Attention models

One of the most impactful of the family of Transformers is BERT

Bidirectional Encoder Representations from Transfomers

From Google, trained on 2 500M words

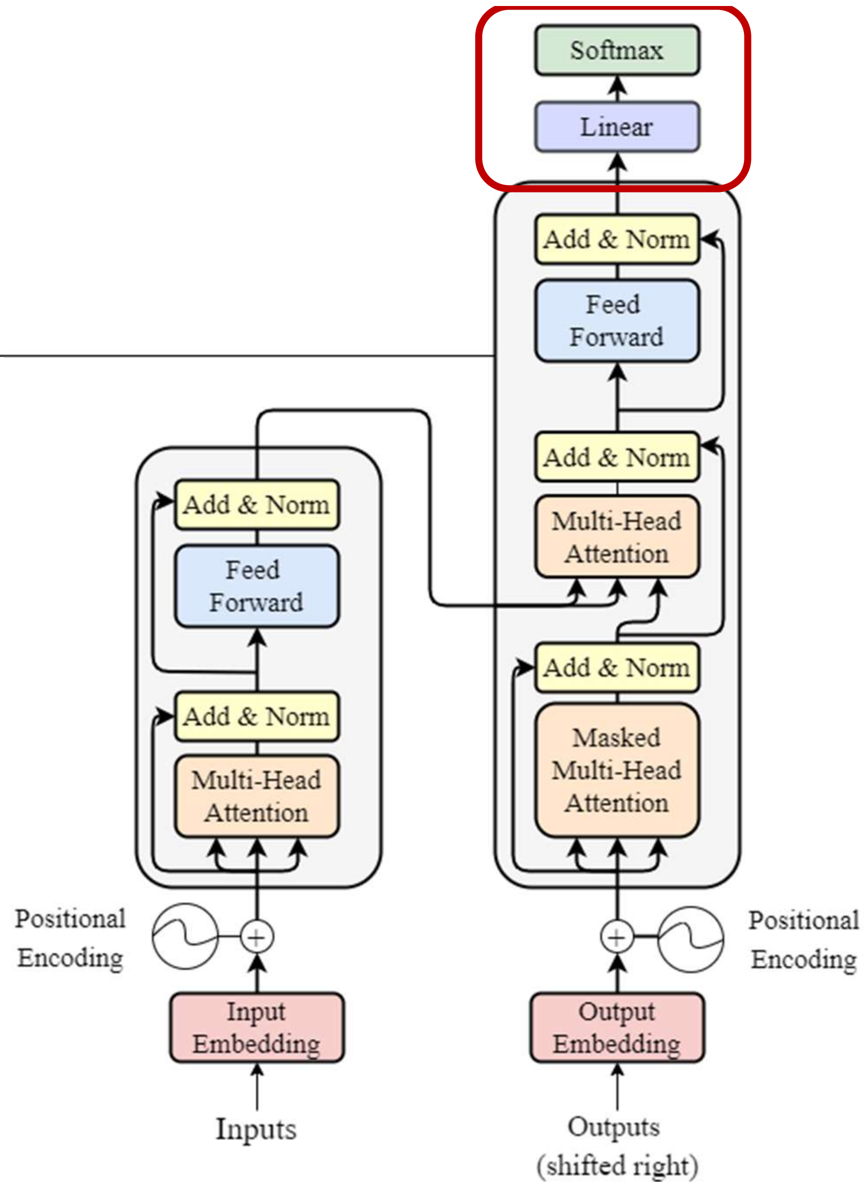
Attention models

Because what do we do with Transformers?

Well… we take these pretrained models and finetune them!

