9-1-2022

Natural Language Processing for Novel Writing

Leqing Qu

Okan Ersoy

Natural Language Processing for Novel Writing

Leqing Xu, Okan Ersoy

1. Instruction of Natural Language Processing

Natural language processing (NLP) is a sub-field of linguistics, computer science, and artificial intelligence. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. NLP mainly includes three parts: natural language generation, speech recognition, and natural language understanding. This project mainly focus on natural language generation.

2. Neural Network

(1) . Basic introduction of RNN

With the development of this field, nowadays, Neural Network is widely used in designing NLP projects. Convolutional neural network (CNN) and recurrent neural network (RNN), the two main types of DNN architectures, are widely explored to handle various NLP tasks.[1] RNN performs better in NLP, and it is widely used in it.

The value s of the hidden layer of RNN not only depends on the current input x, but also depends on the value s of the last hidden layer:

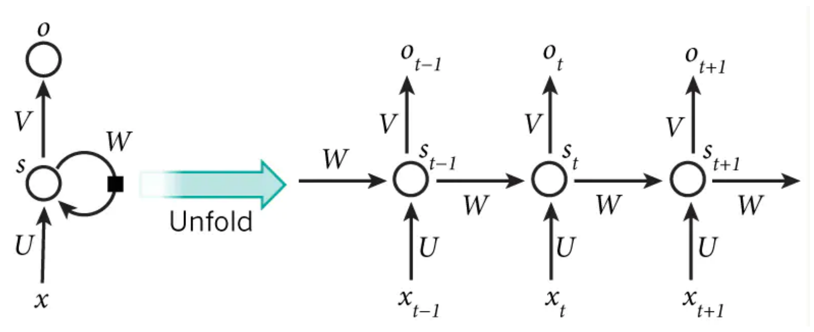


Figure 1. schematic diagram of RNN

Where W is the last value of the hidden layer as the weight of the input this time. t is the time, x is the input layer, s is the hidden layer, o is the output layer.
Different from normal neural network, RNN can look forward to any number of input values:

$$o_t = g(V_{S_t}) \tag{1}$$

$$s_t = f(U_{X_t} + W_{S_{t-1}}) \tag{2}$$

Continue substituting Eq. 2 into Eq. 1:

[1] Comparative Study of CNN and RNN for Natural Language Processing. Wenpeng Yin

$$
\begin{aligned}
o_t &= g(V_{S_t}) \\
&= Vf(U_{X_t} + W_{S_{t-1}}) \\
&= Vf(U_{X_t} + Wf(U_{X_{t-1}} + W_{X_{t-2}})) \\
&= Vf(U_{X_t} + Wf(U_{X_{t-1}} + Wf(U_{X_{t-2}} + \ldots)))
\end{aligned}
$$

3. GPT-2

GPT-2 is a language model released by OpenAI to predict the next word in 40GB of Internet text. GPT-2 is a large transformer-based language model with 1.5 billion parameters. Simply, the role of the language model is to predict what the next word will be based on a part of an existing sentence. Through this project, using GPT-2-simple to finetune the model and generate text.[2]

4. GPT-3

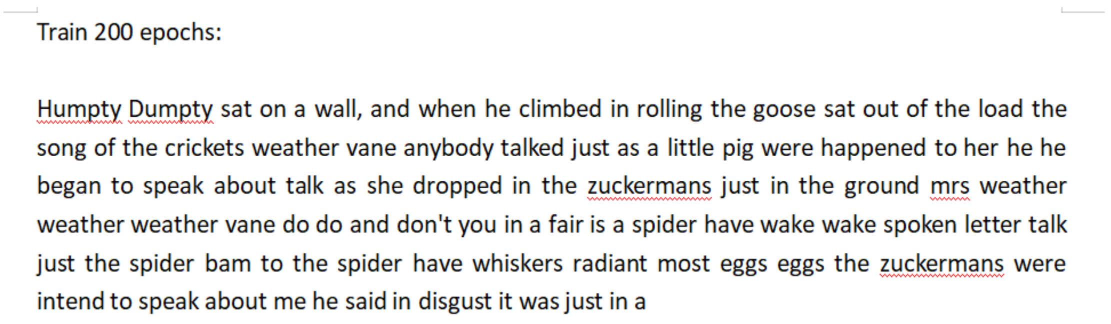
GPT-3 is the most powerful language model ever. Its predecessor, GPT-2, released last year, was already able to spit out convincing streams of text in a range of different styles when prompted with an opening sentence. But GPT-3 is a big leap forward. The model has 175 billion parameters (the values that a neural network tries to optimize during training), compared with GPT-2's already vast 1.5 billion.[3] Different from GPT-2, GPT-3 doesn't require to be finetuned in the conventional way, with GPT-3, it helps to anthropomorphize it: sometimes users literally just have to ask for what they want.[4]

5. Project description

The goal of this project is using the fictions written by Dr. Seuss, E.B White, Ian Fleming, James Herriot and Samin Nostrat to generate an novel which hero is called Humpty Dumpty, and talk about his adventure from table to field.

The first approach of this project is using all this book as the training data, using Keras and Tensorflow, build the neural network and train it. But the result cannot case sensitive and there are no punctuation. The contents of text generated is not satisfactory.

[2] Python package to easily retrain OpenAI's GPT-2 text-generating model on new texts.
[3] OpenAI's new language generator GPT-3 is shockingly good—and completely mindless. Will Douglas Heaven
[4] GPT-3 Creative Fiction.


