Assorted topics

Part 1

IR - keeps evolving...

As with everything else, the field of information retrieval keeps changing - there are newer forms of data, newer algorithms, cross-pollination from other fields...

In this lecture and the next, we take a BRIEF look at a variety of developments. 'BRIEF' because each topic (eg. vector DBs) is a detailed area, we simply can't cover all of them in any detail. In other words, you can look into what interests you, later (after the course ends).

The following topics are grouped where possible, but are otherwise in no particular order.

Image understanding and search

ImageNet is a massive DB, of human-annotated images.

It can serve as a vehicle to drive ML algorithms (eg. CNN, VT): https://www.youtube.com/watch?v=40riCqvRoMs

You can learn more here.

Code search

GitHub does code search this way: https://github.blog/2023-02-06-the-technology-behind-githubs-new-code-search/

LBS/PS

Location Based Search (LBS), ie. Proximity Search (PS), is about using location data where the query originates, and using that to return responses.

Here is an intro: https://www.businessnewsdaily.com/5386-location-based-services.html

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8399432/ presents an efficient way to do LBS.

LBS can be exploited, to find (track) a user precisely:
https://www.cs.uic.edu/~polakis/tr/proximity-tr.pdf

Similarity search using vector DBs

Pinecone is a NEW type of DB called 'vector DB', which embeds into vector (XYZ...) orthogonal space, the items (data) we want to search for (using 'cosine similarity'!). Here is a primer.

We can also use vector DBs as 'infinite memory', for use with generative QA sessions.

LDA, for topic modeling

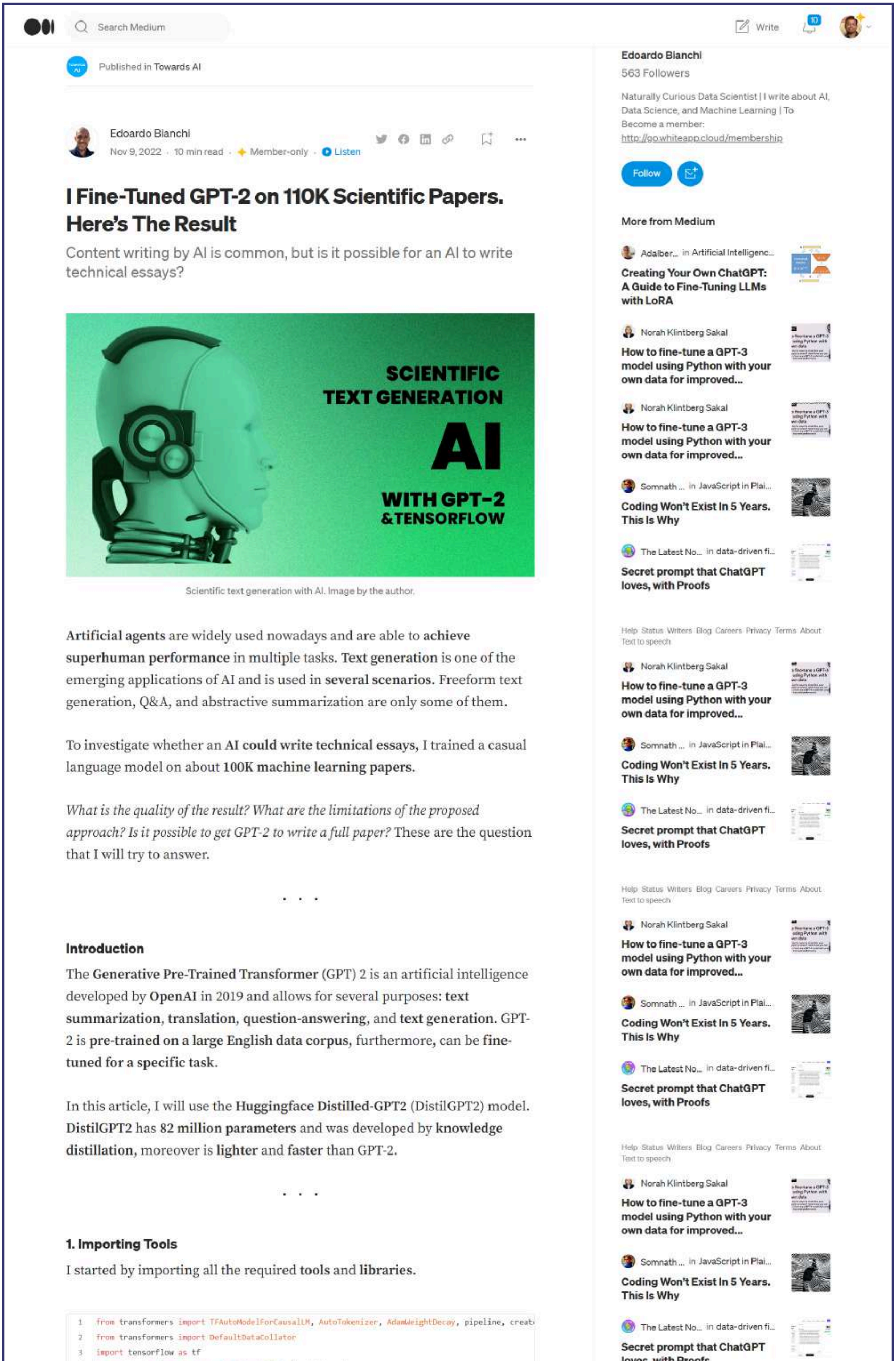
Given a document, how to predict its dominant topic(s)? Here is one approach.