Attention models

So mostly this bit…

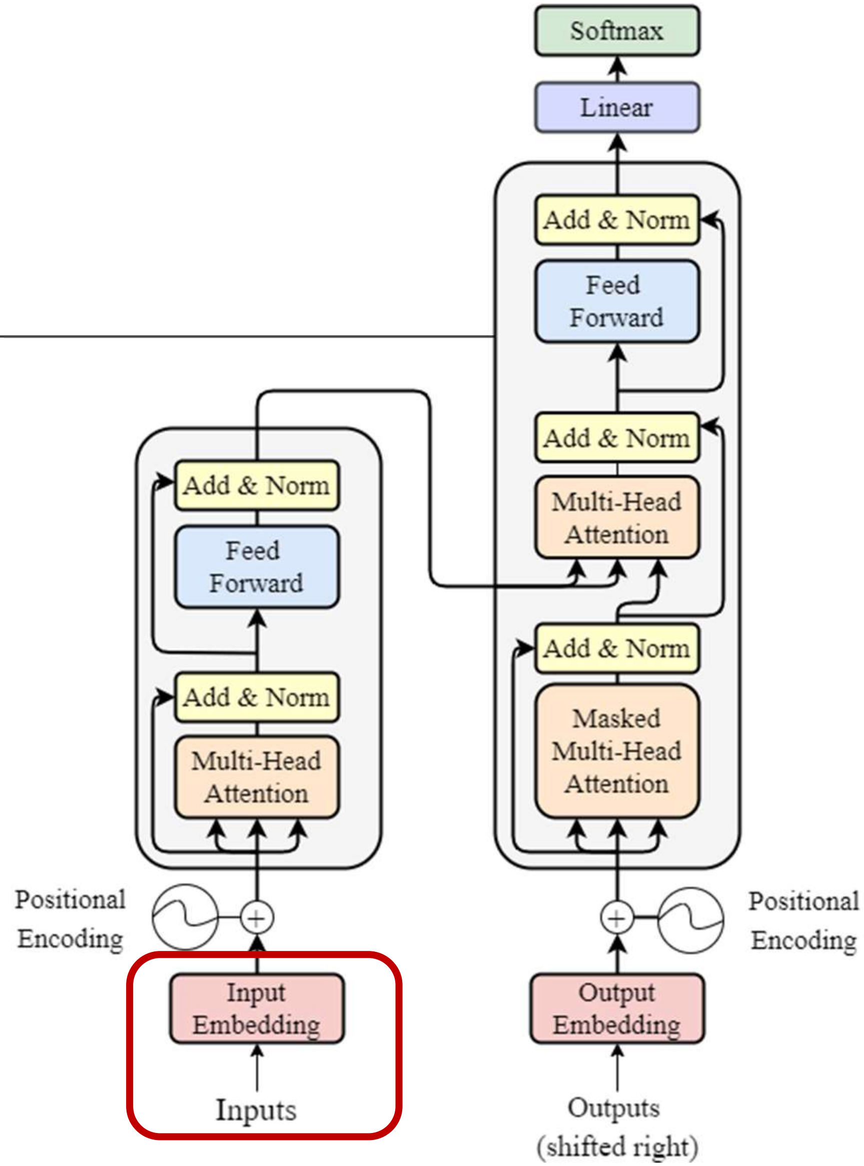


FUTURE CLUB

Attention models

So mostly this bit…

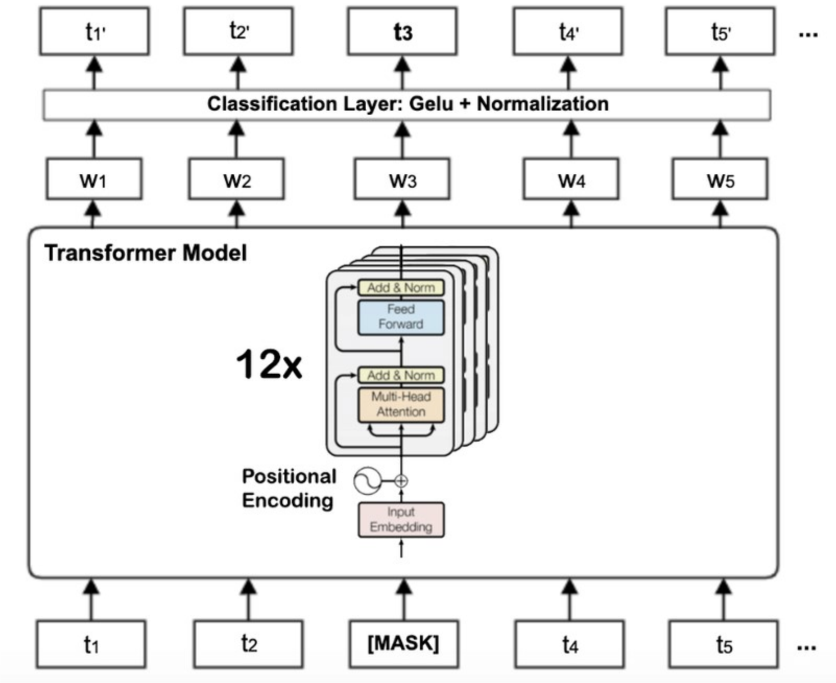
With its backprop influence, all the way to this bit



