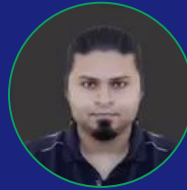


Advanced NLP

Deep Transfer Learning for Natural Language Processing with Transformers

Dipanjan (DJ) Sarkar

Your speaker for today



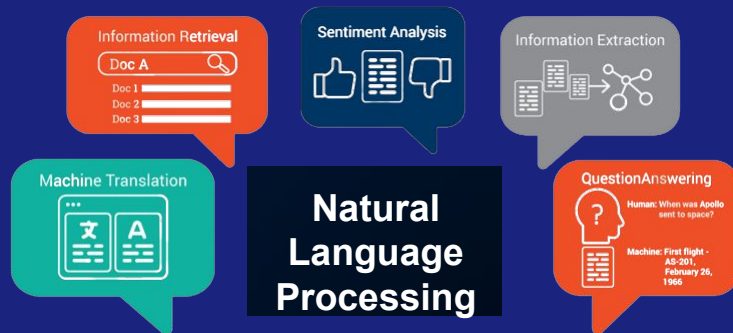
Dipanjan (DJ) Sarkar

Lead Data Scientist,
Consultant & Author

NLP Applications

- NLP has a wide range of applications, including:

- Sentiment analysis
- Text categorization or classification
- Language translation
- Speech recognition
- Question answering



- NLP is used in almost every industry including social media, healthcare, finance, marketing, and more

NLP Applications - Language Translation



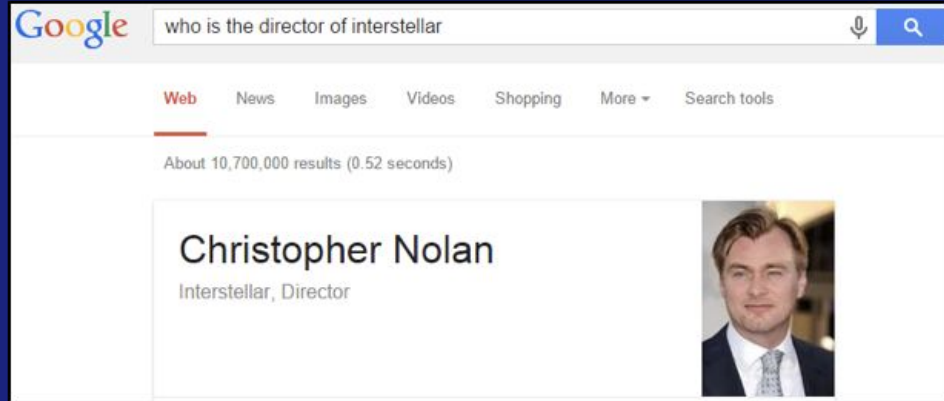
这是一个问题，但也不敢相信的表情。
这些谁迷路驾驶可以使用GPS。如果你失去了你的iPhone，有一个应用程序来追查。科学家成功绘制过程中的飞船降落在小行星飞驰。
没有天气如何影响亚航的班机吗？
但不顺心的事一艘123英尺，67吨重的喷气式客机和救援人员必须求助于淘海洋？
“为什么更容易找到一个iPhone（比）找到飞机？”1 Twitter的用户，卡特丽娜Buitano，问道。
有几十个在社交媒体上类似的问题。他们暗示相同的感悟：在这个世界上，人的位置进行跟踪，一切从地图应用程序，以广告出现在网页浏览器，为什么大哥的目光避开天空是什么？



NLP Applications - QA Systems



Apple Siri



Google search

NLP Applications - NER

Locating entities
in unstructured
text

April attended KDD 2019 in
Anchorage

Person

Event



April attended KDD 2019 in
Anchorage.



GPE

NLP Applications - Sentiment Analysis

Understanding
sentiment from
text

This tutorial is great!

Positive

I do not dislike this tutorial

Neutral

This is a perfect example of what a
tutorial should not be

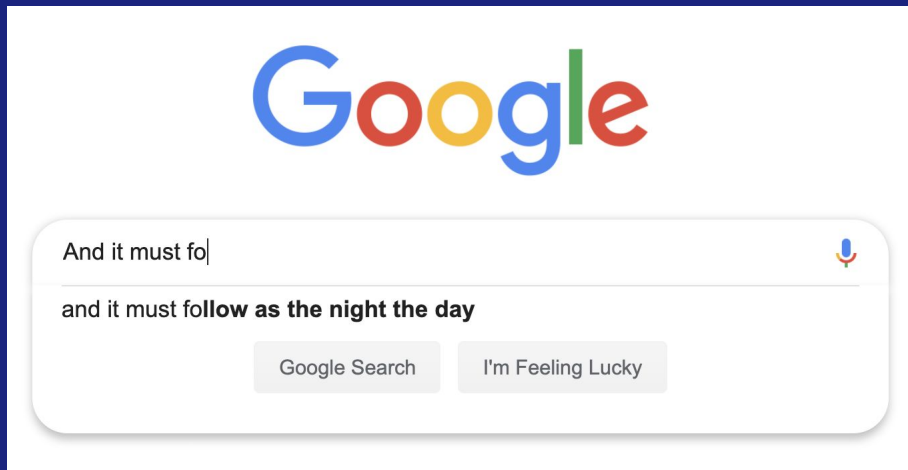
Negative

NLP Applications - Language Modeling

*"And it must follow, as the night the day,
Thou canst not then be false to any man.
Farewell: my blessing season this in thee!"*

<http://shakespeare.mit.edu/hamlet/full.html>

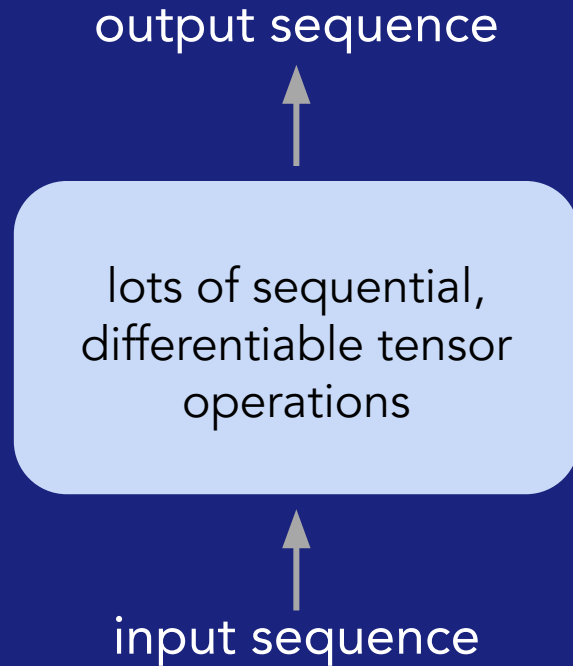
Models that predict
tokens from a
sequence of
previous ones.