Train 200 epochs:

Humpty Dumpty sat on a wall, and when he climbed in rolling the goose sat out of the load the song of the crickets weather vane anybody talked just as a little pig were happened to her he he began to speak about talk as she dropped in the zuckermans just in the ground mrs weather weather weather vane do do and don't you in a fair is a spider have wake wake spoken letter talk just the spider bam to the spider have whiskers radiant most eggs eggs the zuckermans were intend to speak about me he said in disgust it was just in a

Figure 2. Text generated by using Keras and Tensorflow

The second approach is funetuning GPT-2 pre-trained model with training data and generate text. After choosing the biggest model and adjust the parameters, it can generate more natural sentences.

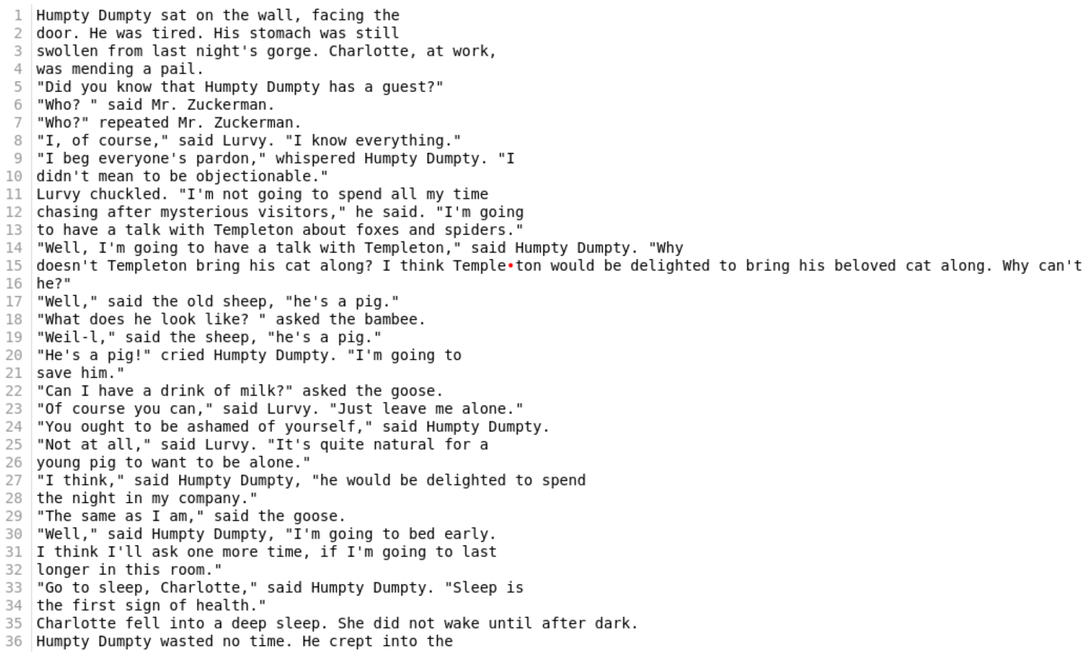

```
Humpty Dumpty sat on the wall, facing the
door. He was tired. His stomach was still
swollen from last night's gorge. Charlotte, at work,
was mending a pail.
"Did you know that Humpty Dumpty has a guest?"
"Who? " said Mr. Zuckerman.
"Who?" repeated Mr. Zuckerman.
"I, of course," said Lurvy. "I know everything."
"I beg everyone's pardon," whispered Humpty Dumpty. "I
didn't mean to be objectionable."
Lurvy chuckled. "I'm not going to spend all my time
chasing after mysterious visitors," he said. "I'm going
to have a talk with Templeton about foxes and spiders."
"Well, I'm going to have a talk with Templeton," said Humpty Dumpty. "Why
doesn't Templeton bring his cat along? I think Temple•ton would be delighted to bring his beloved cat along. Why can't
he?"
"Well," said the old sheep, "he's a pig."
"What does he look like? " asked the bambee.
"Weil-l," said the sheep, "he's a pig."
"He's a pig!" cried Humpty Dumpty. "I'm going to
save him."
"Can I have a drink of milk?" asked the goose.
"Of course you can," said Lurvy. "Just leave me alone."
"You ought to be ashamed of yourself," said Humpty Dumpty.
"Not at all," said Lurvy. "It's quite natural for a
young pig to want to be alone."
"I think," said Humpty Dumpty, "he would be delighted to spend
the night in my company."
"The same as I am," said the goose.
"Well," said Humpty Dumpty, "I'm going to bed early.
I think I'll ask one more time, if I'm going to last
longer in this room."
"Go to sleep, Charlotte," said Humpty Dumpty. "Sleep is
the first sign of health."
Charlotte fell into a deep sleep. She did not wake until after dark.
Humpty Dumpty wasted no time. He crept into the
```