ANN (Faiss)

Approximate Nearest Neighbor calcs are much faster than exact ones. Here is more.

Task-specific fine-tuning

'Fine-tuning' is the concept of adapting a general-purpose model, for a specific purpose. The following is an account of GPT-2 was fine-tuned to generate scientific paper abstracts:



Published in Towards AI

Edoardo Bianchi
Nov 9, 2022 · 10 min read · Member-only · Listen

I Fine-Tuned GPT-2 on 110K Scientific Papers. Here's The Result

Content writing by AI is common, but is it possible for an AI to write technical essays?

Scientific text generation with AI. Image by the author.

Artificial agents are widely used nowadays and are able to **achieve superhuman performance** in multiple tasks. **Text generation** is one of the emerging applications of AI and is used in **several scenarios.** Freeform text generation, Q&A, and abstractive summarization are only some of them.

To investigate whether an **AI could write technical essays,** I trained a casual language model on about **100K machine learning papers.**

What is the quality of the result? What are the limitations of the proposed approach? Is it possible to get GPT-2 to write a full paper? These are the question that I will try to answer.

. . .

Introduction

The **Generative Pre-Trained Transformer** (GPT) 2 is an artificial intelligence developed by **OpenAI** in 2019 and allows for several purposes: **text summarization, translation, question-answering,** and **text generation.** GPT-2 is **pre-trained on a large English data corpus,** furthermore, can be **fine-tuned for a specific task.**

In this article, I will use the **Huggingface Distilled-GPT2** (DistilGPT2) model. **DistilGPT2** has **82 million parameters** and was developed by **knowledge distillation,** moreover is **lighter** and **faster** than GPT-2.

. . .

1. Importing Tools

I started by importing all the required **tools** and **libraries.**

```
from transformers import TFAutoModelForCausalLM, AutoTokenizer, AdamWeightDecay, pipeline, creat
from transformers import DefaultDataCollator
import tensorflow as tf
```

Edoardo Bianchi
563 Followers

Naturally Curious Data Scientist | I write about AI, Data Science, and Machine Learning | To Become a member: http://go.whiteapp.cloud/membership

Follow

More from Medium

Adalber... in Artificial Intelligenc...
Creating Your Own ChatGPT: A Guide to Fine-Tuning LLMs with LoRA

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

```python
from datasets import Dataset, DatasetDict, load_dataset
import plotly.express as px
import plotly.io as pio
import pandas as pd
import math
import os
os.environ["TOKENIZERS_PARALLELISM"] = "false"
pio.renderers.default = 'notebook_connected'
```

FTGPT_import.py hosted with ❤ by GitHub view raw

. . .

