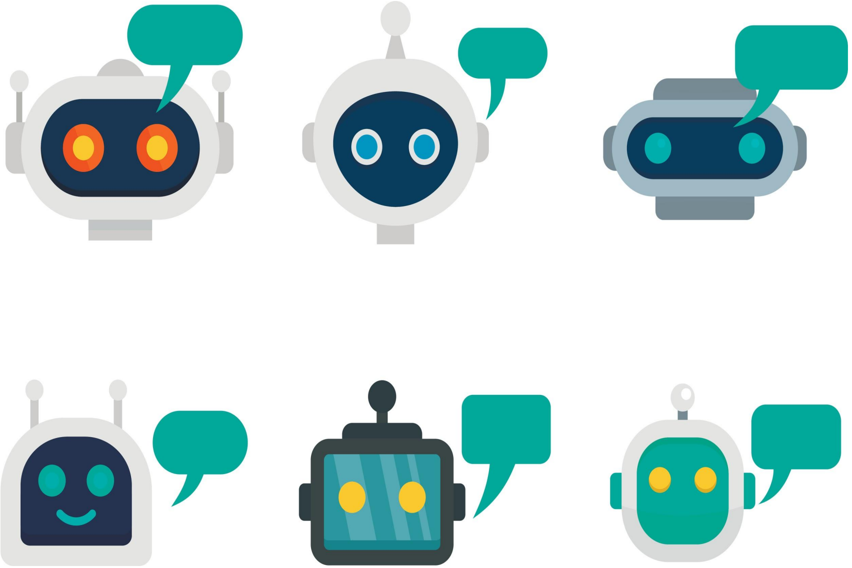


# Transformer ChatBot App

Dany Maron danygontowmaron@my.unt.edu

Daniel Mata danielmata2@my.unt.edu

Theo Medeiros theophilusmedeiros@my.unt.edu

YingHsuan Lo (Lori Schuan) vooloaa@gmail.com

# Abstract

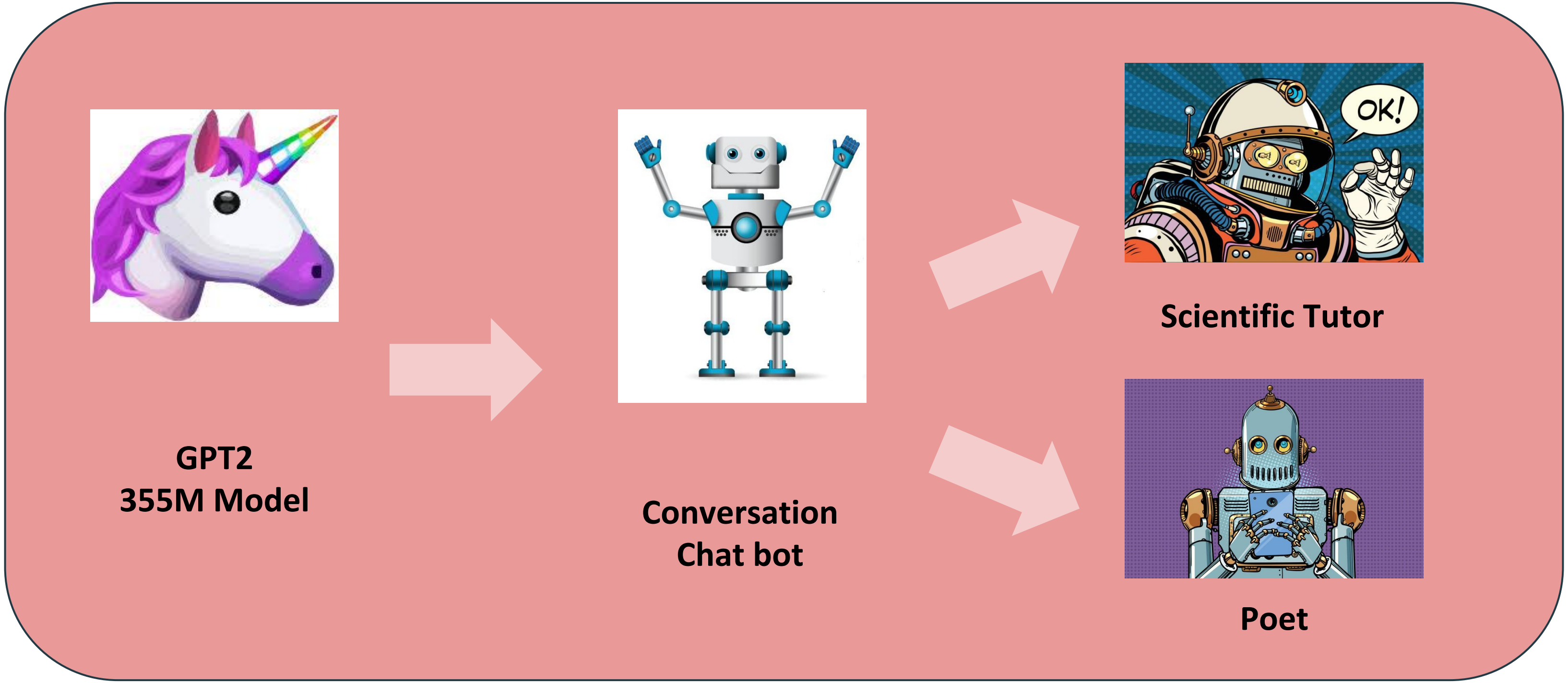
Chatbots have been around since the 1960s and have been used to conduct conversation between a human and a program via text or text-to-speech with little to no cost. Chatbots need constant tuning and testing to meet given standards and, as a result, various models have been developed. From simple web virtual assistants to sophisticated NLP and AI models, chatbots are widely required by different industries such as e-commerce, finance, healthcare, and more. For our project, we will be focusing on a Transformer.