Attention models

Although BERT uses a different architecture because it does

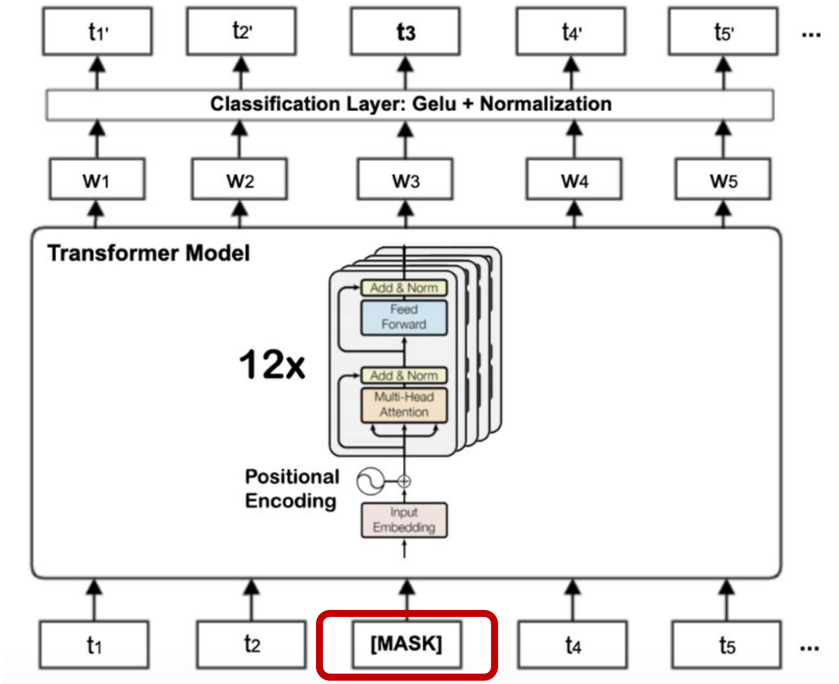
- Masked language modeling
- Next sentence prediction



Attention models

Although BERT uses a different architecture because it does

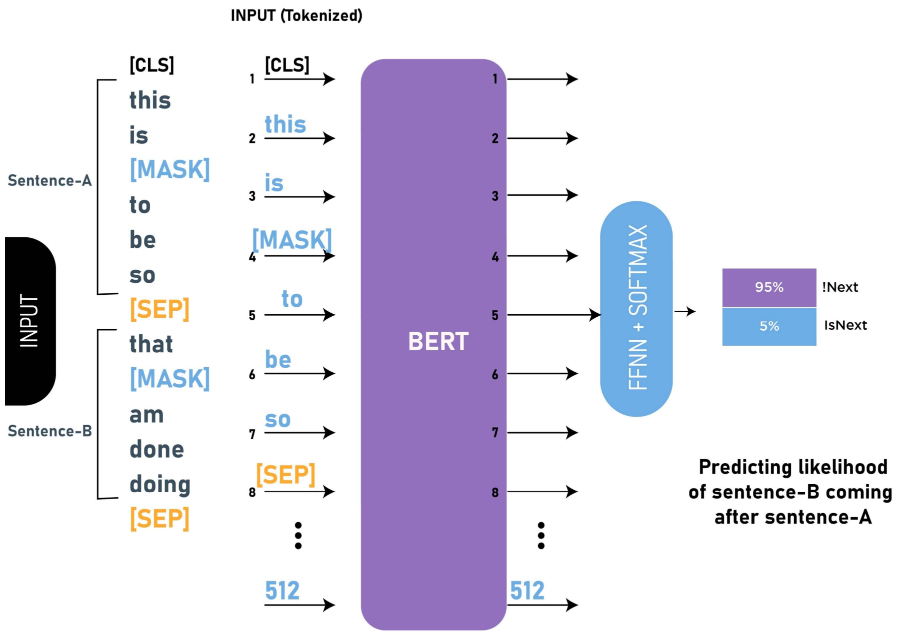
- Masked language modeling
- Next sentence prediction