<https://www.google.com/>

Generative NLP in a nutshell

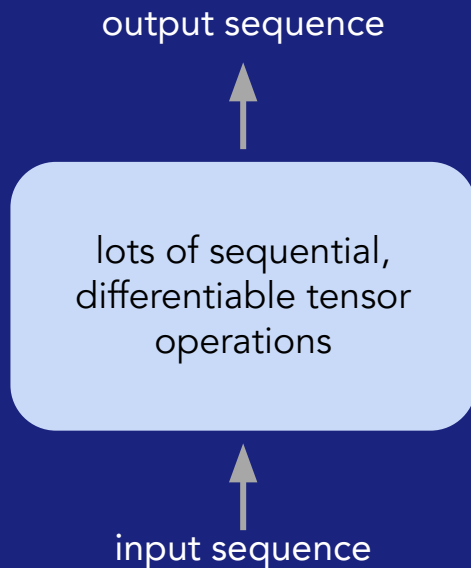
A versatile set of models that can be used to process and generate sequential data



Generative NLP - Sequence to Sequence Models

Examples of

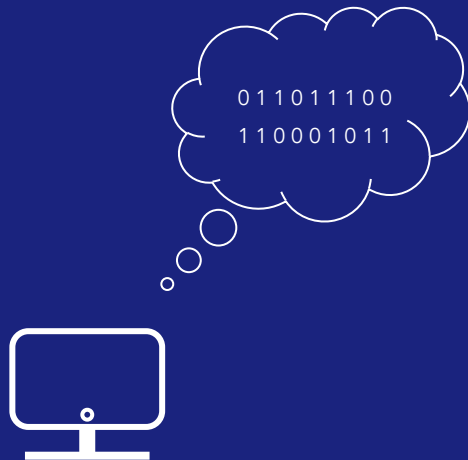
sequence-to-sequence models.



	inputs	outputs
Sentiment Analysis	Incredibly creative!	★★★★★
Machine Translation	Incredibly creative!	Incrivelmente criativo!
Dialog systems	Incredibly creative!	Thank you
Speech recognition		Incredibly creative!

Two big questions?

How can we make numeric representations out of documents?



Two big questions?

What sorts of models are better suited for processing sequential data?




How to represent words and documents?

Embeddings

Embeddings: model
learnt latent
representations of
words

embedding size (\ll vocab size)

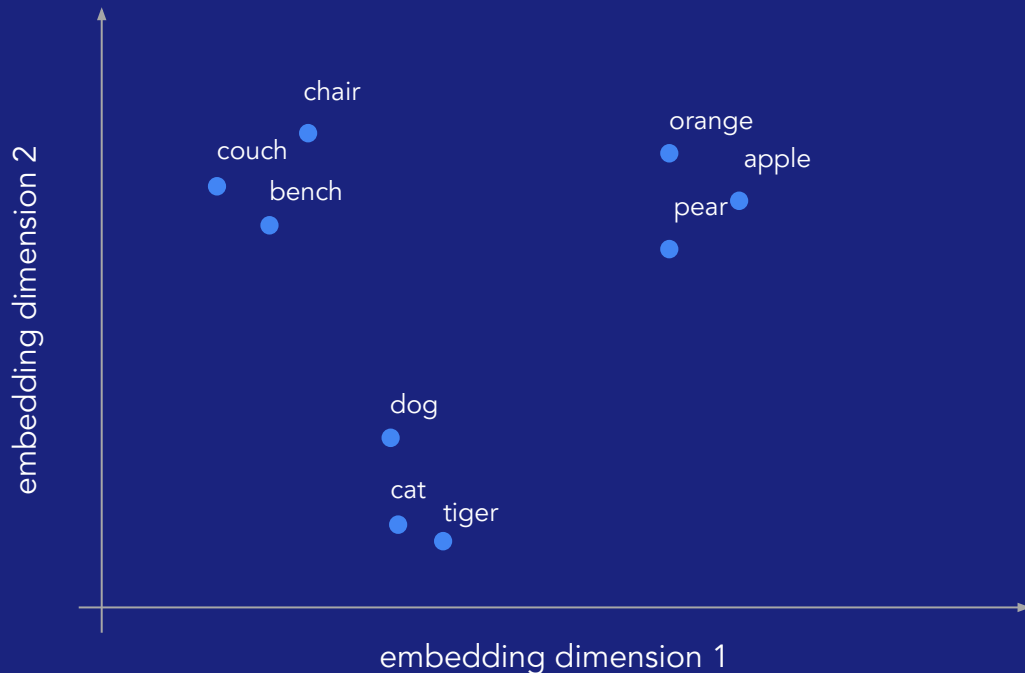


a	→	.3452 .7162 .1827 .9382 .9182 ...
ability	→	.1234 .8172 .6473 .5630 .0263 ...
able	→	.1263 .8054 .5632 .5589 .0374 ...
about	→	.7364 .2039 .2831 .2837 .1923 ...
above	→	.9283 .0023 .0065 .2938 .5472 ...
acarus	→	.1938 .2938 .0293 .5647 .2348 ...
•		•
•		•
•		•

How to represent words and documents?

Embeddings

An illustrative point



Models? Language Models

- Language Modeling is the task of predicting what word comes next



Current State of NLP

- In recent years, NLP has seen a surge in advancements due to the development of deep learning techniques and the availability of large language datasets.
- These advancements have led to significant progress in several NLP applications thanks to the advent of sequential transfer learning and transformers
- Transfer learning involves the use of already pre-trained HUGE models, most notably transformers, and adapting it to solve diverse problems, including machine translation, sentiment analysis, and question-answering systems

Transfer Learning

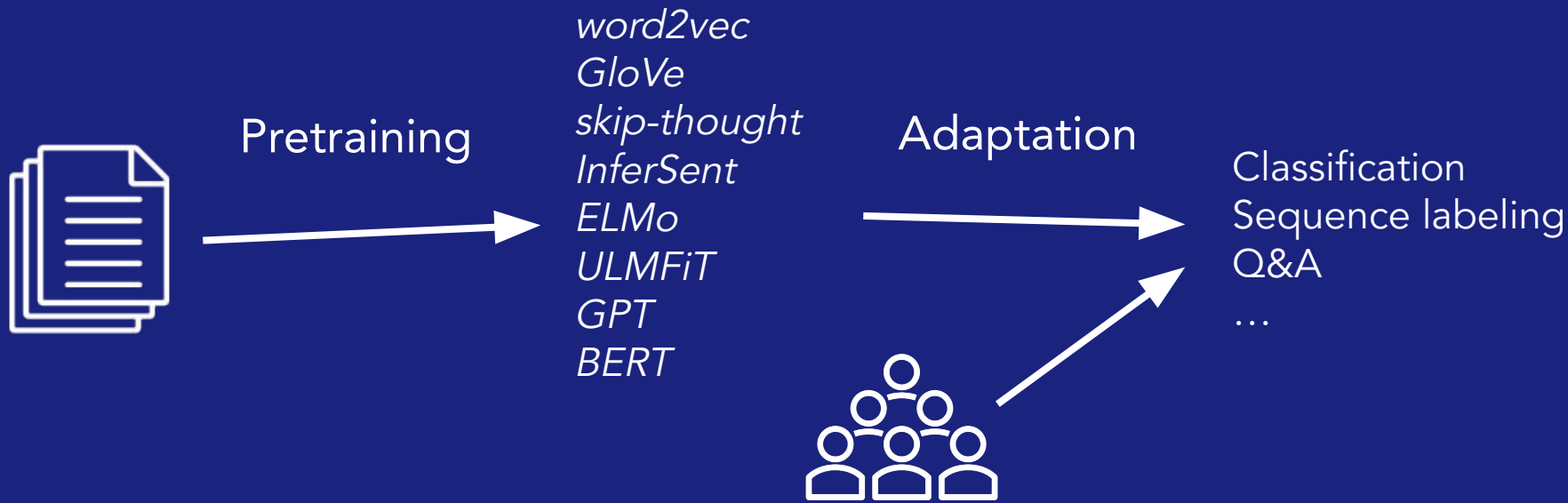
The ability to utilize knowledge learnt in prior tasks to new and novel tasks