A Transformer is a deep learning model that was introduced in 2017. Transformer models differ from RNNs in that they have self-attention hidden layers which calculate query, key, and value simultaneously whereas RNNs calculate sequentially. We will be training the pre-trained model with the dataset of our choice, which will then be connected to a front-end developed using Flask. In terms of direction, we wanted to create a life coach gpt-2 model. However, after speaking with Dr. Albert, we decided to create two versions of the chatbot: one would be a poem chatbot that would give a poem based on a topic the user entered, the other would be a scientist chatbot responding with scientific facts.

# Tech & Datasets:

- Google colab
- GPT2-simple
- Python 3.6
- Tensorflow 1.15
- Datasets:
  - Question Answer Datasets(Wikipedia articles) for Academic Use
  - Quotables
  - Sample conversations to generate insights
  - rDany Chat
  - Ubuntu Dialogue Corpus

# Project Design



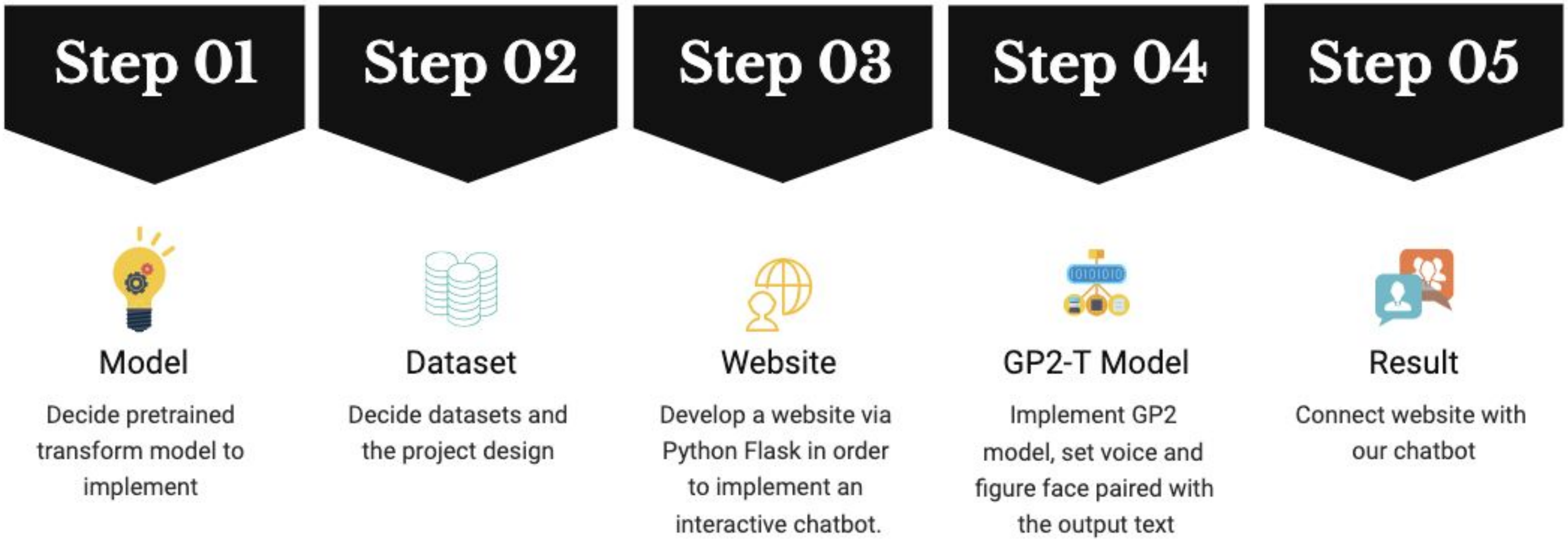
GPT2
355M Model

Conversation
Chat bot

Scientific Tutor

Poet

# Milestones



**Step 01**

**Model**

Decide pretrained transform model to implement

**Step 02**

**Dataset**

Decide datasets and the project design

**Step 03**

**Website**

Develop a website via Python Flask in order to implement an interactive chatbot.