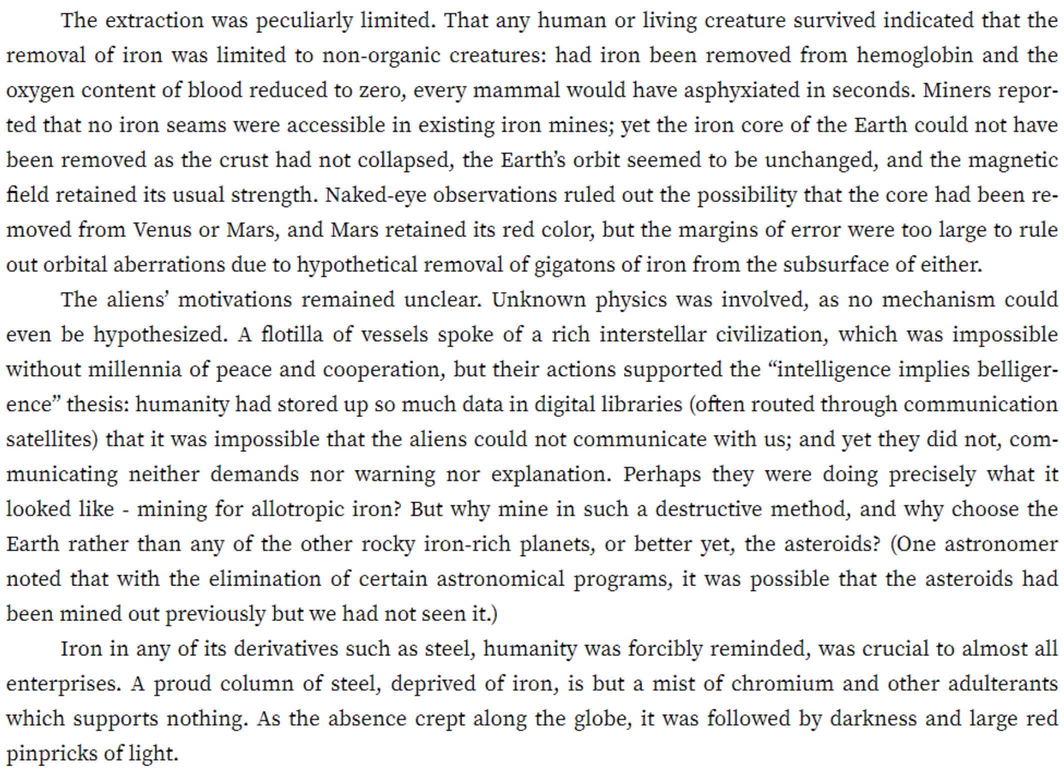
Figure 3. Text generated after finetune GPT-2 model

GPT-2 is better at maintaining context over its entire generation length, making it good for generating conversational text. The text is also generally gramatically correct, with proper capitalization and few typoes.[5] But when looking into to the full text, it doesn't build the relationship between sentences and sentences, in other words, the whole structure of the novel has not been built.

In order to solve this problem, using GPT-3 to generate text becomes the third approach, since this model has 175 billion parameters which is much bigger than GPT-2. Given any text prompt like a phrase or a sentence, GPT-3 returns a text completion in natural language. Developers

[5] Python package to easily retrain OpenAI's GPT-2 text-generating model on new texts.

can ”program” GPT-3 by showing it just a few examples or ”prompts.”[6]


The extraction was peculiarly limited. That any human or living creature survived indicated that the removal of iron was limited to non-organic creatures: had iron been removed from hemoglobin and the oxygen content of blood reduced to zero, every mammal would have asphyxiated in seconds. Miners reported that no iron seams were accessible in existing iron mines; yet the iron core of the Earth could not have been removed as the crust had not collapsed, the Earth's orbit seemed to be unchanged, and the magnetic field retained its usual strength. Naked-eye observations ruled out the possibility that the core had been removed from Venus or Mars, and Mars retained its red color, but the margins of error were too large to rule out orbital aberrations due to hypothetical removal of gigatons of iron from the subsurface of either.

The aliens' motivations remained unclear. Unknown physics was involved, as no mechanism could even be hypothesized. A flotilla of vessels spoke of a rich interstellar civilization, which was impossible without millennia of peace and cooperation, but their actions supported the "intelligence implies belligerence" thesis: humanity had stored up so much data in digital libraries (often routed through communication satellites) that it was impossible that the aliens could not communicate with us; and yet they did not, communicating neither demands nor warning nor explanation. Perhaps they were doing precisely what it looked like - mining for allotropic iron? But why mine in such a destructive method, and why choose the Earth rather than any of the other rocky iron-rich planets, or better yet, the asteroids? (One astronomer noted that with the elimination of certain astronomical programs, it was possible that the asteroids had been mined out previously but we had not seen it.)

Iron in any of its derivatives such as steel, humanity was forcibly reminded, was crucial to almost all enterprises. A proud column of steel, deprived of iron, is but a mist of chromium and other adulterants which supports nothing. As the absence crept along the globe, it was followed by darkness and large red pinpricks of light.

Figure 4. Novel generated by GPT-3

Based on the huge mount of parameters and complicated neural network of GPT-3, we expect to generate the real novel which describe the adventure of Humpty Dumpty from the table to field. The content and structure of this novel will imitate Dr. Seuss, E.B White, Ian Fleming, James Herriot and Samin Nostrat’s books.

6. Introduction of GPT-3.

GPT-3 can be applied to virtually any task that involves understanding or generating natural language or code. OpenAI offers a spectrum of models with different levels of power suitable for different tasks, as well as the ability to fine-tune your own custom models. These models can be used for everything from content generation to semantic search and classification. The completions endpoint is at the center of this API. There are four models provides by OpenAI: Davinci, Curie, Babbage and Ada. Davinci is the most capable engine and can perform any task the other models can perform and often with less instruction. Davinci is good at complex intent, cause and effect, summarization for audience. Curie is extremely powerful, and very fast, it is good at language translation, complex classification, text sentiment and summarization. Babbage can perform straightforward tasks like simple classification and it is good at moderate classification, semantic search classification. Ada is usually the fastest model and can perform tasks like parsing text, address correction and certain kinds of classification tasks that don’t require too much nuance. Ada’s performance can often be improved by providing more context. It is good at parsing text, simple classification, address correction, keywords.