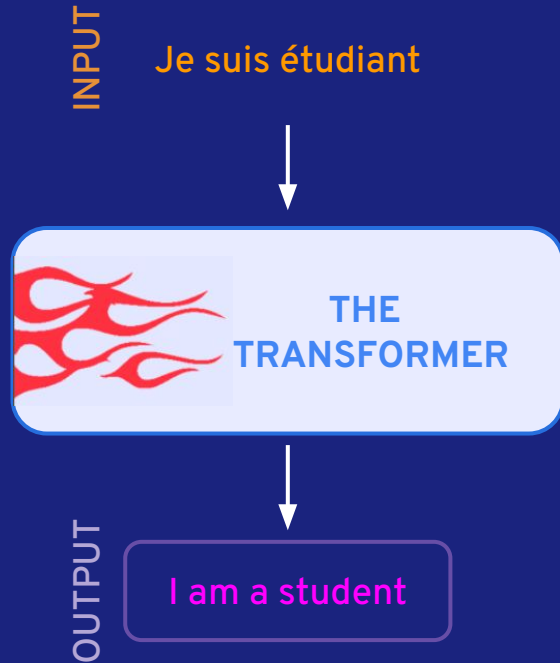
- Know how to ride a motorbike
➡ Learn how to ride a car
- Know how to play classic piano
➡ Learn how to play jazz piano
- Know math and statistics ➡ Learn machine learning

Sequential Transfer Learning for NLP

Sequential transfer learning has led to the biggest improvements so far in the field of NLP

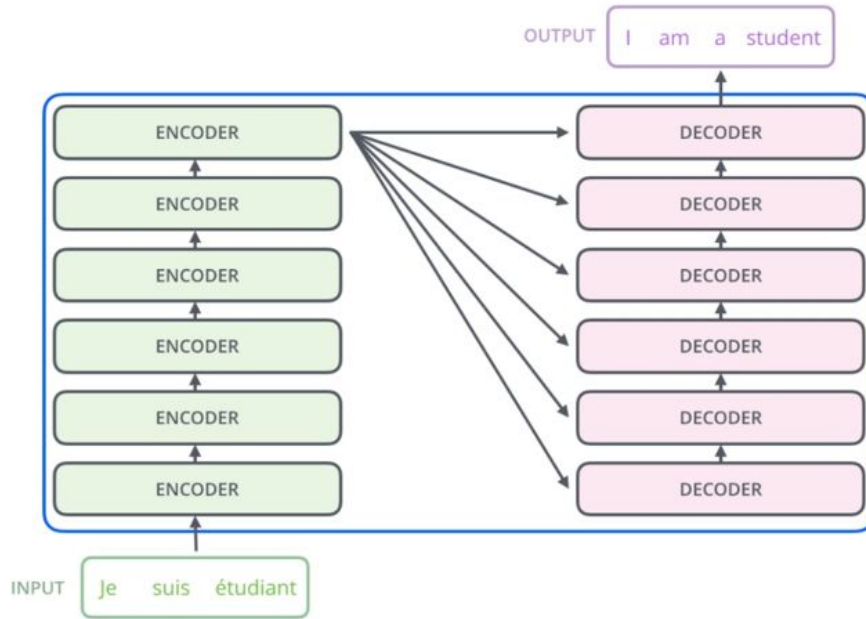


Biggest success for NLP - Transformers



- Layered & Stacked Encoder-Decoder Model
- Relies on Multi-headed Self-Attention & Encoder-Decoder Attention
- No sequential RNN-based training (completely parallelized)
- Achieved state-of-the-art performance on several NLP tasks

Transformer Model Architecture



- At-heart a transformer model is a stacked encoder-decoder model
- Has 6-12 stacked encoder and decoder blocks usually based on standard architectures
- Based on the type of transformer model (only encoder \ decoder or both are used)

Transformer Model Architecture

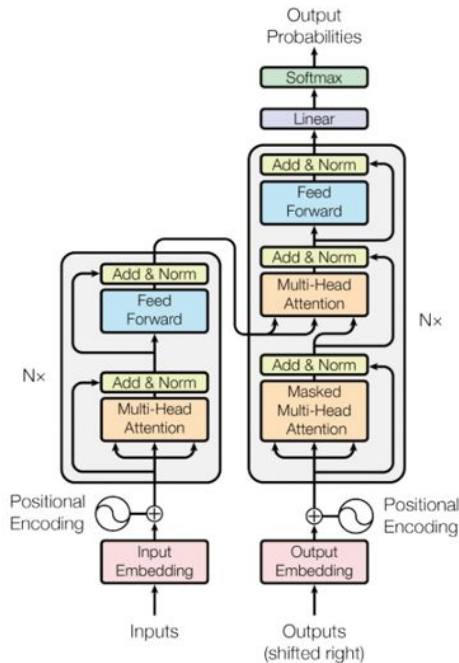
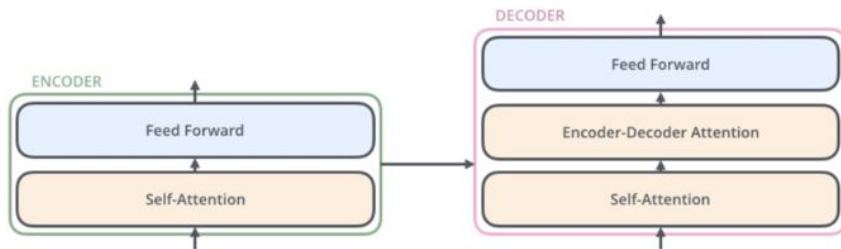


Figure 1: The Transformer - model architecture.