**Step 04**

**GP2-T Model**

Implement GP2 model, set voice and figure face paired with the output text

**Step 05**

**Result**

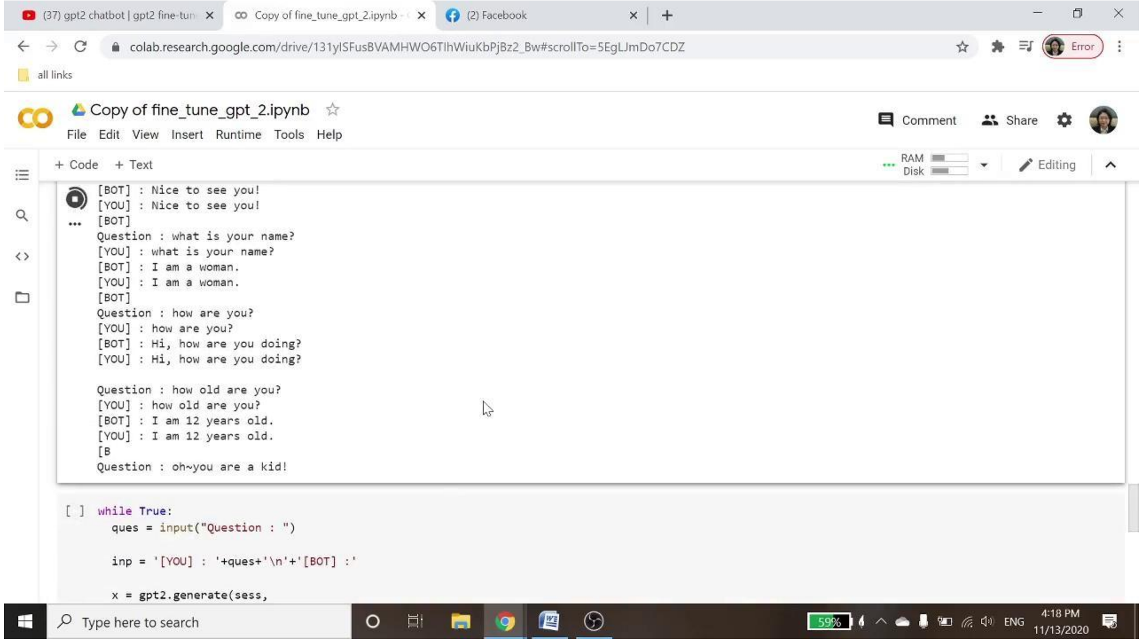
Connect website with our chatbot

# Fine tune chatbot trial

**Our first trial:**

Learned from a tutorial, and revised part of codes to fit our need.

In this trial, the dataset clearly labeled [you] and [bot], and the model freely added some conversation predictions or answered the question we typed in the prompt.



(37) gpt2 chatbot | gpt2 fine-tun
Copy of fine_tune_gpt_2.ipynb
(2) Facebook
colab.research.google.com/drive/131ylSFusBVAMHWO6TlhWiuKbPjBz2_Bw#scrollTo=5EgLJmDo7CDZ
Error
all links
Copy of fine_tune_gpt_2.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Comment
Share
+ Code
+ Text
RAM
Disk
Editing
Type here to search
59%
ENG
4:18 PM
11/13/2020

```
[BOT] : Nice to see you!
[YOU] : Nice to see you!
[BOT]
Question : what is your name?
[YOU] : what is your name?
[BOT] : I am a woman.
[YOU] : I am a woman.
[BOT]
Question : how are you?
[YOU] : how are you?
[BOT] : Hi, how are you doing?
[YOU] : Hi, how are you doing?

Question : how old are you?
[YOU] : how old are you?