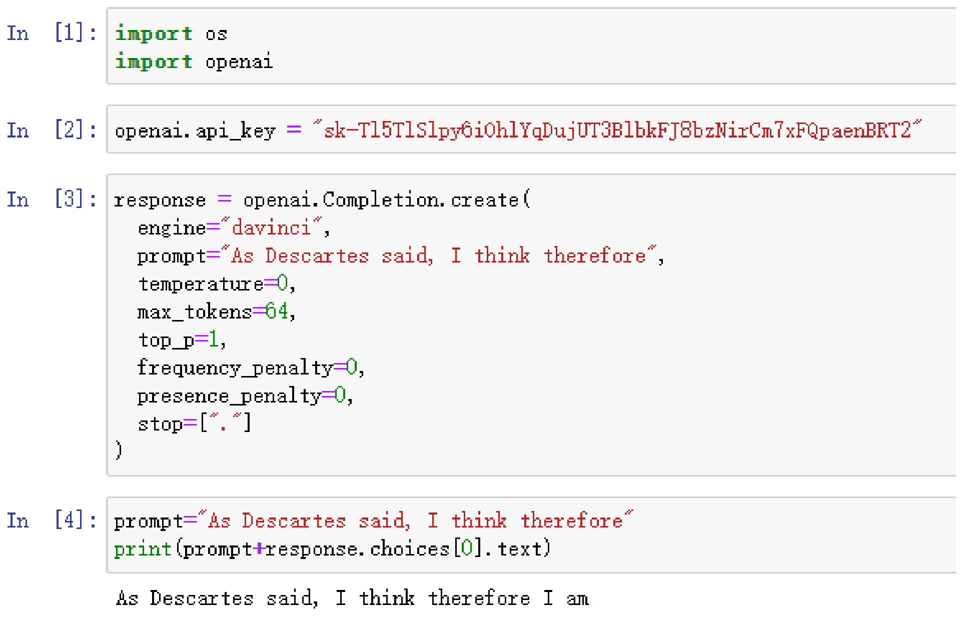
[6] GPT-3 Powers the Next Generation of Apps.

All of these models understand and process text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word “hamburger” gets broken up into the tokens “ham”, “bur” and “ger”, while a short and common word like “pear” is a single token. Many tokens start with a whitespace behind, for example “hello” and “ bye”.[7]

Through this project, the completions endpoint is the most important part. Input some text as a prompt, and the model will generate a text completion that attempts to match whatever context or pattern you gave it.

7. Text completion in GPT-3.

The GPT-3 completions endpoint can be used for a wide variety of tasks. As discussed above, if input some text as a prompt, the model will generate a text completion. That is the basic and the most important idea of this project.



```
In [1]: import os
        import openai

In [2]: openai.api_key = "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"

In [3]: response = openai.Completion.create(
          engine="davinci",
          prompt="As Descartes said, I think therefore",
          temperature=0,
          max_tokens=64,
          top_p=1,
          frequency_penalty=0,
          presence_penalty=0,
          stop=["."]
        )

In [4]: prompt="As Descartes said, I think therefore"
        print(prompt+response.choices[0].text)

        As Descartes said, I think therefore I am
```

Figure 5. Simple example of text completion

This figure shown above is a very simple text completion task, for the task which is more complicated, adjusting the parameter of the model to get different results.

8. Fine-tuning of GPT-3.

8.1 Fine-tune.

As mentioned above, the goal of this project is to generate the real novel which describe the adventure of Humpty Dumpty from the table to field. The content and structure of this novel will imitate Dr. Seuss, E.B White, Ian Fleming, James Herriot and Samin Nostrat’s books. If directly generate this text without these books, it will generate the text not corresponding to the goal. So

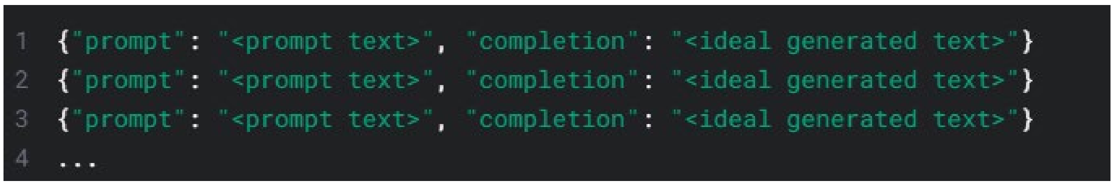
[7] GPT-3 key concepts.

it is necessary to fine-tune the model based on our training data.

8.2 Fine-tuning in GPT-3.

GPT-3 has been pre-trained on a vast amount of text from the open internet. When given a prompt with just a few examples, it can often intuit what task are trying to perform and generate a plausible completion. This is often called "few-shot learning." It can also get higher quality results than prompt design, it also has the ability to train on more examples than can fit in a prompt.