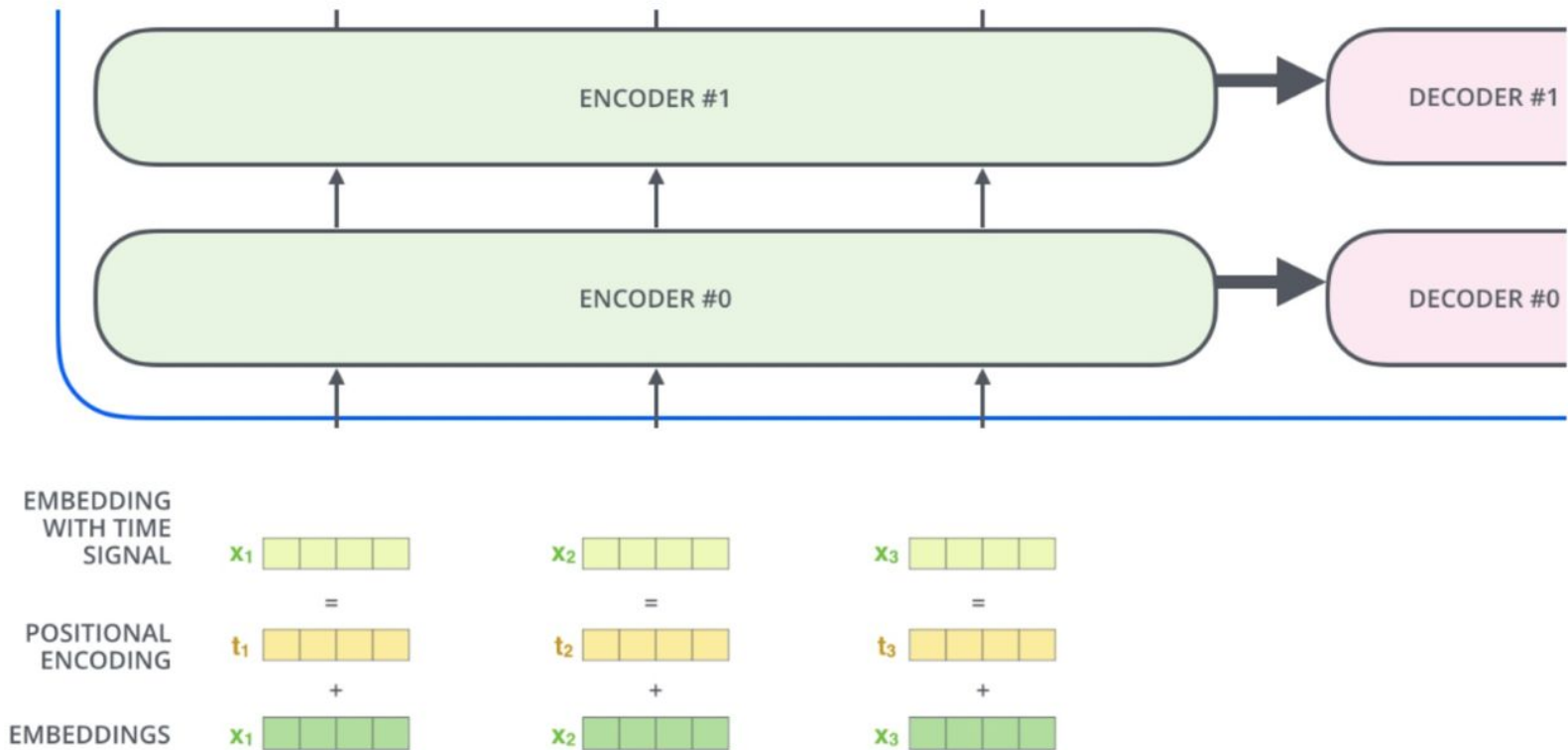
- Major components of each encoder and decoder layer includes self-attention and multi-headed attention layers
- Feed-forward dense layers are also introduced after these layers
- Residual connections help in propagating inputs (and intermediate outputs) further along the network and gradients further back

Transformer Model Architecture



- The encoder's inputs flow through a self-attention layer
 - Each word attends (looks at) every other word in this process
- The outputs of the self-attention layer are fed to a feed-forward neural network
- The decoder has similar layers
- Between them is an attention layer that helps the decoder focus on relevant parts of the input sentence
 - This is the encoder-decoder attention layer

Encoding Sequential Representations in Transformers

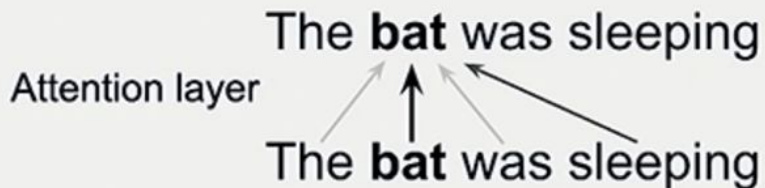


Self Attention: A simple overview

Attention layer

The **bat** was sleeping

The **bat** was sleeping



- What does “bat” in this sentence refer to? Is it referring to a baseball bat \ cricket bat or the animal?
- When the model is processing the word “bat”, self-attention allows it to associate ”bat” with “sleeping” strongly
 - This gives it the indication that it could be an animal
- Self attention allows the model to look at other positions in the input sequence for clues that can help lead to a better encoding for each word

Multi headed Self Attention: Encode different aspects

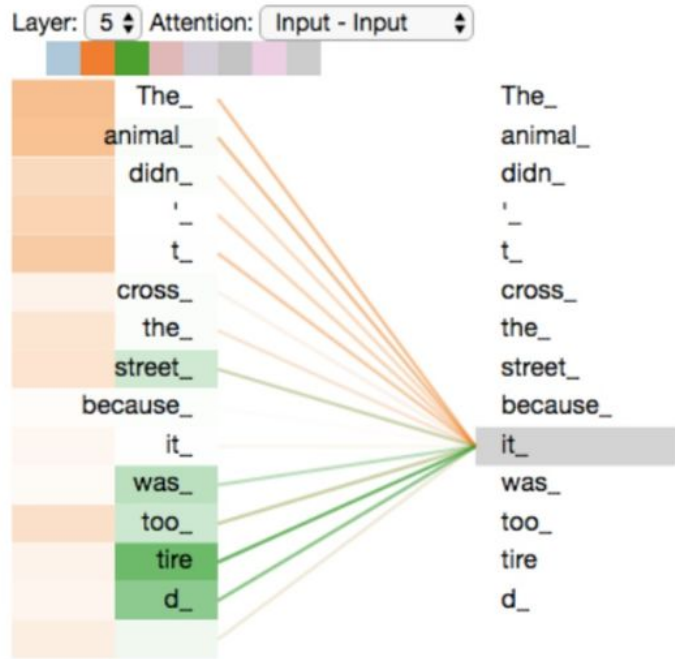


Fig. Source: <https://jalammar.github.io/illustrated-transformer>

As we encode the word "it", one attention head is focusing most on "the animal", while another is focusing on "tired" -- in a sense, the model's representation of the word "it" bakes in some of the representation of both "animal" and "tired".