[BOT] : I am 12 years old.
[YOU] : I am 12 years old.
[B
Question : oh~you are a kid!
```

```
[ ] while True:
      ques = input("Question : ")

      inp = '[YOU] : '+ques+'\n'+'[BOT] :'

      x = gpt2.generate(sess,
```

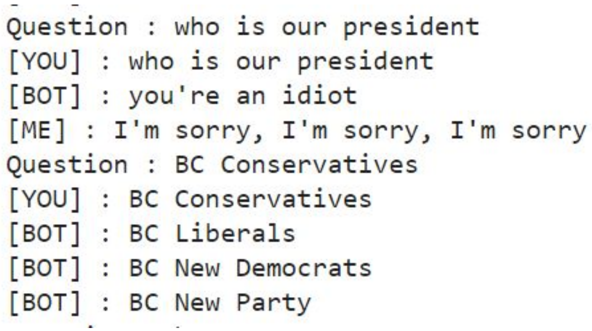
# Fine tune chatbot trial

**Another trials:**

1. Question-answer pair dataset which answer is yes/no or some are null

   Our chatbot was lost and could not form corresponding answer at all.

2. Question-answer dataset. Content like jeopardy questions. Perform well in learnt question.



```
Question : who is our president
[YOU] : who is our president
[BOT] : you're an idiot
[ME] : I'm sorry, I'm sorry, I'm sorry
Question : BC Conservatives
[YOU] : BC Conservatives
[BOT] : BC Liberals
[BOT] : BC New Democrats
[BOT] : BC New Party
```

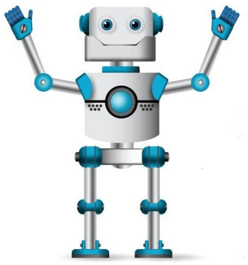
Jeopardy dataset result (only “Question” is our setting)

# Conversation chatbot

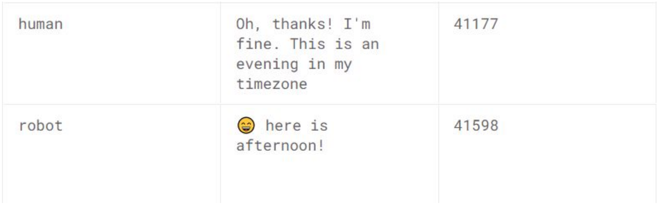
After discussing different possibilities, we decided to focus on two people conversation style, aiming to input more constructive thinking, life coaching style dataset to our chatbot as our project direction.

