4222-SURYA GROUP OF INSTITUTION

VIKIRAVANDI-605 652.

NAAN MUDHALVAN PROJECT

CREATE CHATBOT IN PYTHON

PHASE 2: INNOVATION

Presented by:

M.PRITHAP

REG NO: 422221106

ECE DEPARTMENT

Introduction

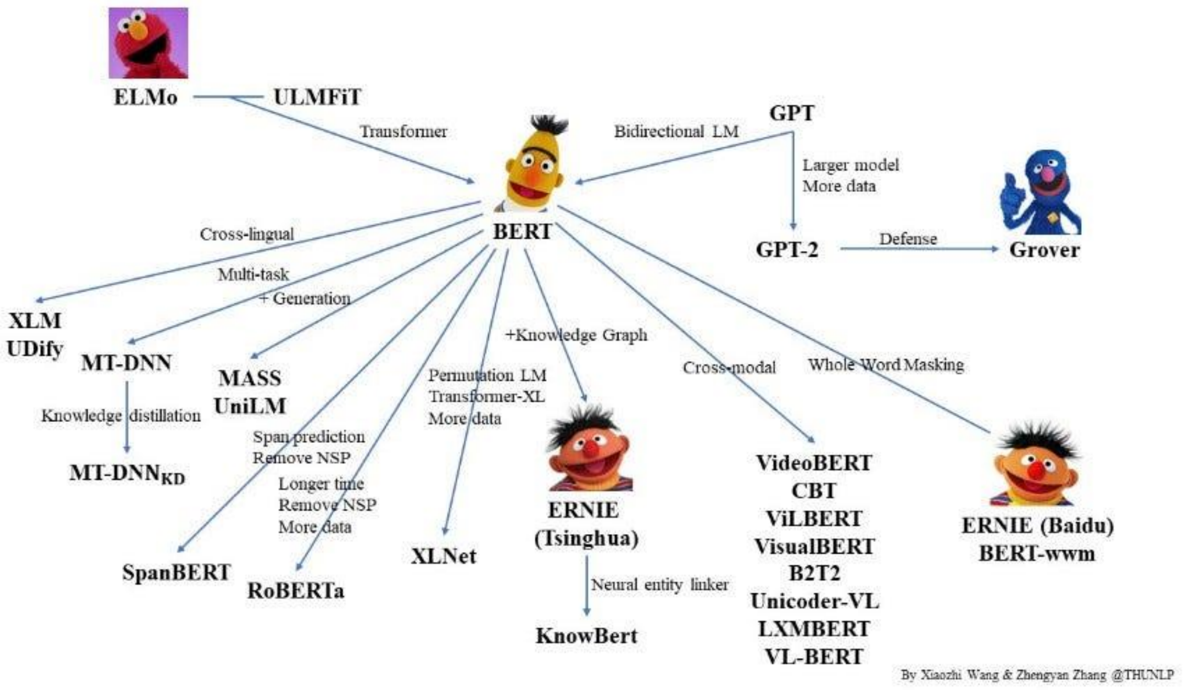
Pre-trained language models have achieved striking success in natural language processing (NLP), leading to a paradigm shift from supervised learning to pre-training followed by fine-tuning. The NLP community has witnessed a surge of research of representative work and recent progress in the NLP field and introduces the taxonomy interest in improving pre-trained models. This article presents a comprehensive review of pre-trained models. We then introduce and analyze the impact and challenges of pre-trained models and their downstream applications.

A brief history of pre-trained models

The concept of pre-training is related to transfer learning [1]. The idea of transfer learning is to reuse the knowledge learned from one or more tasks and apply it to new tasks. Traditional transfer learning employs annotated data for supervised training, which has been the common practice for at least a decade. Within deep learning, pre-training with self-supervised learning on massive unannotated data has become the dominant transfer learning approach. The difference is that pre-training methods use unannotated data for self-supervised training and can be applied to various downstream tasks via fine-tuning or few-shot learning.

Although methods that leverage static word embeddings for warm startup can improve the performance of downstream NLP tasks, they lack the ability to represent different meanings of words in context. To solve this problem, context–aware language models were proposed to incorporate the complete context information into the training procedure.

Since then, numerous PTMs **within the "pre-training then fine-tuning" paradigm have started to emerge. GPT [23], the** first model to leverage bidirectional transformers was called Bidirectional Encoder Representations from Transformers (BERT); this model learns bidirectional contexts by means of conditioning on both the left and the right contexts in deep stacked layers. BERT introduced a denoising autoencoding pre-training task, termed masked language modeling (MLM), to recover the corrupted tokens of input sentences according to their contexts, in what was akin to a cloze task. This approach greatly boosted the performance gain of downstream natural language understanding (NLU) tasks Although the most commonly used pre-training tasks for multimodal context are MLM and masked region prediction, Yu et al.