Transformer Architectures

1 Autoregressive Models

- Correspond to the decoder of the original transformer model
- Mask is used on top of the full sentence so that the attention heads can only see previous words
- Pretrained on the classic language modeling task: guess the next token
- Example - GPT family

2 Autoencoding Models

- They correspond to the encoder of the original transformer model
- They get access to the full inputs without any mask
- Pretrained by corrupting the input tokens in some way and trying to reconstruct the original sentence (Masked Language Modeling)
- Example - BERT family

3 Sequence to Sequence Models

- Uses both the encoder and the decoder of the original transformer
- Most popular applications are translation, summarization and question answering
- Example - BART, T5

4 Multi-modal Models

- Mixes data of different modalities e.g text & images

Autoencoding Transformer - BERT

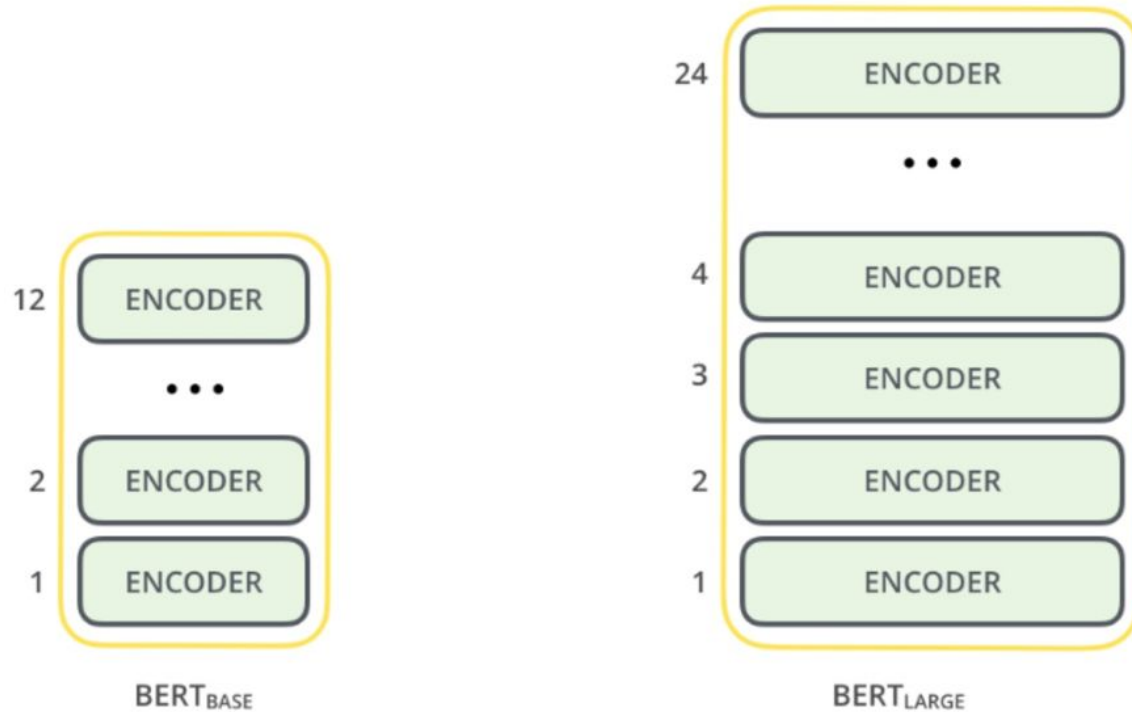
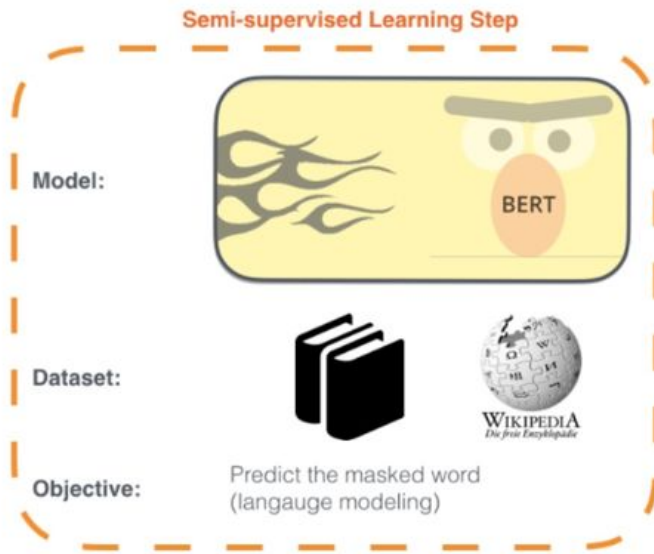


Fig. Source: <https://jalammar.github.io/illustrated-bert>

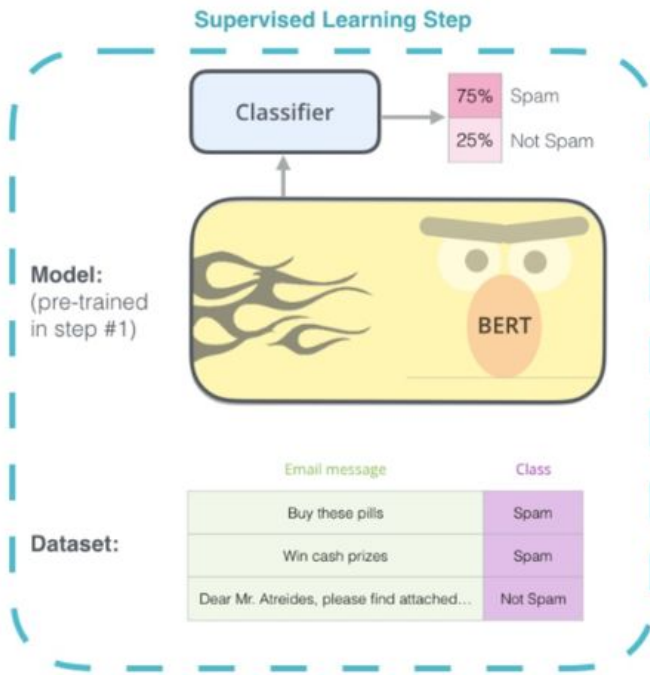
BERT Training Workflow

1 - **Semi-supervised** training on large amounts of text (books, wikipedia..etc).