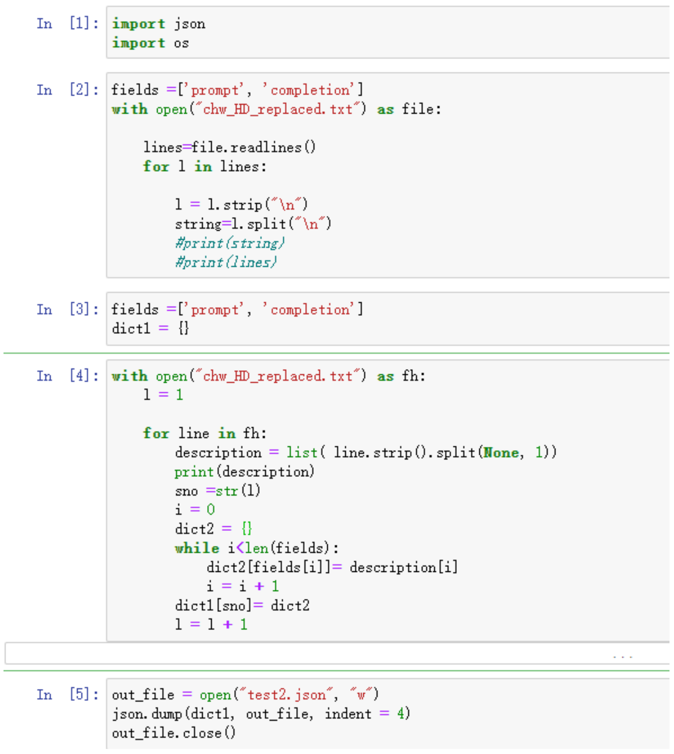
Different from GPT-2 and other former products, preparing the GPT-3 training data requires the format which is shown below:[8]



```
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
...
```

Figure 6. Format of GPT-3 training data.

This data must be JSONL document. since our training data are text files, the first step is transferring these text files into JSON format, the figure below shows how to transfer one og these books into JSON file:

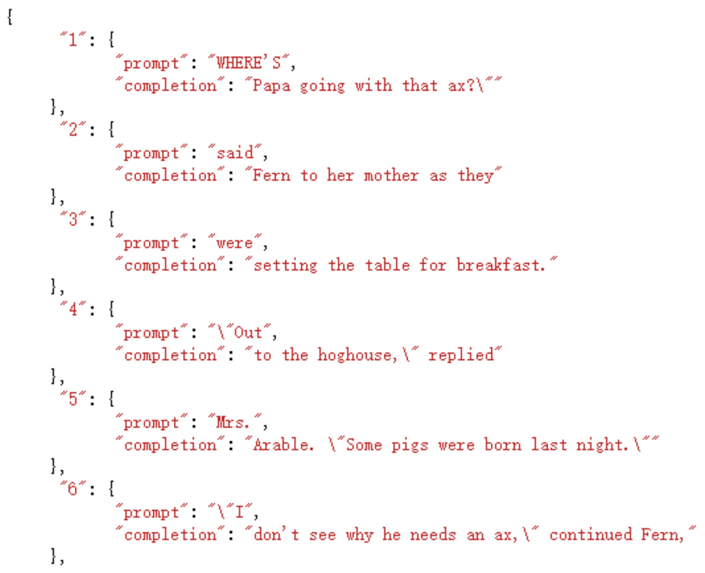


```
In [1]: import json
        import os

In [2]: fields =['prompt', 'completion']
        with open("chw_HD_replaced.txt") as file:

            lines=file.readlines()
            for l in lines:

                l = l.strip("\n")
                string=l.split("\n")
                #print(string)
                #print(lines)

In [3]: fields =['prompt', 'completion']
        dict1 = {}

In [4]: with open("chw_HD_replaced.txt") as fh:
            l = 1

            for line in fh:
                description = list( line.strip().split(None, 1))
                print(description)
                sno =str(l)
                i = 0
                dict2 = {}
                while i<len(fields):
                    dict2[fields[i]]= description[i]
                    i = i + 1
                dict1[sno]= dict2
                l = l + 1

                                                                    ...

In [5]: out_file = open("test2.json", "w")
        json.dump(dict1, out_file, indent = 4)
        out_file.close()
```

[8] Prepare training data of GPT-3

Figure 7. Code of transfer text to JSON.

After finishing these transferring tasks, training data becomes the format shown below:



```
{
    "1": {
        "prompt": "WHERE'S",
        "completion": "Papa going with that ax?\""
    },
    "2": {
        "prompt": "said",
        "completion": "Fern to her mother as they"
    },
    "3": {
        "prompt": "were",
        "completion": "setting the table for breakfast."
    },
    "4": {
        "prompt": "\"Out",
        "completion": "to the hoghouse,\" replied"
    },
    "5": {
        "prompt": "Mrs.",
        "completion": "Arable. \"Some pigs were born last night.\""
    },
    "6": {
        "prompt": "\"I",
        "completion": "don't see why he needs an ax,\" continued Fern,"
    },
```

Figure 8. JSON file of training data.

OpenAI also offers a tool called CLI data preparation tool[9] to convert the data into this file format. After get all files in a correct format, fine-tuning job can be started.

The fine-tuning job has not been finished, I expect to finish it in the next week, at that time, this project report will be updated.