2. Importing the baseline model and tokenizer

Then, I used `TFAutoModelForCasualLM` and `AutoTokenizer` to **automatically** load the **correct model** based on a specific **checkpoint.** A checkpoint contains the **weights of a pre-trained model.**

In this case, I imported the **DistilGPT-2 checkpoint.** I also set the **end-of-sequence token** as a padding token.

```python
tokenizer = AutoTokenizer.from_pretrained("distilgpt2")
tokenizer.pad_token = tokenizer.eos_token
model = TFAutoModelForCausalLM.from_pretrained("distilgpt2", pad_token_id=tokenizer.eos_token_id)
```

FTGPT_load.py hosted with ❤ by GitHub view raw

. . .

3. Importing Data

The dataset for the **fine-tuning** operation is available on the **Huggingface Hub**, and it's a **subset of a bigger dataset** hosted on **Kaggle.**

The original dataset, published by **Cornell University**, contains **titles** and **abstracts** of **1.7M+ scientific papers** belonging to the **STEM category.** The subset hosted on the Huggingface Hub contains information on around **100K papers** pertaining to the **machine learning category**.

I decided to **fine-tune** DistilGPT-2 on **abstracts** only**.** I started by **loading the dataset** from the Huggingface Hub.

```python
data = load_dataset("CShorten/ML-ArXiv-Papers", split='train')
data
```

FTGPT_data.py hosted with ❤ by GitHub view raw

The dataset consists of **117592 rows** and has **4 columns** (two of them are useless).

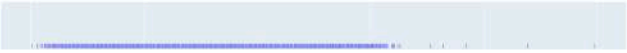
```
Dataset({
    features: ['Unnamed: 0', 'Unnamed: 0.1', 'title', 'abstract'],
    num_rows: 117592
})
```

FTGPT_data.txt hosted with ❤ by GitHub view raw

After this step, I decided to visualize the **length distribution of the abstracts** (in terms of words) with a histogram.

```python
abstracts = [len(x.split()) for x in data["abstract"]]
px.histogram(abstracts, nbins=400, marginal="rug", labels={"value":"Article Length (words)"})
```

FTGPT_plot.py hosted with ❤ by GitHub view raw



Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

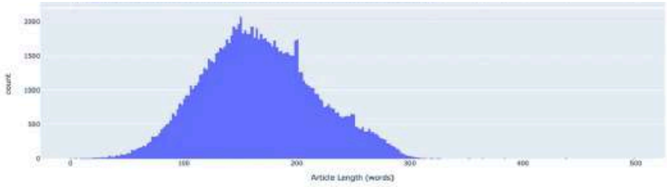
Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech



Abstracts length distribution. Image by the author.

Most of the abstracts are **between about 100 and 250 words** in length, and only a few are **over 300 words.** In particular: **mode**=150, **mean**=167, and **median**=164.

In addition to giving information about the dataset, the histogram allowed me to **determine the maximum length of the inputs** to be fed to the model.

I decided to set the **maximum input length** to **300 tokens:** abstracts longer than this **will be truncated.** This is because all inputs must be **padded to the same length**, and long sequences of text **greatly increase the training time.**

. . .

4. Split into Train and Validation Set

Next, I **split the dataset** into **train** and **validation sets** with `train_test_split()` . It is also possible to **specify the partition** sizes with the `test_size` parameter.

`train_test_split()` returns a dictionary of `Datasets` , formerly a `DatasetDict` . While it is possible to work with a `DatasetDict` , I prefer to use two separate `Datasets` : `train` and `val` .

```
data = data.train_test_split(shuffle = True, seed = 200, test_size=0.2)

train = data["train"]
val = data["test"]
```

FTGPT_split.py hosted with ❤ by GitHub view raw

. . .

5. Tokenize Data with HF Tokenizer

To tokenize the data I defined a **generic tokenization function**, and then I **applied this function to all the samples** by using `map()` . Inside the tokenization function, I used the **tokenizer** imported in the beginning.

The tokenizer has some **important parameters to set:**

1. *column to tokenize.* In this case "abstract".
2. *padding.* In this case = "max_lenght" to **pad a sequence** to the **maximum length** specified by the max_length parameter.
3. *truncation.* If true, **truncates** sequences **longer than the maximum length**, specified by the max_length parameter.
4. *max_length.* Specifies the **maximum length** of a sequence.

