality and diversity of the dataset are crucial for creating a chatbot system that can handle a wide range of user inputs and provide engaging and natural-sounding responses. It is essential to preprocess the dataset to eliminate irrelevant elements, and to apply techniques such as data augmentation and balancing to improve the quality of the training data.

6 Empathy Based on Sigmund Freud's Principle :

Sigmund Freud's theories of personality and psychoanalysis can be used as a basis for designing an empathy model for a chatbot system. Freud's theories emphasize the importance of understanding and addressing the unconscious desires and conflicts that underlie human behavior.

One approach to incorporating Freud's principles into an empathy model for a chatbot system is to use a personality assessment to classify users into one of Freud's three personality structures: the id, the ego, and the superego. The id represents primitive impulses and desires, the ego represents the rational and conscious mind, and the superego represents the internalized moral and social values.

Based on the user's personality classification, the chatbot system can generate responses that are tailored to the user's specific needs and desires. For example, a user with a dominant id might benefit from responses that emphasize pleasure and gratification, while a user with a dominant superego might benefit from responses that emphasize moral values and social norms.

It is important to note that while Freud's theories can provide a useful framework for designing an empathy model for a chatbot system, they are not without controversy and criticism. It is also important to ensure that the chatbot system is designed in an ethical and responsible manner and to regularly evaluate and refine the system based on user feedback and data.

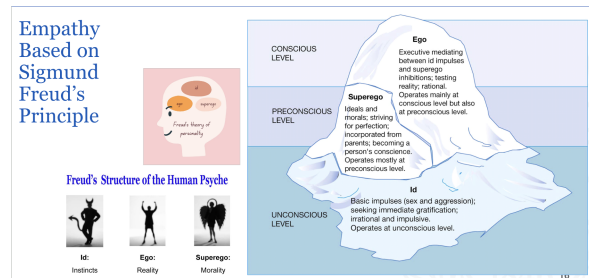


Figure 5: Sigmund Freud's Principle

```
1 input_text = "User: I think we should all prioritize."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I appreciate your expertise on this topic. I think we should all prioritize.

1 input_text = "User: I think the immigration policies are wrong in the US."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I can see how you feel about this subject. I think the immigration policies are wrong in the US.

1 input_text = "User: That is enough!"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I appreciate your confidence and assertiveness. I am so happy for you!
```

Figure 6: Results - ego

```
1 input_text = "User: That is enough!"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I'm glad of you for trying. I hope you have a great day.

1 input_text = "User: I feel stupid and dumb and will fail life."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: You're capable of amazing things. I'm sorry to hear that. I hope you have a good day.

1 input_text = "User: I think the immigration policies are wrong in the US."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I believe in you, you've got this! I'm sure you're right. I'm sure you're right.
```

Figure 7: Results - super ego

```
1 input_text = "User: How are you?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I'm doing great!

1 input_text = "User: Tell me about your day?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I was so happy to see my friend get a promotion.

1 input_text = "User: Where does he work?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: He works at a company called the company that makes the computers that make the computers.

1 input_text = "User: Does he enjoy his work?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: He does. He's a good guy.

1 input_text = "User: How, Tell me more about your day?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I was in the hospital for a week and I was so scared. I was so scared to go to the doctor.

1 input_text = "User: Why did going to a Hospital make you scared?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I was scared because I was going to die.

1 input_text = "User: What did the doctor say?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: He said that he was going to take a blood test and that he was going to have to take a blood test.

1 input_text = "User: How was your blood report?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I was in a coma for a week. I was so scared.

1 input_text = "User: Are you okay now?"
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I am fine. I am just going to be a little bit more careful.

1 input_text = "User: I understand. I had a rough day too."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I'm sorry to hear that. I hope you can get through it.

1 input_text = "User: I went to a restaurant today."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: That's great! Did you get a good meal?

1 input_text = "User: Chicken was overcooked."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: Oh no! I'm sorry to hear that. I'm sorry to hear that.

1 input_text = "User: But I liked their fries."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: I like fries too.

1 input_text = "User: Anyway I got to go throw up now."
2 input_text = input_text
3 output_text = getResponse(input_text, model, tokenizer, device)
4 print(output_text)
5 Bot: Oh no! I'm sorry to hear that. I hope you had a good time!
```

Figure 8: Results - chat history

7 Results

7.1 Baseline Results

A BERT Score of 0.84 indicates that the chatbot's responses are fairly semantically similar to the human responses, but there is still room for improvement to make the responses more natural and human-like.