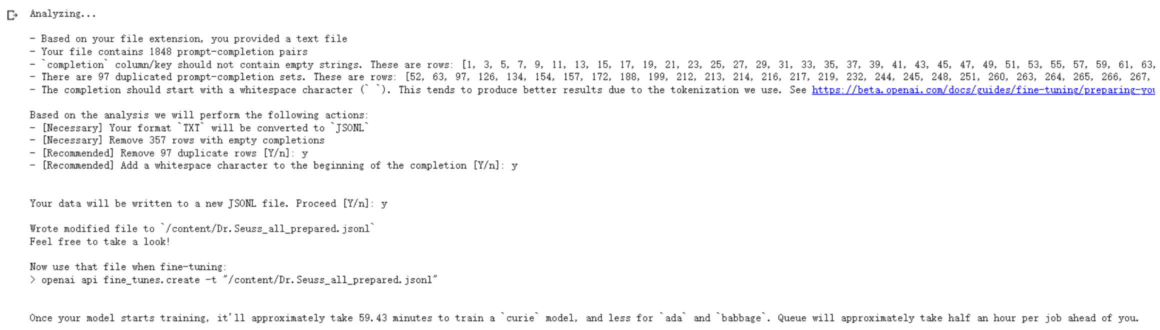
8.3 Implement of fine-tuning in GPT-3

To implement fine-tuning, we will first need to set up the environment, in this example, we will use all of novels by Dr. Seuss as our fine-tuning data. Run the code shown below:

```
!openai tools fine_tunes.prepare_data -f "/content/Dr.Seuss_all.txt"
```

This code will convert the TXT file into JASONL file which is the correct format of our fine-tuning job, it will also remove empty completions and duplicate rows. The result is shown below:

[9] GPT-3 fine-tune.



```
Analyzing...

- Based on your file extension, you provided a text file
- Your file contains 1848 prompt-completion pairs
- `completion` column/key should not contain empty strings. These are rows: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63,
- There are 97 duplicated prompt-completion sets. These are rows: [52, 63, 97, 126, 134, 154, 157, 172, 188, 199, 212, 213, 214, 216, 217, 219, 232, 244, 245, 248, 251, 260, 263, 264, 265, 266, 267,
- The completion should start with a whitespace character (` `). This tends to produce better results due to the tokenization we use. See https://beta.openai.com/docs/guides/fine-tuning/preparing-you

Based on the analysis we will perform the following actions:
- [Necessary] Your format `TXT` will be converted to `JSONL`
- [Necessary] Remove 357 rows with empty completions
- [Recommended] Remove 97 duplicate rows [Y/n]: y
- [Recommended] Add a whitespace character to the beginning of the completion [Y/n]: y


Your data will be written to a new JSONL file. Proceed [Y/n]: y

Wrote modified file to `/content/Dr.Seuss_all_prepared.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:
> openai api fine_tunes.create -t "/content/Dr.Seuss_all_prepared.jsonl"


Once your model starts training, it'll approximately take 59.43 minutes to train a `curie` model, and less for `ada` and `babbage`. Queue will approximately take half an hour per job ahead of you.
```

Figure 9. convert TXT file into JASONL file.

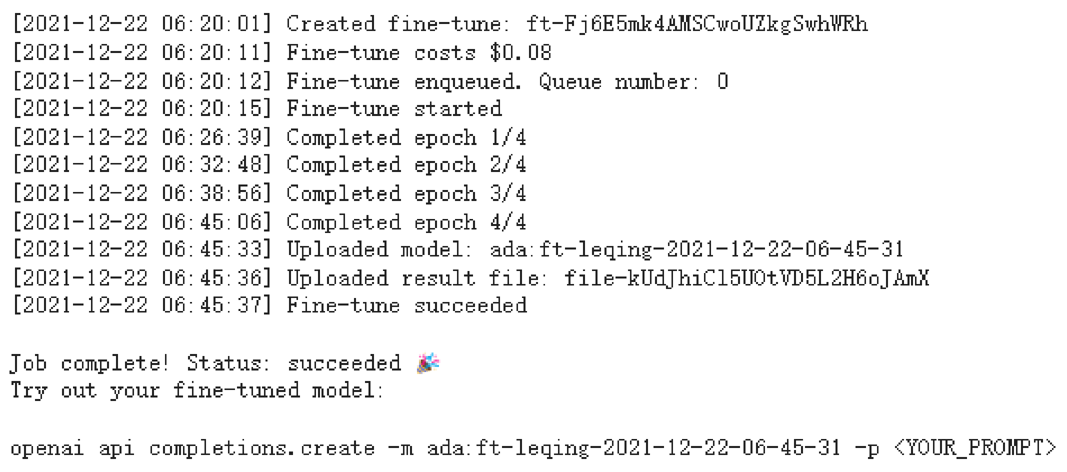
Once we got the JASON format of dataset, we should be able to use it in creating a fine-tuning job, run the code below:

```
!openai api fine_tunes.create -t "/content/Dr.Seuss_all_prepared.jsonl" -m "ada"
```

Since this project is implemented on Google colab environment, the stream may interrupte due to client disconnected, in order to solve this problem, run the code below to resume the routine:

```
!openai api fine_tunes.follow -i ft-JpsUMgZ3sGDcA781N2MBUFFF
```

In this example, we choose model "ada" as our fine-tuning model, it can give us a shorter time to complete our fine-tune job. After fine-tuning, we should be able to see the result below:



```
[2021-12-22 06:20:01] Created fine-tune: ft-Fj6E5mk4AMSCwoUZkgSwhWRh
[2021-12-22 06:20:11] Fine-tune costs $0.08
[2021-12-22 06:20:12] Fine-tune enqueued. Queue number: 0
[2021-12-22 06:20:15] Fine-tune started
[2021-12-22 06:26:39] Completed epoch 1/4
[2021-12-22 06:32:48] Completed epoch 2/4
[2021-12-22 06:38:56] Completed epoch 3/4
[2021-12-22 06:45:06] Completed epoch 4/4
[2021-12-22 06:45:33] Uploaded model: ada:ft-leqing-2021-12-22-06-45-31
[2021-12-22 06:45:36] Uploaded result file: file-kUdJhiCl5UOtVD5L2H6oJAmX
[2021-12-22 06:45:37] Fine-tune succeeded

Job complete! Status: succeeded 🎉
Try out your fine-tuned model:

openai api completions.create -m ada:ft-leqing-2021-12-22-06-45-31 -p <YOUR_PROMPT>
```

Figure 10. a demo of successful fine-tune.