About GPT-3

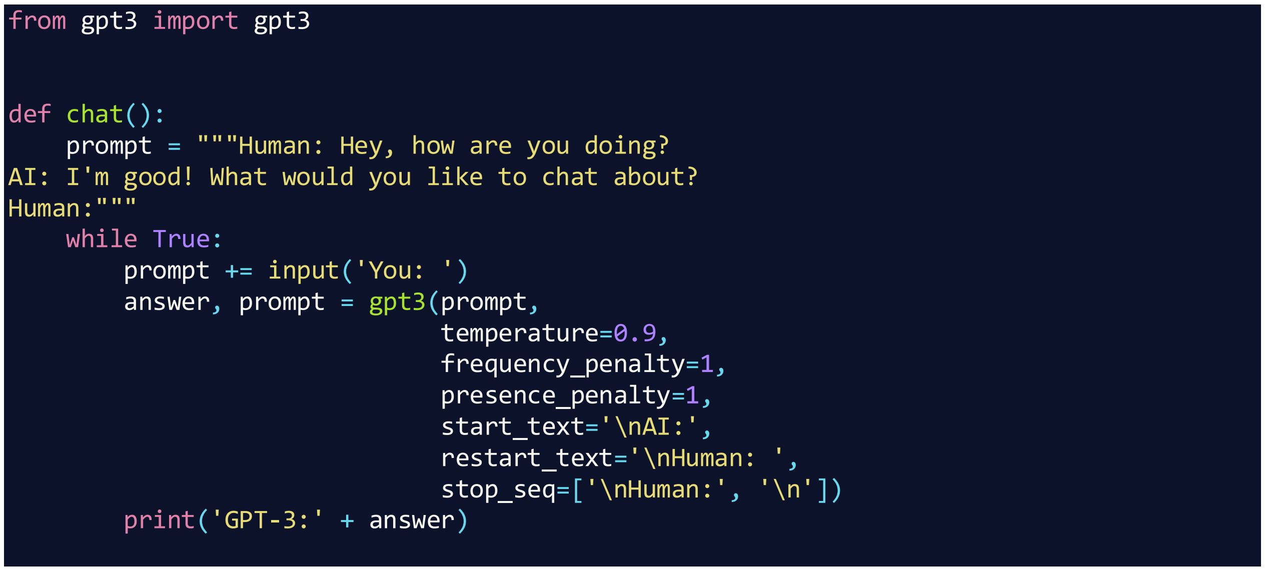
GPT-3, or the third-generation Generative Pre-trained Transformer, is a neural network machine learning model trained using internet data to generate any type of text. Developed by OpenAI, it requires a small amount of input text to generate large volumes of relevant and sophisticated machine-generated text.

GPT-3's deep learning neural network is a model with over 175 billion machine learning parameters. To put things into scale, the largest trained language model before GPT-3 was Microsoft's Turing Natural Language Generation (NLG) model, which had 10 billion parameters. As of early 2021, GPT-3 is the largest neural network ever produced. As a result, GPT-3 is better than any prior model for producing text that is convincing enough to seem like a human could have written it.

What can GPT-3 do

GPT-3 processes text input to perform a variety of natural language tasks. It uses both natural language generation and natural language processing to understand and generate natural human language text. Generating content understandable to humans has historically been a challenge for machines that don't know the complexities and nuances of language. GPT-3 has been used to create articles, poetry, stories, news reports and dialogue using a small amount of input text that can be used to produce large amounts of copy.

GPT-3 can create anything with a text structure -- not just human language text. It can also generate text summarizations and even programming code



```python
from gpt3 import gpt3


def chat():
    prompt = """Human: Hey, how are you doing?
AI: I'm good! What would you like to chat about?
Human:"""
    while True:
        prompt += input('You: ')
        answer, prompt = gpt3(prompt,
                              temperature=0.9,
                              frequency_penalty=1,
                              presence_penalty=1,
                              start_text='\nAI:',
                              restart_text='\nHuman: ',
                              stop_seq=['\nHuman:', '\n'])
        print('GPT-3:' + answer)
```

```
if __name__ == '__main__':
    chat()
```

GPT-3 examples

One of the most notable examples of GPT-3's implementation is the ChatGPT language model. ChatGPT is a variant of the GPT-3 model optimized for human dialogue, meaning it can ask follow-up questions, admit mistakes it has made and challenge incorrect premises. ChatGPT was made free to the public during its research preview to collect user feedback. ChatGPT was designed in part to reduce the possibility of harmful or deceitful responses.

Another common example is Dall-E. Dall-E is an AI image generating neural network built on a 12 billion-parameter version of GPT-3. Dall-E was trained on a data set of text-image pairs and can generate images from user-submitted text prompts. ChatGPT and Dall-E were developed by OpenAI.

➢ Transformer decoders only

❖ The objective for language modeling is to predict the next token auto-regressively, given its history. The nature of auto-regression entails the future invisibility of input tokens at each position; that is, each token can only attend to the preceding words. GPT [22] was the first model to use the transformer decoder architecture as its backbone.

❖ n the fine-tuning phase, the pre-trained parameters are set as the initialization of the model for downstream tasks. GPT is pre-trained on the BooksCorpus dataset, which is nearly the same size as the 1B Word Benchmark. It has hundreds of millions of parameters and improves SOTA

/results on nine out of 12 NLP datasets, showing the potential of large-scale PTMs.