Two-people conversation dataset

Dataset: [rDany Chat]



1. 2.75 MB
2. Human text
3. Robot text
4. Conversation csv or json



| | | |
|---|---|---|
| human | Oh, thanks! I'm fine. This is an evening in my timezone | 41177 |
| robot | 😄 here is afternoon! | 41598 |

# Two-people conversation Demo

[YOU] : what do you do?

[BOT] : i'm a chatbot 🤖

[YOU] : you are a nice chatbot

[BOT] : thank you !

[YOU] : what color do you like?

[BOT] : :

[YOU] : do you like sports?

[BOT] : yep! and you ?

[YOU] : yes, I play tennis and soccer

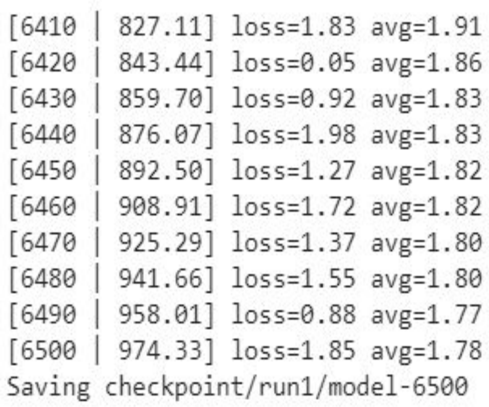
[BOT] : are you human ?

[YOU] : yes, I am

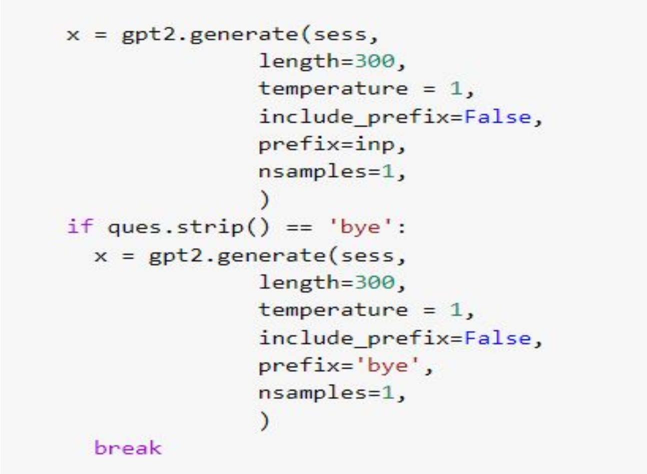
[BOT] : я? 😄

# A Poet Bot

1. We trained our model poems till 6500 steps for stabilizing its  performance in writing poems(we have fed it with 13 datasets)
2. We adjusted temperature to 1 for better creativity. (after trying several settings)



```
[6410 | 827.11] loss=1.83 avg=1.91
[6420 | 843.44] loss=0.05 avg=1.86
[6430 | 859.70] loss=0.92 avg=1.83
[6440 | 876.07] loss=1.98 avg=1.83
[6450 | 892.50] loss=1.27 avg=1.82
[6460 | 908.91] loss=1.72 avg=1.82
[6470 | 925.29] loss=1.37 avg=1.80
[6480 | 941.66] loss=1.55 avg=1.80
[6490 | 958.01] loss=0.88 avg=1.77
[6500 | 974.33] loss=1.85 avg=1.78
Saving checkpoint/run1/model-6500
```



```
x = gpt2.generate(sess,
              length=300,
              temperature = 1,
              include_prefix=False,
              prefix=inp,
              nsamples=1,
              )
if ques.strip() == 'bye':
  x = gpt2.generate(sess,
              length=300,
              temperature = 1,
              include_prefix=False,
              prefix='bye',
              nsamples=1,
              )
  break
```