After completing our job, a fine-tuned model is created and saved to account, for example, if we want to generate Dr. Seuss's style text, we can directly use this model, just same as we use models given by OpenAI.

To test performance of our model, run the code below:

```
response=openai.Completion.create(
    model="ada:ft-leqing-2022-01-12-04-12-45",
    prompt="I' m sending you right up!",
    max_tokens=200,
    temperature=0.9)
prompt="I' m sending you right up!"
print(prompt+response.choices[0].text)
```

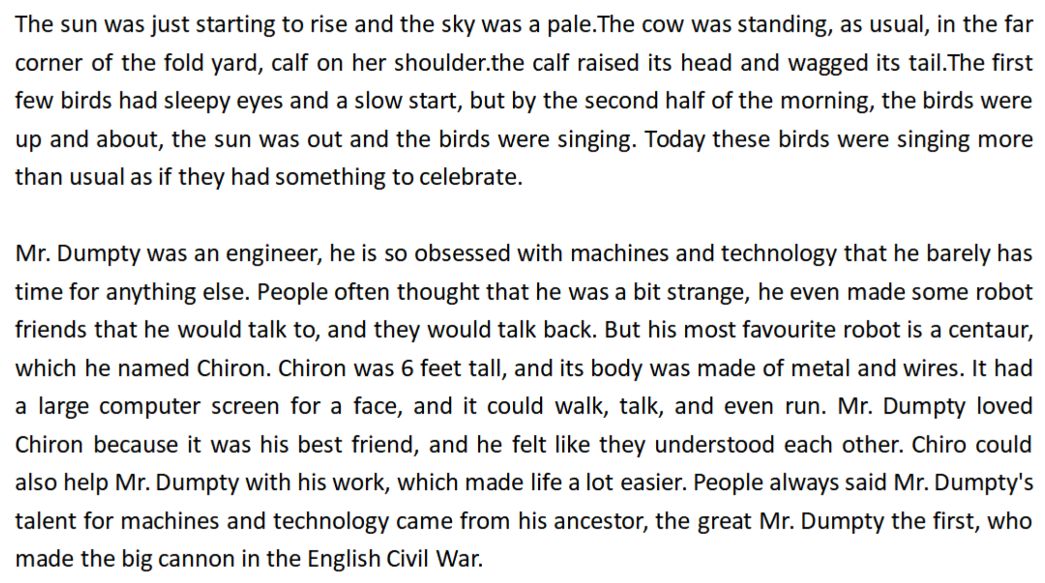
By observing the result we found that it generated Dr. Seuss style text, that means our fine-tuning job successed. However, our ultimate goal is to generate a novel which contains logic, rich content and specific style, we will need to combine different novels written by different authors.The result given by our fine-tuned model is shown below:

```
I'm sending you right up!
You may, I say.
You don't read books.
You read people.
You read places.
You read people's worries.
You read people's needs.
You read people's desires.
You read people's needs.
You read people's aspirations.
You read people's dreams.
You read people's agraves.
You read places where people are Harmony and hope.
You read books that tell you what to task your artsingers are doing.
You read a book that outlines you'll give you's culture a stacker.
You read books that put you's in the terrarium.
```

Figure 11. Dr. Seuss's style text generated by fine-tuned model

9. Difficulties and solution

As mentioned in pervious section, our ultimate goal is to generate a novel which contains logic, rich content and specific style. Sometimes if we only do the fine-tuning job and only use it to generate long text, duplicated and chaos text might be generated. Also, the maxium tokens allowed in GPT-3 is 2048 tokens, which means our prompt plus generated text should not exceed 2048 tokens. It also becomes a difficulty because we want to build the connection between paragraph and paragraph. After thinking about those difficulties, we proposed an idea that we can use our fine-tuned model to complete the first a few sentences as our start of our novel, then we switch back to model given by OpenAI and use that model to continue our job. The reason is these model contain a big amount of parameters and they have a very strong self-adaption abilty. Once we use our fine-tuned model to generate some sentences and use those sentences as our input, for example, the model "Davinci" could be able to imitate those input and finish the text completion. A result below could prove our idea is correct:

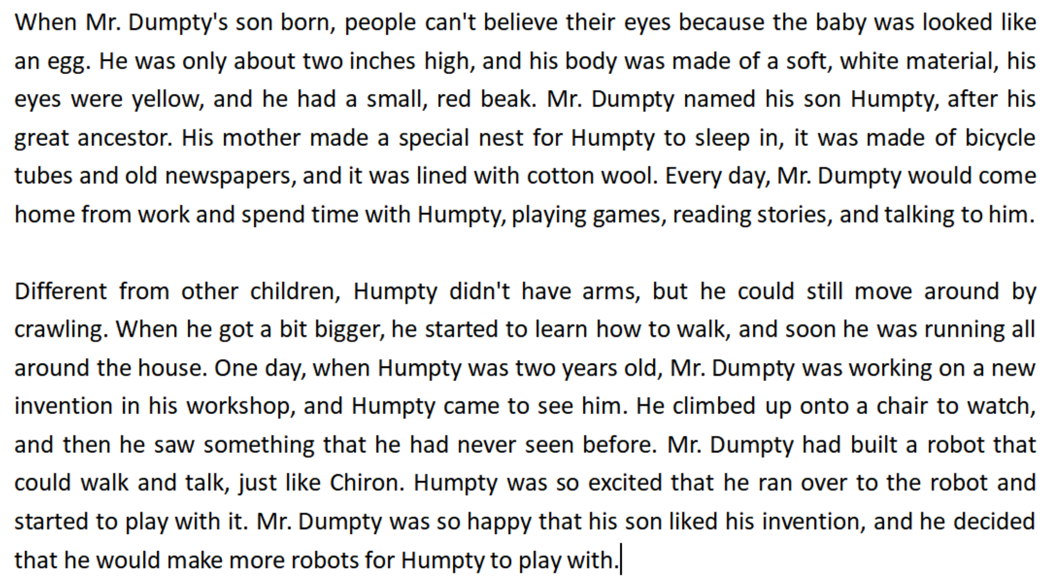


The sun was just starting to rise and the sky was a pale.The cow was standing, as usual, in the far corner of the fold yard, calf on her shoulder.the calf raised its head and wagged its tail.The first few birds had sleepy eyes and a slow start, but by the second half of the morning, the birds were up and about, the sun was out and the birds were singing. Today these birds were singing more than usual as if they had something to celebrate.

Mr. Dumpty was an engineer, he is so obsessed with machines and technology that he barely has time for anything else. People often thought that he was a bit strange, he even made some robot friends that he would talk to, and they would talk back. But his most favourite robot is a centaur, which he named Chiron. Chiron was 6 feet tall, and its body was made of metal and wires. It had a large computer screen for a face, and it could walk, talk, and even run. Mr. Dumpty loved Chiron because it was his best friend, and he felt like they understood each other. Chiro could also help Mr. Dumpty with his work, which made life a lot easier. People always said Mr. Dumpty's talent for machines and technology came from his ancestor, the great Mr. Dumpty the first, who made the big cannon in the English Civil War.

Figure 12. Long text generation by different styles

As the figure show above, we fisrt use our fine-tune model to generate Jame Herriot's style text and use it as our first paragraph, then we generate E.B. White's style text, use those sentences as the input to model "Davinci" and continue text completion. In order to escape the maxium tokens issue and build connection between paragraph and paragraph, our idea is to generate a paragraph fisrt, and use this output paragraph as our next generation input. It will somehow give the logic to the whole novel, we call it iteration idea. Based on our fine-tuned model and iteration idea, we have reason to believe that it can generate a whole novel which contains different elements by different authors with a strong logic.



When Mr. Dumpty's son born, people can't believe their eyes because the baby was looked like an egg. He was only about two inches high, and his body was made of a soft, white material, his eyes were yellow, and he had a small, red beak. Mr. Dumpty named his son Humpty, after his great ancestor. His mother made a special nest for Humpty to sleep in, it was made of bicycle tubes and old newspapers, and it was lined with cotton wool. Every day, Mr. Dumpty would come home from work and spend time with Humpty, playing games, reading stories, and talking to him.

Different from other children, Humpty didn't have arms, but he could still move around by crawling. When he got a bit bigger, he started to learn how to walk, and soon he was running all around the house. One day, when Humpty was two years old, Mr. Dumpty was working on a new invention in his workshop, and Humpty came to see him. He climbed up onto a chair to watch, and then he saw something that he had never seen before. Mr. Dumpty had built a robot that could walk and talk, just like Chiron. Humpty was so excited that he ran over to the robot and started to play with it. Mr. Dumpty was so happy that his son liked his invention, and he decided that he would make more robots for Humpty to play with.

Figure 12. Paragraphs generated by using iteration idea

10. Conclusion

This project start from September 2021, we have gone through data collection, trained neural network ourselves, tried to use GPT-2 and finally we found that GPT-3 is better. We also found that with the help of GPT-3, based on iteration idea and fine-tuned model, we are able to generate a novel written by AI. Due to the time limit, we didn't finish the whole novel. Noticed that to complete this job, it can not be purely generated by AI. Every text generation will give totally different sentences and that will need human to check whether every sentences are reasonable. Also if the whole job is finishing under the supervision of a novelist, it will make the whole structure and content more reasonable. Related introduction, codes and results can be found at my Github website: https://github.com/LeqingXu1997/NLP-for-novel-writing

11. Acknowledgements

I would like to say thanks to respectful professor Okan Ersoy for giving me such a good opportunity, also in this project, without professor Ersoy's supervision and suggestion, I cannot make any progress. In this project, I went through many difficulties and I learned a lot in this project. After graduated from Purdue ECE, I will continue diving into Machine Learning and NLP fields, always learn more with the attitude of a newborn.

12. References

(1) . Comparative Study of CNN and RNN for Natural Language Processing. Wenpeng Yin
https://arxiv.org/pdf/1702.01923.pdf

(2) . Python package to easily retrain OpenAI's GPT-2 text-generating model on new texts.
https://github.com/minimaxir/gpt-2-simple

(3) . OpenAI's new language generator GPT-3 is shockingly good—and completely mindless. Will Douglas Heaven
https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/

(4) . GPT-3 Creative Fiction.
https://www.gwern.net/GPT-3

(5) . GPT-3 Powers the Next Generation of Apps.
https://openai.com/blog/gpt-3-apps/

(6) , (7), (8), (9). https://beta.openai.com/docs/introduction