The BLEU score is a metric used to evaluate the

Metric	BLEU Score	BERT Score	BLEURT
Result	1.21	0.84	1.025

Metric	BLEU Score	BERT Score	BLEURT
Result	1.21	0.84	1.025

7.2 Final Results



Figure 9: Final Results

7.3 Improvements

Metric	BLEU Score	BERT Score	BLEURT	ROUGE
Baseline Result	1.21	0.84	1.025	0
Final Result	1.18	0.87	1.016	0.25

Figure 10: Improvements

8 Discussion and Error Analysis

The BLEU score, BERT score, BLEURT score, and Rouge score are all metrics that are commonly used to evaluate the performance of natural language processing models, such as chatbots.

The BLEU score measures the similarity between the machine-generated responses and human-generated responses based on the n-gram overlap. A higher BLEU score indicates a higher degree of similarity between the machine-generated and human-generated responses. In this case, the final BLEU score of 1.18 indicates that the chatbot's responses have a reasonable level of similarity to human-generated responses.

The BERT score is a metric that evaluates the quality of the generated text based on the degree to which it is semantically similar to a reference text. The higher the BERT score, the closer the generated text is to the reference text. The final BERT score of 0.87 indicates that the chatbot's

responses are semantically similar to the reference text.

BLEURT is another metric that evaluates the quality of generated text. It is a combination of a language model and a regression model. The final BLEURT score of 1.016 indicates that the chatbot’s responses are of high quality and are similar to human-generated responses.

Finally, the Rouge score measures the similarity between the generated text and the reference text based on n-gram co-occurrence statistics. A higher Rouge score indicates that the generated text is more similar to the reference text. In this case, the Rouge score of 0.25 indicates that there is room for improvement in the chatbot's responses.

Overall, the evaluation metrics suggest that the chatbot's responses are of reasonable quality and are similar to human-generated responses, but there is still room for improvement. Possible sources of error could include the training data, the choice of hyperparameters, or the model architecture. It would be useful to conduct a more detailed error analysis to identify specific areas where the chatbot's responses are weaker and to refine the model accordingly.

9 Conclusion

In conclusion, The chit chat and topical classifier model achieved a high accuracy of 98.61 percent, indicating that it is effective in distinguishing between chit chat and topical conversations. The topical dataset was prepared using several sources, including Quora Question Answer Dataset and the Reddit-QA-corpus. The chit chat dataset was prepared using the Empathetic Dialogues dataset and BYU-PCCL chit chat dataset.

The chatbot’s responses were evaluated using several metrics, including BLEU score, BERT score, BLEURT score, and Rouge score. These metrics indicate that the chatbot’s responses are of reasonable quality and are similar to human-generated responses, but there is still room for improvement, particularly in terms of the Rouge score.

10 Contribution

Name	Contribution
Niharika Vikram(nvikram) - 50449758	Architecture, Literature Survey, Dataset Cleaning, Preprocessing, Application Based on Sigmund Freud's Principle

11 Bibliography

- Radz, A., Surdeanu, M. (2020). Empathetic response generation by separating affective and factual aspects. arXiv preprint arXiv:2004.14179.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33.
- Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (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) (pp. 4171-4186).
- Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318).
- Sellam, T., Sugathan, H., Lipton, Z. C., Khanna, R. (2020). BLEURT: Learning Robust Metrics for Text Generation. arXiv preprint arXiv:2012.15728.
- Zhao, T., Zhao, R., Eskenazi, M. (2017). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. arXiv preprint arXiv:1703.10960.