Please note that by default the `map()` method sends batches of 1000 samples.

```
# The tokenization function
def tokenization(data):
    tokens = tokenizer(data["abstract"], padding="max_length", truncation=True, max_length=300)
    return tokens

# Apply the tokenizer in batch mode and drop all the columns except the tokenization result
train_token = train.map(tokenization, batched = True, remove_columns=["title", "abstract", "Unnam
val_token = val.map(tokenization, batched = True, remove_columns=["title", "abstract", "Unnamed: 
```

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

FTGPT_tokenize.py hosted with ❤ by GitHub view raw

. . .

6. Adding Labels to Train and Validation Sets

In Casual Language Modeling, the labels are the **input tokens** (input_ids) **right-shifted.** This operation is **automatically done by the Huggingface transformer,** thus I created a *labels* column in the datasets with a **copy of the tokens** (input_ids).

After this operation, the train and validation sets had **three columns:** `input_ids` and `attention_mask` from the **tokenization** process, and `labels` from the `create_labels()` process.

```python
# Create labels as a copy of input_ids
def create_labels(text):
    text["labels"] = text["input_ids"].copy()
    return text

# Add the labels column using map()
lm_train = train_token.map(create_labels, batched=True, num_proc=10)
lm_val = val_token.map(create_labels, batched=True, num_proc=10)
```

FTGPT_labels.py hosted with ❤ by GitHub view raw

. . .

7. Converting Train and Validation Sets to TF Datasets

Next, I **converted the datasets** to `tf.data.Dataset`, that **Keras can understand natively**; for this purpose I used `Model.prepare_tf_dataset()`.

With respect to the `Dataset.to_tf_dataset()` method, `Model.prepare_tf_dataset()` can **automatically** determine **which column names to use as input** and provides a **default data collator.**

Note that I only **shuffled the train data**. After some experiments, I found that the **optimal batch size = 16.**

```python
train_set = model.prepare_tf_dataset(
    lm_train,
    shuffle=True,
    batch_size=16
)

validation_set = model.prepare_tf_dataset(
    lm_val,
    shuffle=False,
    batch_size=16
)
```

FTGPT_toDataset.py hosted with ❤ by GitHub view raw

. . .

8. Compiling, Fitting, and Evaluating the Model

Before fitting the model, I set up a **learning rate scheduler** and an **optimizer.** I used the `ExponentialDecay` scheduler from **Keras** and the `AdamWeightDecay` optimizer from **Huggingface.**

Learning rate decay is a technique to **reduce the learning rate over time.** With **exponential decay,** the learning rate is reduced **exponentially.**

```python
# Setting up the learning rate scheduler
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.0005,
    decay_steps=500,
    decay_rate=0.95,
```

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

```python
    staircase=False)

# Exponential decay learning rate
optimizer = AdamWeightDecay(learning_rate=lr_schedule, weight_decay_rate=0.01)
```

FTGPT_scheduler.py hosted with ❤ by GitHub view raw

Next, I **compiled the model.** Transformers models generally **compute loss internally,** and there is no need to specify a loss parameter. For **language modeling,** the selected loss is **cross-entropy.**

```python
model.compile(optimizer=optimizer)
model.summary()
```

FTGPT_compile.py hosted with ❤ by GitHub view raw

```
Model: "tfgpt2lm_head_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 transformer (TFGPT2MainLaye  multiple                 81912576
 r)

-----------------------------------------------------------------
Total params: 81,912,576
Trainable params: 81,912,576
Non-trainable params: 0
_________________________________________________________________
```

FTGPT_compile.txt hosted with ❤ by GitHub view raw

At this point, I set up a **callback to the Huggingface Hub** to **save the fine-tuned model.**

```python
# This cell is optional
from transformers.keras_callbacks import PushToHubCallback

model_name = "GPT-2"
push_to_hub_model_id = f"{model_name}-finetuned-papers"

push_to_hub_callback = PushToHubCallback(
    output_dir="./clm_model_save",
    tokenizer=tokenizer,
    hub_model_id=push_to_hub_model_id,
    hub_token="your HF token"
)
```

FTGPT_callbackHF.py hosted with ❤ by GitHub view raw

I also set up a **callback to Tensorboard.**

```python
#This cell is optional
from tensorflow.keras.callbacks import TensorBoard

tensorboard_callback = TensorBoard(log_dir="./tensorboard",
                                   update_freq=1,
                                   histogram_freq=1,
                                   profile_batch="2,10")
```