Attention models

Although BERT uses a different architecture because it does

- Masked language modeling
- Next sentence prediction



Attention models

BERT is an encoder only, so what do Q, K and V mean here?

- MLM – Query is [MASK], Key and Value are all other tokens in the input ([MASK] too)
  - Attention is used to explain (unveil) [MASK] by the other words
- NSP – Works the same but Q is from first sentence and K and V are from **both** to capture relationships
  - Attention is used to explain dependencies between first sentence and next sentence, capturing information from both

Attention models

Wait – what is this [MASK]?

Attention models

What does BERT's input look like?

- A special token, [SEP], to mark the end of a sentence
- A special token, [CLS], to mark start of text. This token is used for classification tasks, but BERT expects it no matter what your application is.
- Tokens that conform with the fixed vocabulary used in BERT
- The Token IDs for the tokens, from BERT's tokenizer
- Mask IDs to indicate which elements in the sequence are tokens and which are padding elements
- Segment IDs used to distinguish different sentences
- Positional Embeddings used to show token position within the sequence

Attention models

Note that BERT's vocabulary does not just consist of words!

- Whole words
- Characters
- Subword information from the start of the word (superceded with some special token, mostly ##)
- Special subword information not at the start (preceded with some special token, mostly ##)

This is called WordPiece tokenization

Attention models

Note – getting a <u>word</u> vector from BERT is not trivial

Attention models

Note – getting a word vector from BERT is not trivial

Because a word will have different vectors based on its context

Attention models

Note – getting a word vector from BERT is not trivial

There are multiple approaches and each library chooses its own…

They all use some pooling mechanism though

Attention models

Note: a sentence vector is simply the (contextualized) word-vector for [CLS] !

Isn't that cool?

Attention models

So what can we use these Transformers for?

Attention models

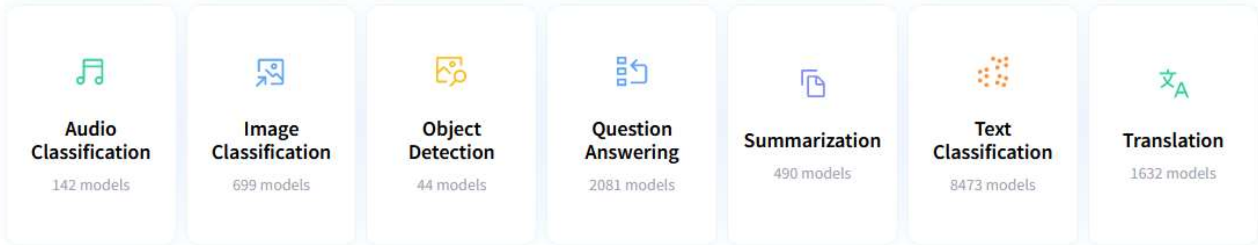
What not?!

Attention models

- Vectors models
- Text classifiers
- Sequence to sequence (machine translation)
- Other fields like image recognition
- Multi-modal learning (NLP + Image in one go)
- Text generation

Attention models

- Vectors models
- Text classifiers
- Sequence to sequence (machine translation)
- Other fields like image recognition
- Multi-modal learning (NLP + Image in one go)
- Text generation



Attention models

Notably, BERT (& co, there are many derivatives) is used for (contextualized) embeddings and hence classification

Attention models

Notably, BERT (& co, there are many derivatives) is used for (contextualized) embeddings and hence classification

BERT, RoBERTa, CamemBERT, DistilBERT, etc. etc.