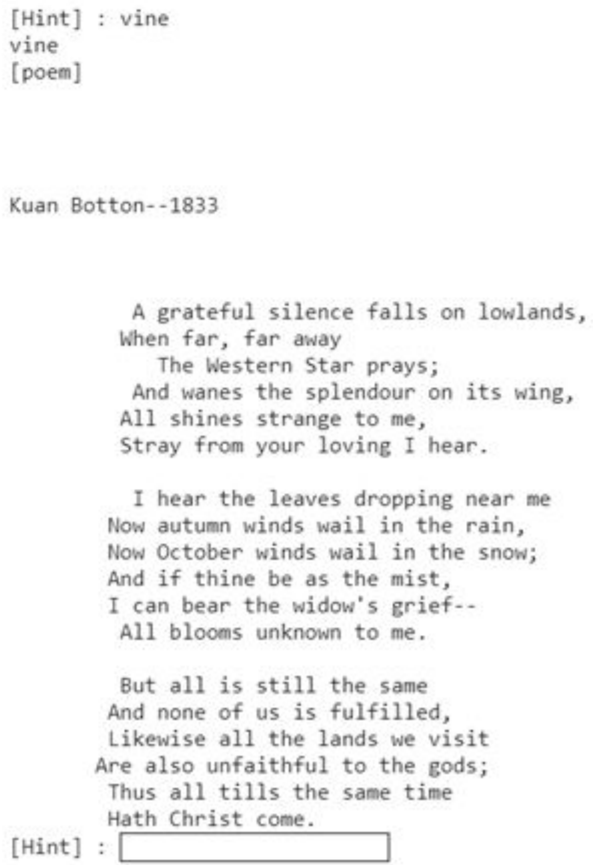
# What we found in training

- **Dataset:** we made our own dataset by using poem collections from Project Gutenberg. It provides copyright free ebooks. So we tended to select classics for our poet bot, most of the poems are from 19 century to early 20 century, as the result, poet bot's poems are more classical in word usages and style.
- **Training** : As mentioned before the quality of the dataset is critical for breeding the style and content of its performance. The more steps we feed, more stable the performance be.
- The last dataset is more influential, but the overall level of stability still build on how much we fed to the bot.
- The bot would learn to write articles after it learned from poet's letter written to his/her friends or diary style writings. When the bot started to confuse about writing styles, we input another dataset ensuring it gets more poems.

# Good work:

It has some good works, and it is one of them.
We noticed the poet named: Kuan Botton, 1833, and tried to search in Google, but failed to find this name. Neither could we find the poem on the internet.