FTGPT_callbackTB.py hosted with ❤ by GitHub view raw

```python
callbacks = [push_to_hub_callback, tensorboard_callback]
```

FTGPT_callbacksAll.py hosted with ❤ by GitHub view raw

Finally, I **fitted the model** by calling the `fit()` method. **I** specified the **train** and **validation** sets and the **number of epochs.**

```python
# Fit with callbacks
model.fit(train_set, validation_data=validation_set, epochs=1, workers=9, use_multiprocessing=Tru
```

FTGPT_fit.py hosted with ❤ by GitHub view raw

After the training step, I **evaluated the model** and got its **cross-entropy loss** on the validation set.

```
eval_loss = model.evaluate(validation_set)
```

FTGPT_eval.py hosted with ❤ by GitHub view raw

Loss=2.2371. Generally, the **quality of a language model** is measured in **'perplexity'**. To convert cross-entropy to perplexity, I simply **raised** ***e*** **to the power of the cross-entropy** loss.

```
print(f"Perplexity: {math.exp(eval_loss):.2f}")
```

FTGPT_perplexity.py hosted with ❤ by GitHub view raw

In this case, **perplexity=9.37.**

. . .

9. Generating Text Using a Pipeline

At this point, I leveraged the `pipeline` functionality provided by Huggingface to **see the model in action.**

I set up a **text-generation pipeline and** specified the **fine-tuned model,** the **tokenizer,** and the **framework** to use. `max_new_tokens` allows specifying the **maximum number of tokens** (words) to generate in addition to the initial prompt provided.

```
# Setting up the pipeline
text_generator = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,
    framework="tf",
    max_new_tokens=500
)
```

FTGPT_pipeline.py hosted with ❤ by GitHub view raw

Two lines of code are enough to **generate text with a pipeline:**

```
test_sentence = "clustering"
text_generator(test_sentence)
```

FTGPT_inference.py hosted with ❤ by GitHub view raw

The `pipeline` **is not the only way** to use a model: **it** is possible to **manually tokenize the prompt, generate new tokens,** and **decode the tokens** to natural language. Here's an example:

```
input_ids = tokenizer.encode(test_sentence, return_tensors="tf")
output = model.generate(input_ids, max_length=50)
tokenizer.decode(output[0], skip_special_tokens=True)
```

FTGPT_inferenceNoPipeline.py hosted with ❤ by GitHub view raw

. . .

12. Results Analysis

After fine-tuning the model, I wanted to understand **what the model has learned** and how the **generated text is influenced** by the fact that **paper abstracts were used for training.**

First, I generated a sample text by using *"the role of recommender systems"* as a prompt. This is the **output generated by the model:**

```
'the role of recommender systems in the real-world is still largely
to be demonstrated by the lack of data and the need for data. Hence,
```

Help Status Writers Blog Careers Privacy Terms About
Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About
Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About
Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About
Text to speech

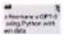
Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About
Text to speech

Norah Klintberg Sakal



```
to be demonstrated by the lack of data and the need for data. Hence,
for many recommendation systems such as Amazon or Spotify, it is
necessary to provide a user knowledge of the content that has been
clicked during the recommendation and provide a user knowledge of the
user preferences. The previous works attempt to exploit data related
to items they have clicked during an appropriate time frame. But
little attention has been paid to the problem of item classification
where a suitable time-frame is available for user prediction. In this
paper, we propose a multi-task learning approach to address the
problem of item classification. For each task, we apply the
contextual cues introduced by the user, and then learn to predict the
user's purchased items' interests.   Since the contexts of user
preferences, we consider the feature that the user's preference (the
time-frame) is present at the time of recommendation. In particular,
we propose an alternative method for attribute-aware learning that
utilizes the contextual cues in the sequence and the user's
preferences to learn a classifier that classifies the user according
to the contextual cues. This is done by maximizing the mutual
information between the user's rating and the content-aware
prediction task. The experimental results show that our model
achieves better accuracy than the existing state-of-the-art methods,
achieving up to 33.6% more accuracy on real-world recommendation
tasks compared to the state-of-the-art methods. Our source code is
available at http://github.com/J-medylerFashion/jmedian.github.'
```

This result sounded somehow copied & pasted from one of the existing abstracts, but after a check with some **anti-plagiarism** solutions, I realized that it is **100% unique.**

During learning, the model **captured common features of the abstracts** and learned **how to replicate them** while still **generating fresh text.** Interestingly, the model used **scientific language** and **common expressions**: *The previous works…, In this paper…, We propose…, The experimental result….*

The model also learned that sometimes a **repository is added to the abstract**: in this example, the text generated contains an **URL to a GitHub repository**. The URL and the repository don't exist, thus **have been generated by the model** (and not copied).