Attention models

There is a competitor named XLM (cross lingual model)

Attention models

Then we have machine translation, mostly done using BART or NLLB

Attention models

Then there are a bunch focused on non-NLP tasks

Attention models

And then there is text generation!

Attention models

- GPT1
- GPT2
- GPT3
- WuDAO
- BLOOM
- ChatGPT / GPT 4

FUTURE CLUB

Attention models

- GPT1
  - This was fun, promising
- GPT2
- GPT3
- WuDAO
- BLOOM
- ChatGPT / GPT 4

Attention models

- GPT1
- GPT2
  - This made reasonably well-written documents when used well
- GPT3
- WuDAO
- BLOOM
- ChatGPT / GPT 4

Attention models

- GPT1
- GPT2
- GPT3
  - This stuff scared the world!
- WuDAO
- BLOOM
- ChatGPT / GPT 4

FUTURE CLUB

Attention models

- GPT1
- GPT2
- GPT3
  - This stuff scared the world!
- WuDAO
- BLOOM

THE VERGE TECH REVIEWS SCIENCE CREATORS ENTERTAINMENT VIDEO MORE

TECH \ ARTIFICIAL INTELLIGENCE

A college student used GPT-3 to write fake blog posts and ended up at the top of Hacker News

He says he wanted to prove the AI could pass as a human writer

By Kim Lyons | @SocialKimLy | Aug 16, 2020, 1:55pm EDT

Attention models

- GPT1
- GPT2
- GPT3
- WuDAO
  - Uses "GPT3-like" architecture to reason and "communicate" – and not just English!
- BLOOM
- ChatGPT / GPT 4

Attention models

- GPT1
- GPT2
- GPT3
- WuDAO
- BLOOM
  - 46 languages(!!), collaboration of many researchers
- ChatGPT / GPT 4

Attention models

- GPT1
- GPT2
- GPT3
- WuDAO
- BLOOM
  - 46 languages(!!), collaboration of many researchers
    And 13 programming languages ☺
- ChatGPT / GPT 4

FUTURE CLUB

Attention models

- ~~GPT1~~
- ~~GPT2~~
- ~~GPT3~~
- ~~WuDAO~~
- ~~BLOOM~~

ChatGPT

FUTURE CLUB

Attention models

- ~~GPT1~~
- ~~GPT2~~
- ~~GPT3~~
- ~~WuDAO~~
- ~~BLOOM~~

ChatGPT

(Well, LLMs like ChatGPT: LLaMa, Claude, Vicuna, etc.)

ChatGPT

So what is this thing?