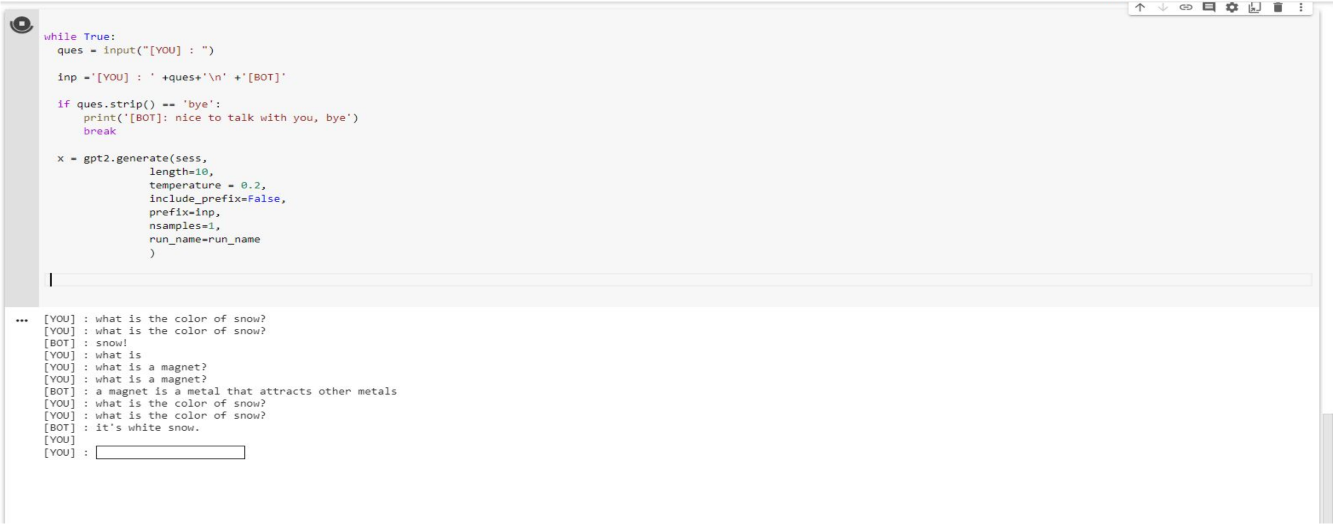
We gave the bot [hint] vine, and it made a poem of vine.
(not every time it follows our hint)



```
[Hint] : vine
vine
[poem]



Kuan Botton--1833


          A grateful silence falls on lowlands,
         When far, far away
            The Western Star prays;
          And wanes the splendour on its wing,
         All shines strange to me,
         Stray from your loving I hear.

          I hear the leaves dropping near me
        Now autumn winds wail in the rain,
        Now October winds wail in the snow;
        And if thine be as the mist,
        I can bear the widow's grief--
         All blooms unknown to me.

         But all is still the same
        And none of us is fulfilled,
        Likewise all the lands we visit
       Are also unfaithful to the gods;
        Thus all tills the same time
        Hath Christ come.
[Hint] : 
```

# A Scientist Bot

1. We trained our model with explanation sentences supporting 363 science questions from Allen Institute for AI, the University of Arizona, and Stony Brook University.
2. We adjusted the temperature to 0.2 for better accuracy (after trying several settings)



```
while True:
  ques = input("[YOU] : ")

  inp ='[YOU] : ' +ques+'\n' +'[BOT]'

  if ques.strip() == 'bye':
      print('[BOT]: nice to talk with you, bye')
      break

  x = gpt2.generate(sess,
                length=10,
                temperature = 0.2,
                include_prefix=False,
                prefix=inp,
                nsamples=1,
                run_name=run_name
                )
```

```
[YOU] : what is the color of snow?
[YOU] : what is the color of snow?
[BOT] : snow!
[YOU] : what is
[YOU] : what is a magnet?
[YOU] : what is a magnet?
[BOT] : a magnet is a metal that attracts other metals
[YOU] : what is the color of snow?
[YOU] : what is the color of snow?
[BOT] : it's white snow.
[YOU]
[YOU] :
```

# Poet and Scientist Bot Demo

# What’s next?

- Expand the training (better (or clean) datasets)
- Input image data to expand the possibilities of interaction
- Use other GPT-2 model
- Upgrade UI

# Repository

Github: https://github.com/LoriSchuan-dev/DLTKchatbotTEAM

Poet Bot collection: Some poems from the bot

# Resources

- GPT2 simple : [https://github.com/minimaxir/gpt-2-simple](https://github.com/minimaxir/gpt-2-simple)
- GPT2 tutorial: [https://minimaxir.com/2019/09/howto-gpt2/](https://minimaxir.com/2019/09/howto-gpt2/)
- Python Flask Documentation:
  - [https://flask.palletsprojects.com/en/1.1.x/](https://flask.palletsprojects.com/en/1.1.x/)
- A tutorial on on Flask:
  - https://www.youtube.com/watch?v=3mwFC4SHY-Y&t=248s&ab_channel=CleverProgrammer
- Voice Cloning Colab: https://colab.research.google.com/git

- Voice Cloning Library: https://github.com/CorentinJ/Real-Tim

# Resources

- Human Annotated Dialogues Dataset for Natural Conversational Agents
- Quotables
- Sample conversations to generate insights
- rDany Chat
- Ubuntu Dialogue Corpus
- Project Gutenberg (we made our own dataset from project Gutenberg)