. . .

As a second experiment, I generated a sample text by using *"clustering"* as a prompt. In this case, the prompt consisted of **only one word**, so the text generation is not driven by **additional context**. This is the **output:**

```
'clustering can be used to extract clusters from data points.
However, in many real-world scenarios, data points often appear in
non-Euclidean relaxations, which allows different clusters to be
discovered simultaneously without the need for specific optimization.
In this paper, we consider a class of applications where clustering
methods are applied. A common example is that in image selection
problems, we show to the optimizer that the expected improvement will
be obtained by minimizing the average performance of the clustering
method. Our main contribution is a characterization of optimization
problems with respect to clustering methods. Specifically, we present
a generalization of the optimal transport method for which our
special bounds in terms of the maximum number of clusters are
derived. We apply this algorithm to classification of human brain
tumors, showing that it is computationally efficient. We show that
based on our analysis of a particular type of tumor, our analysis
leads to faster convergence to the optimal solution.'
```

Also in this case the text is reported to be **unique** and there are **common expressions**: *In this paper…, Our main contribution…, We present….*

. . .

For the third experiment, I used *"transformers for text generation"*. Here's the result:

```
'transformers for text generation are becoming increasingly popular
in computer vision, and so are the prevalent datasets that contain
high-dimensional representations without manual feature engineering.
We propose two algorithms for image generation from convolutional
networks. We give the first procedure to remove important parts of
this architecture and propose a novel architecture dubbed Multi-scale
Text Generation Network (MTVGNet). Our MTVGNet-like architecture
produces a compact set of representations without changing the model
```

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved…

Somnath… in JavaScript in Plai…
Coding Won't Exist In 5 Years. This Is Why

The Latest No… in data-driven fi…
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved…

Somnath… in JavaScript in Plai…
Coding Won't Exist In 5 Years. This Is Why

The Latest No… in data-driven fi…
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved…

Somnath… in JavaScript in Plai…
Coding Won't Exist In 5 Years. This Is Why

The Latest No… in data-driven fi…
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved…

Somnath… in JavaScript in Plai…
Coding Won't Exist In 5 Years. This Is Why

The Latest No… in data-driven fi…
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved…

Somnath… in JavaScript in Plai…

```
architecture or the pre-trained convolutional network. The resulting
representation is compact and can be used as both training examples
in training and inference in inference. Extensive numerical
experiments on image synthesis and computer vision demonstrate that
MTVGNet-like improves generative model performance by up to 32% over
state-of-the-art methods for unconditional image generation, while
incurring only 20% higher frame quality.'
```

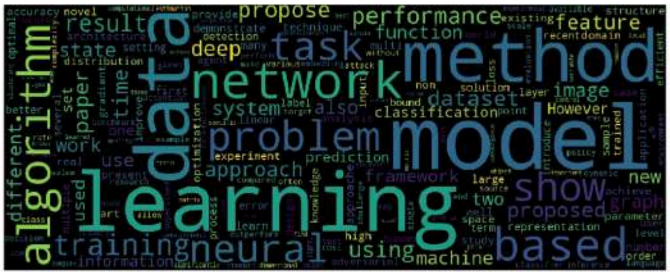
Even in the third example, there are multiple **elements common to scientific abstracts,** and the overall quality is **slightly better than the previous** one.

The cited *Multi-scale Text Generation Network (MTVGNet)* seems to be an **"invention" of the model** since I cannot find references in the literature.