ChatGPT

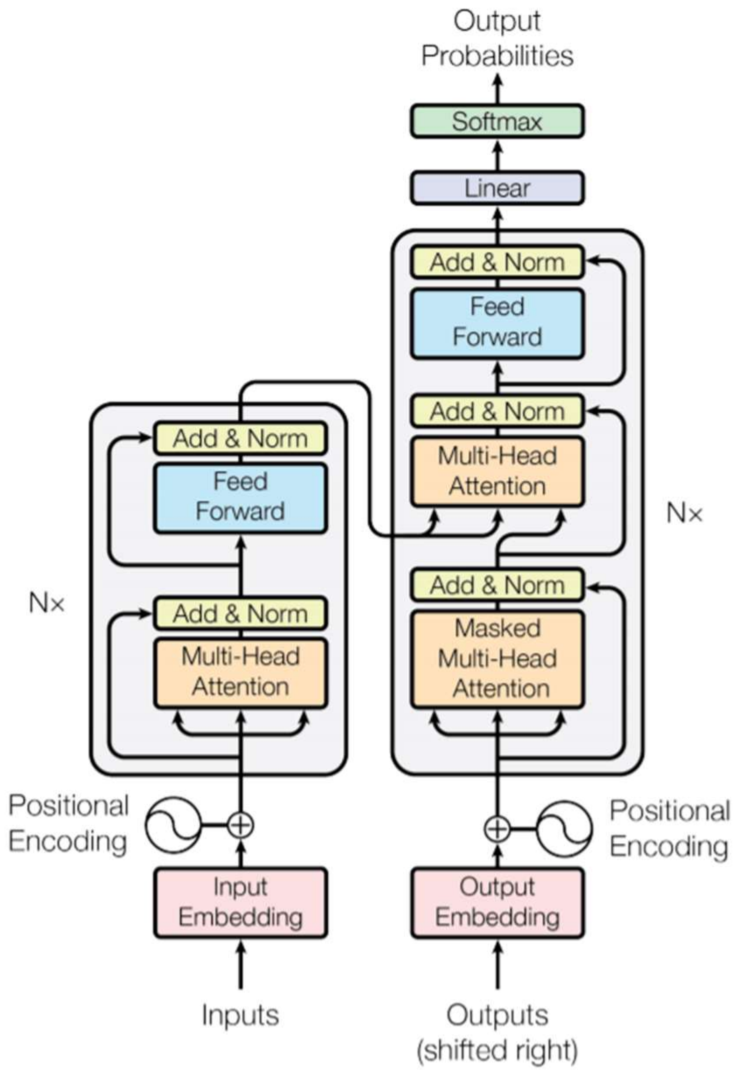


Figure 1: The Transformer - model architecture.

ChatGPT

This is the same base fundament as BERT

Encoder-only

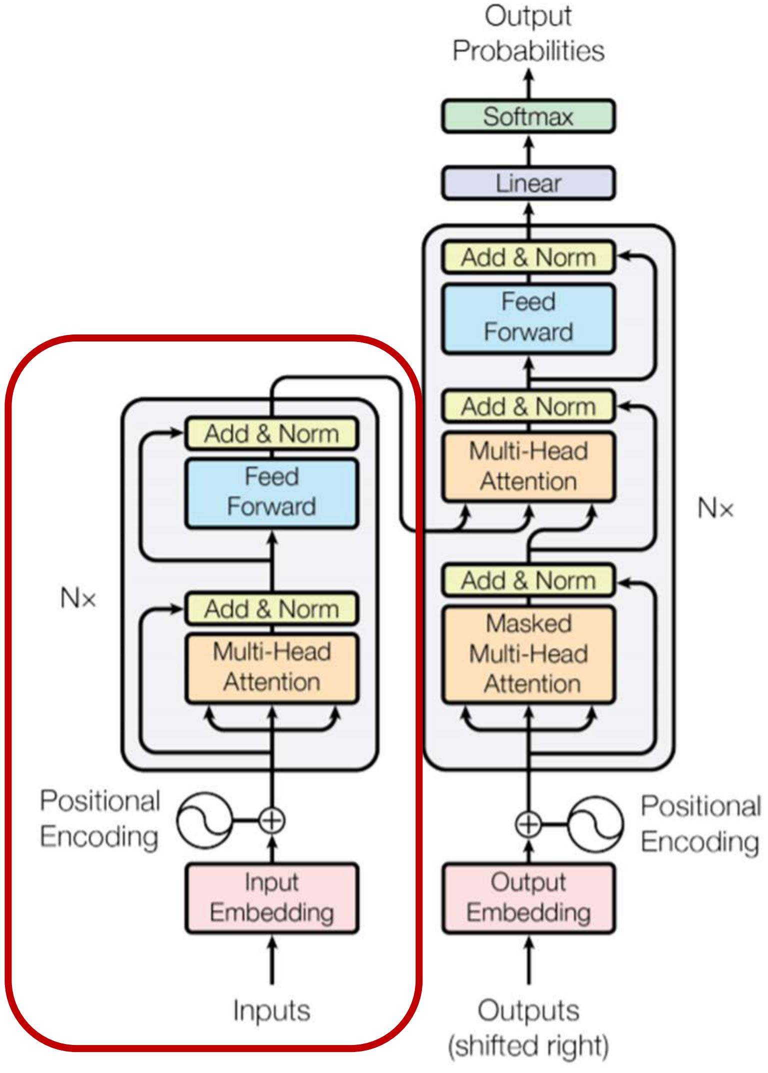


Figure 1: The Transformer - model architecture.