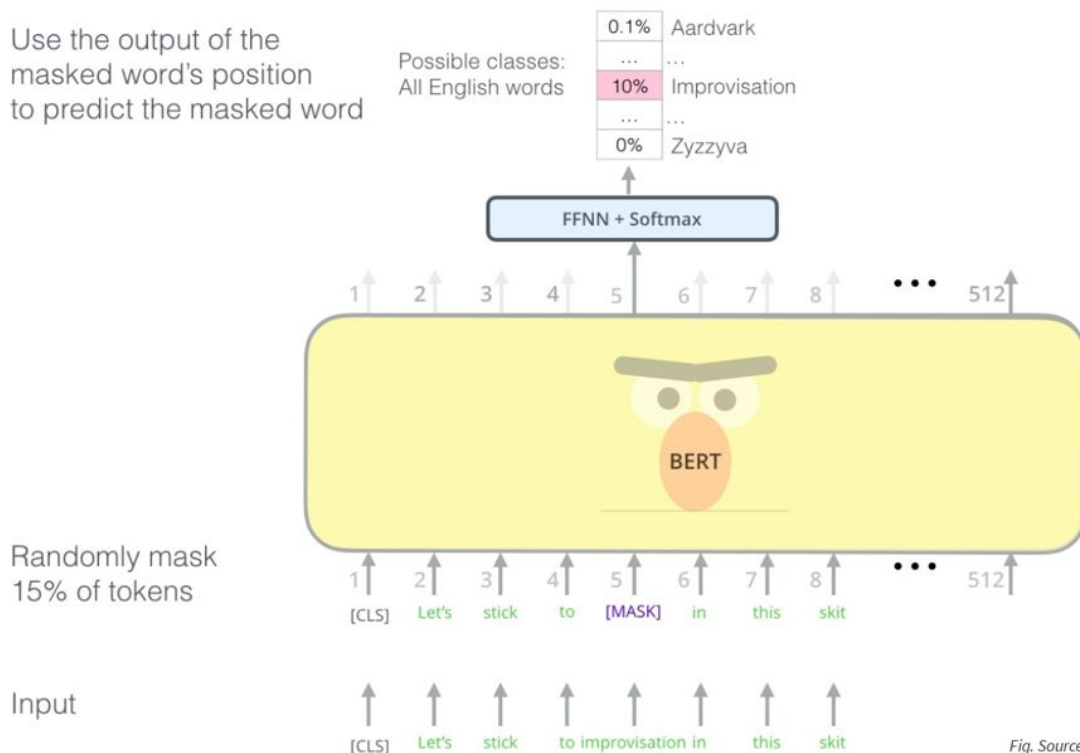
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



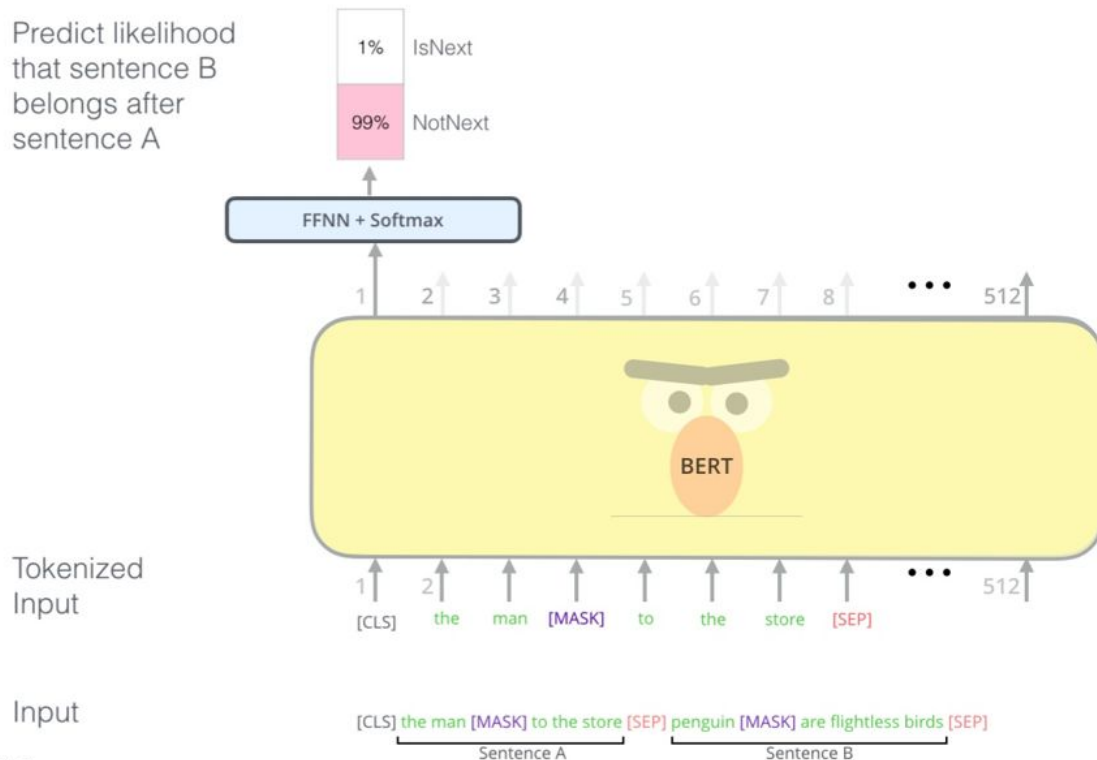
2 - **Supervised** training on a specific task with a labeled dataset.



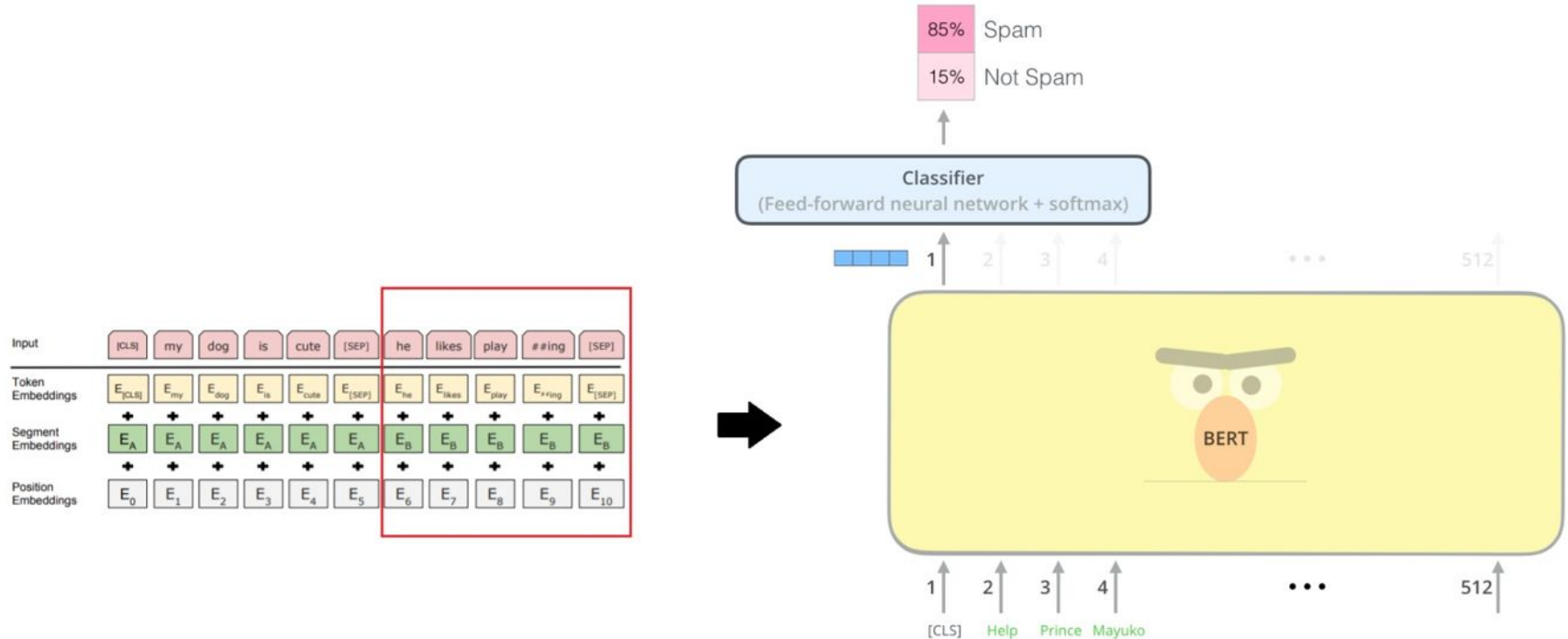
BERT Training Workflow - Masked Language Modeling