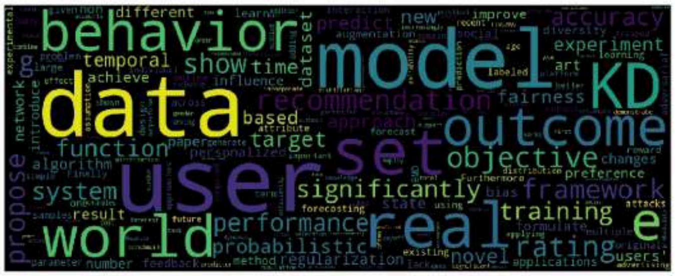
. . .

I'd like to conclude this section with some **word clouds.** The first one represents the **most frequent words in all the abstracts in the dataset.** The others depict the **most common words in 10 text samples** generated from **different prompts.**

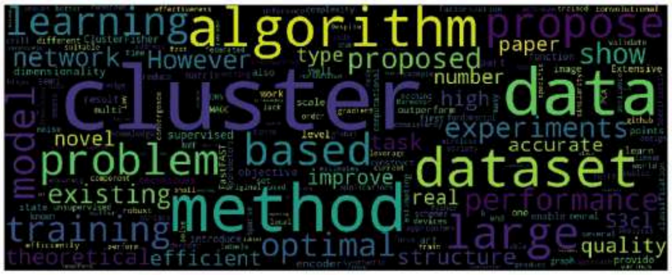
It is possible to immediately notice the **similarity of words** between the dataset **abstracts** and the **generated samples.**



Most frequent words in all the abstracts. Image by the author.



Most frequent words in the 10 AI-generated text samples. Prompt: the role of recommender systems. Image by the author.



Most frequent words in the 10 AI-generated text samples. Prompt: clustering. Image by the author.

model
corpus
based
generate
training
autoregressive
latent
StyleGAN
audio
RNN
network

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

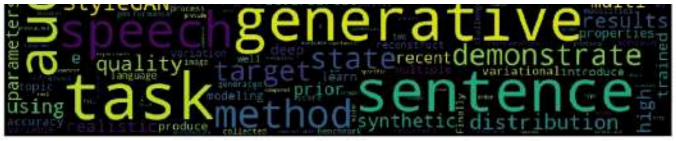
The Latest No... in data-driven fi...
Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal
How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...
Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...



Most frequent words in the 10 AI-generated text samples. Prompt: transformers for text generation. Image by the author.

. . .

Conclusions

In this article, I **fine-tuned a transformer on scientific paper abstracts.** *What is the quality of the result? What are the limitations of this approach? Is it possible to get GPT-2 to write a full paper?*

The model has learned **how abstracts are generally written** and try to **replicate the same** ***style.*** The **results are not bad,** considering the **available data** and **only one training epoch.**

The model seems to be able to **generate technical text about different machine learning topics,** but the result does **not always make complete sense,** and sometimes **there are mistakes.**

Certainly, this approach has **some limitations,** one of which is the **length of generated text.** Although it is possible to overcome the problem by **generating multiple blocks of text,** at some point, it would be **hard to logically connect** the different sections generated.

In conclusion, even if the model **cannot write an entire technical article,** I am still surprised and convinced that **some of the achievable results** can still be **inspiring** or **cueing.**

Thanks for reading!

. . .

Additional Resources

- E. Bianchi, Fine-Tuning GPT-2 for Text Generation with Tensorflow (2022), Google Colaboratory
- Hugging Face, Documentation (2022)
- Hugging Face, Distilgpt2 (2022)
- Cornell University, arXiv Dataset on Kaggle (2022)
- CShorten, ML-ArXiv-Papers dataset (2022)
- Wikipedia contributors, Perplexity (2022)
- Wikipedia contributors, Cross Entropy (2022)
- Wikipedia contributors, GPT-2 (2022)
- Keras team, Keras Documentation: ExponentialDecay (2022)

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Help Status Writers Blog Careers Privacy Terms About Text to speech

Norah Klintberg Sakal

How to fine-tune a GPT-3 model using Python with your own data for improved...

Somnath ... in JavaScript in Plai...

Coding Won't Exist In 5 Years. This Is Why

The Latest No... in data-driven fi...

Secret prompt that ChatGPT loves, with Proofs

Task-specific embedding

Multiple NLP tasks can be handled by a system, if it embeds a document in a way that's related to the task: https://pub.towardsai.net/paper-review-instructor-one-embedder-any-task-6a846b0d3ba