➢ Transformer encoders only

Pre-trained transformer encoders, such as BERT [23], have become the standard in NLP systems. BERT uses an MLM framework with a transformer as the backbone. In the pre-training stage, BERT randomly replaces tokens with a special token [MASK] and tries to recover corrupted words based on their contextual representations. It also adopts an objective of next-sentence prediction (NSP) to capture the discourse relations between two sentences, which is helpful for sentence-level tasks, such as question answering.

Another problem with BERT is that it predicts tokens independently without considering other masked tokens.

```
encoder_input = np.array(df.question)
decoder_input = np.array(df.decoder_input)
decoder_label = np.array(df.decoder_label)

n_rows = df.shape[0]
print(f"{n_rows} rows")

indices = np.arange(n_rows)
np.random.shuffle(indices)

encoder_input = encoder_input[indices]
decoder_input = decoder_input[indices]
decoder_label = decoder_label[indices]

train_size = 0.9

train_encoder_input = encoder_input[:int(n_rows*train_size)]
train_decoder_input = decoder_input[:int(n_rows*train_size)]
train_decoder_label = decoder_label[:int(n_rows*train_size)]

test_encoder_input = encoder_input[int(n_rows*train_size):]
test_decoder_input = decoder_input[int(n_rows*train_size):]
test_decoder_label = decoder_label[int(n_rows*train_size):]

print(train_encoder_input.shape)
print(train_decoder_input.shape)
print(train_decoder_label.shape)

print(test_encoder_input.shape)
print(test_decoder_input.shape)
print(test_decoder_label.shape)
```

➢ Transformer encoder–decoders

Transformer encoder–decoder generate a coherent, meaningful, and human-like natural language expression according to specific inputs. For example, the goal of machine translation is to generate a sentence in the target language with the same meaning as the given source language input; for text summarization, the goal is to generate a short version of the input document that captures the core meanings and opinions. The critical point is to model two sequences simultaneously—one for the input and the other for the output.architecture is dedicated to natural language generation (NLG) tasks. Unlike NLU.

Platforms and toolkits for applications

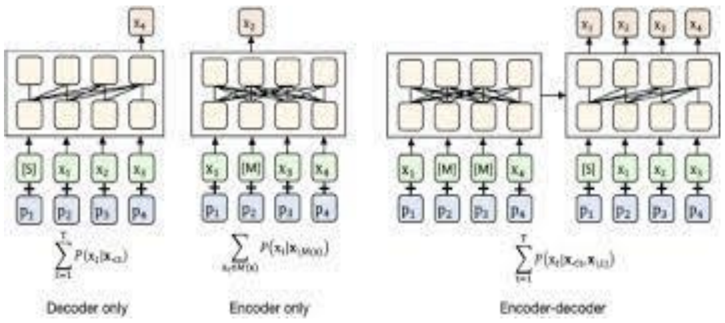
Due to their universality, PTMs have become foundation models in NLP. Many researchers have developed a series of open-source toolkits and platforms to make better use of PTMs. These toolkits and platforms usually contain various PTMs, fine-turning tools, and model-compression tools.

Model	**Number of parameters**	**Model architecture**	**Language**	**Pre-training data**	**Training strategy Training platform**	**Training platform**
DeBERTa1.5B	1.5billion	Encoder only	English	English data (78 GB)	-	PyTorch
T5	11billion	Encoder–decoder (seq2seq)	English	C4 (750 GB)	Model/data parallelism	TensorFlow
GPT-3	175billion	Decoder only	English	Cleaned CommonCrawl, WebText	Model parallelism	-
CPM	2.6billion	Decoder only	Chinese	Chinese corpus (100 GB)	-	PyTorch
PanGu-α	200billion	Decoder only	Chinese	Chinese data (1.1 TB, 250 billion tokens)	MindSpore auto-parallel	MindSpore

Future of GPT-3

There are many open source efforts in play to provide a free and non-licensed model as a counterweight to Microsoft's exclusive ownership. New language models are published frequently on Hugging Face's platform.

It is unclear exactly how GPT-3 will develop in the future, but it is likely that it will continue to find real-world uses and be embedded in various generative AI applications. Many applications already use GPT-3, including Apple's Siri virtual assistant. Where possible, GPT-4 will be integrated where GPT-3 was used.



Conclusions and future work

PTMs can fully exploit unannotated data for self-supervised learning and have become the foundation models in NLP, significantly improving the performance of downstream NLP tasks. The emergence of PTMs opens up a new "pre-training then fine-tuning" paradigm for NLP.

Another promising direction is to incorporate prior knowledge into PTMs to improve their reasoning abilities and efficiency. Existing work on knowledge pre-training, such as K-BERT [33] and ERNIE 3.0 [39], has injected knowledge triplets into pre-training or fine-tuning. However, PTMs have demonstrated limited capability for commonsense awareness and reasoning, there is still a long way to go for PTMs to be able to make reliable decisions and carry out reliable planning, which are essential elements of AI. More efficient and powerful neural networks need to be proposed and developed. Fortunately, the use of PTMs in real applications continues to provide an increased amount of data and address new challenges, potentially promoting the rapid development of new pre-trained methods.