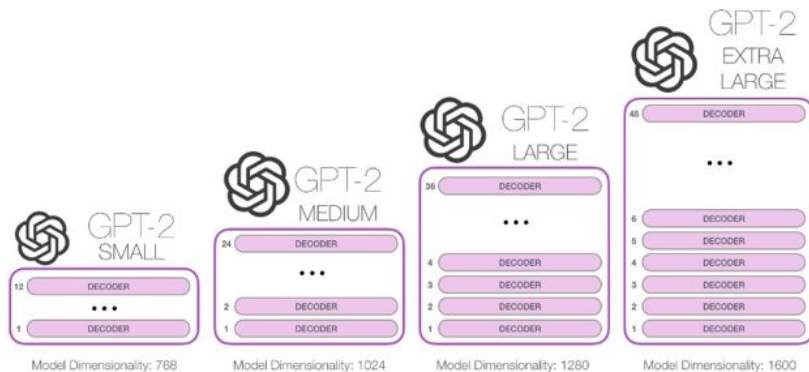
BERT Training Workflow - Next Sentence Prediction



Fine-tuning BERT - Text Classification

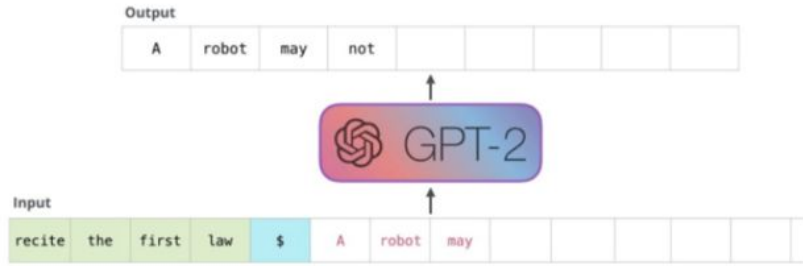


Autoregressive Transformer - GPT



- GPT stands for Generative Pre Training
- Stacked Decoder Model which is a Transformer
- Trained on the classic language modeling task of 'Next Word Prediction'
- Trained on the massive 40GB WebText Dataset

GPT2 Essentials

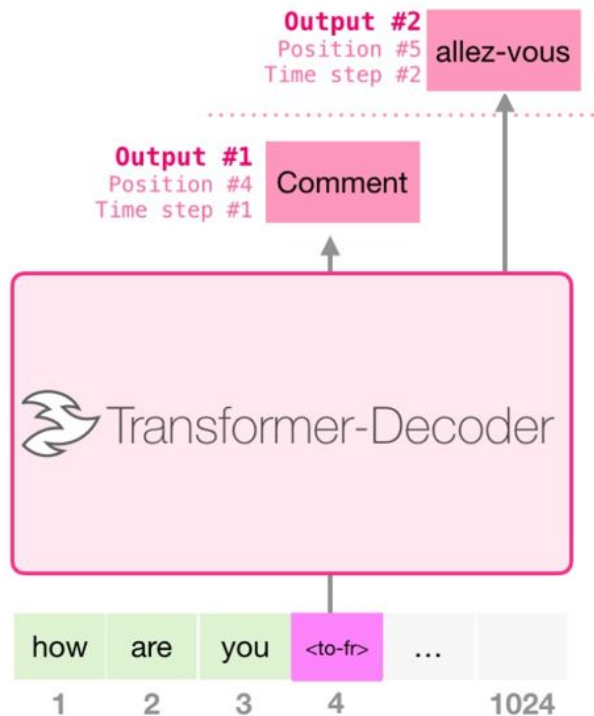


- Auto-regressive model built out of transformer decoder blocks
- GPT-2 generates one token at a time like any other language model
- Generated tokens can be added back to the input sequence to generate further tokens (auto-regression)
- GPT-2 Large has over 1.5 Billion parameters

GPT2 Fine-tuning - Language Translation

Training Dataset

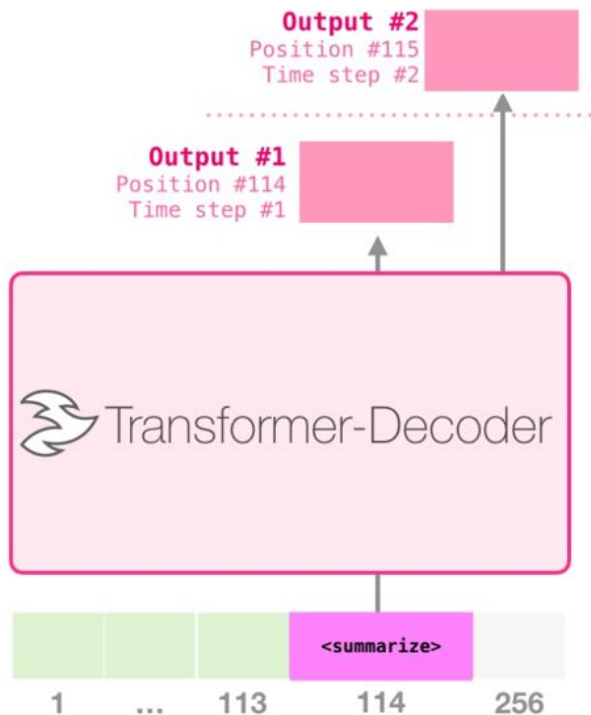
I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				



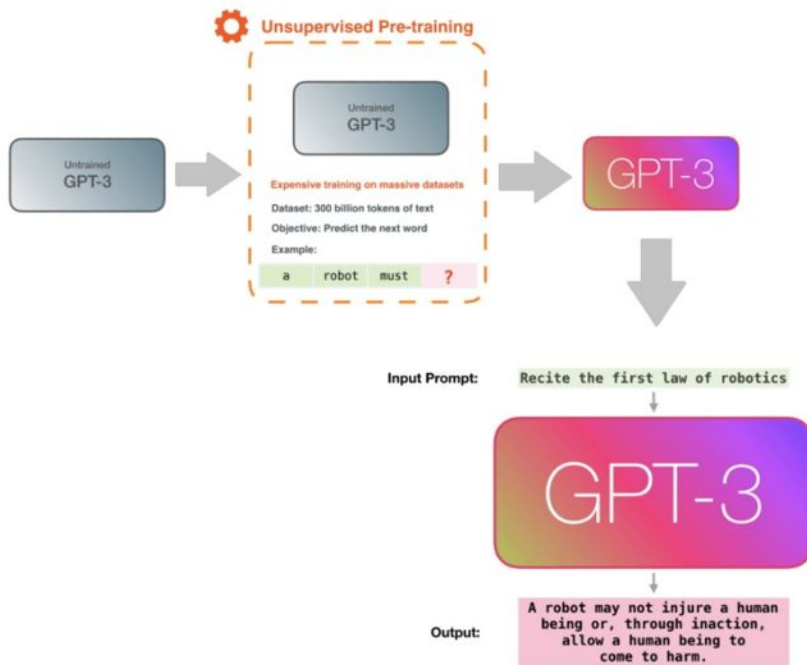
GPT2 Fine-tuning - Text Summarization

Training Dataset

Article #1 tokens	<summarize>	Article #1 Summary	
Article #2 tokens	<summarize>	Article #2 Summary	padding
Article #3 tokens	<summarize>	Article #3 Summary	



GPT3 Essentials



- GPT-3 is one of the latest generative transformer models built by OpenAI
- Massive model consisting of 96 stacked decoder layers
- Over 175 billion model parameters
- Each decoder block has around 1.8 billion parameters
- Trained as a language model (predict the next word objective)
- **Expensive training phase!**
 - Trained on over 300 billion tokens
 - 355 GPU years
 - \$4.6 million cost

What about ChatGPT-based models?

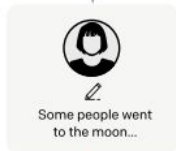
Step 1

Collect demonstration data, and train a supervised policy.

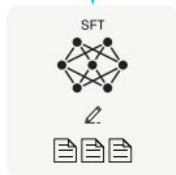
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



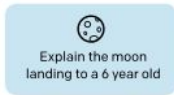
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

Collect comparison data, and train a reward model.

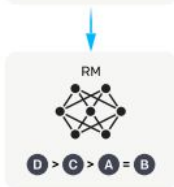
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



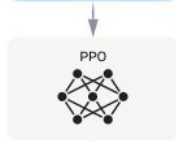
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

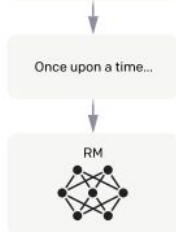


The policy generates an output.



Once upon a time...

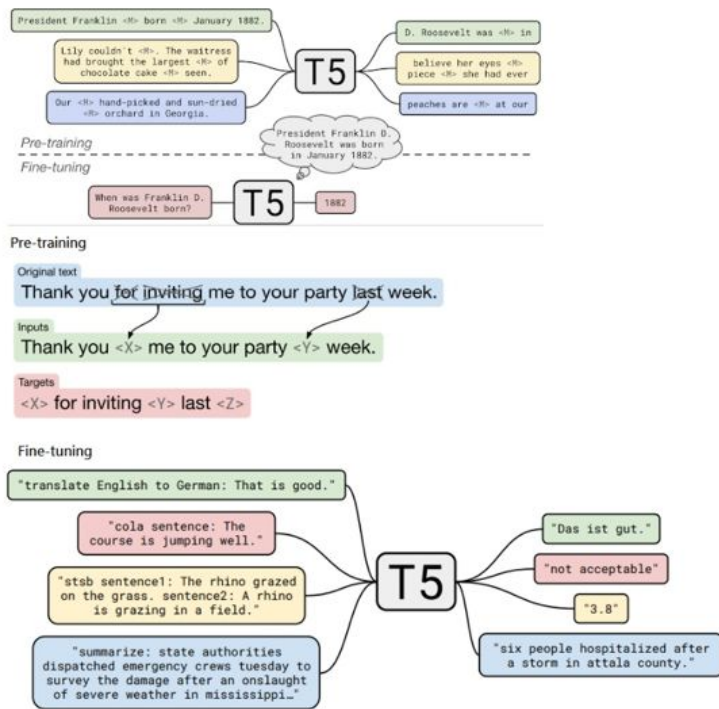
The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Sequence to Sequence Transformers - T5



- T5 stands for Text-to-Text Transfer Transformer
 - Uses a standard encoder-decoder transformer architecture similar to BERT-Base
- Trained on a massive dataset called the Colossal Clean Crawled Corpus (C4) which is almost 750 GB
- T5 combines pre-training first and then fine-tuning for a variety of diverse NLP tasks using the same model
 - Pre-trained on the classic task of masked language modeling called as unsupervised denoising training
 - Fine-tuned on diverse tasks by combining them using a text-to-text format
 - Text-to-text framework suggests using the same model, loss, hyperparameters on all the NLP tasks by formatting the inputs in a specific way
 - For supervised fine-tuning tasks the task name is usually mentioned at the beginning of the input sequence, e.g., **Input Sequence:** "translate English to German: That is good" and **Output Sequence:** "Das ist gut"
- Conducted a variety of experiments to explore transfer learning methodologies
 - encoder-decoder models generally outperformed "decoder-only" language models
 - fill-in-the-blank-style denoising objectives (predict missing words) worked best
 - training on in-domain data can be beneficial
 - pre-training + fine-tuning approaches vs. multi-task learning

Live Demo 1

- **Multi-task NLP with Transformers**

- Sentence Classification (Sentiment Analysis)
- Question-Answering
- Summarization
- Translation
- No-training categorization (Zero-shot classification)

- Get the notebook [from here](#)

- Run it on Colab by copying to your drive or local system & follow along with the speaker
- Recommended to run on Google Colab



Live Demo 2

- **Question-Answering Search Engines with Transformers**
- Get the notebook [from here](#)
- Run it on Colab by copying to your drive or local system & follow along with the speaker
- Recommended to run on Google Colab



Live Demo 3

- **Build your own text classifier with Transformers**
- Get the notebook [from here](#)
- Run it on Colab by copying to your drive or local system & follow along with the speaker
- Recommended to run on Google Colab



Conclusion

- Natural Language Processing is a rapidly growing field with a wide range of applications and challenges
- With advances in machine learning and data science, generative NLP and transformers are becoming increasingly important for businesses and organizations to solve more complex problems
- We hope you learnt something new in NLP. Thank you for attending and open for questions!

References

- Attention Is All You Need; <https://arxiv.org/abs/1706.0376>
- <http://jalammar.github.io/illustrated-transformer/>
- <https://www.ruder.io/state-of-transfer-learning-in-nlp/>
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer; <https://arxiv.org/abs/1910.10683>
- Training language models to follow instructions with human feedback; <https://arxiv.org/abs/2203.02155>