ChatGPT

And this is GPT

Decoder-only

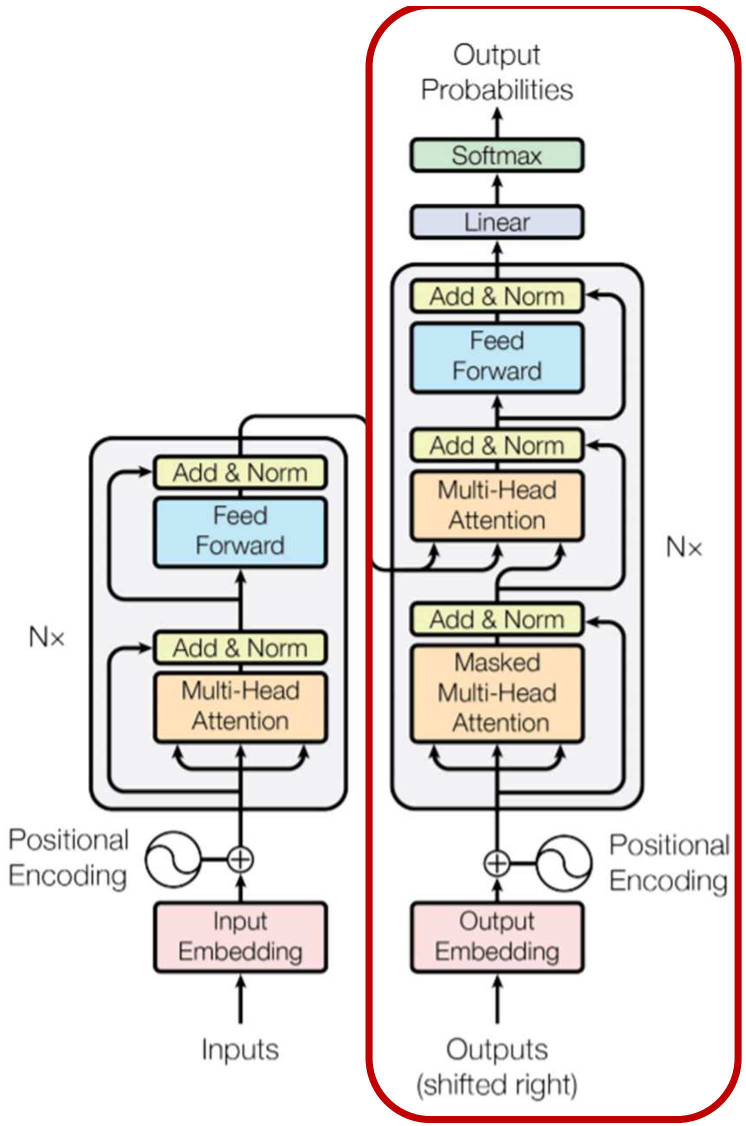


Figure 1: The Transformer - model architecture.

ChatGPT

Where BERT does masked-language-modeling

GPT does **autoregressive** language modeling: predict the next token/word

(based on input + already predicted words)

ChatGPT

Query – The token being generated

Key – Previously generated (or input) tokens

Value – The attention vectors associated with the Keys

So for the current token (Q) we pay attention to the previously generated tokens (K) and how much information (attention, V) they bear towards the current token

ChatGPT

Query – The token being generated

Key – Previously generated (or input) tokens

Value – The attention vectors associated with the Keys

So for the current token (Q) we pay attention to the previously generated tokens (K) and how much information (attention, V) they bear towards the current token

For training: compute self-attention, for generation: sample from probability distribution and select most probable one

FUTURE CLUB

ChatGPT

Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

Masked-language-modeling

The model is given a sequence of words with the goal of predicting a 'masked' word in the middle.

Example
Jacob [mask] reading

Jacob *fears* reading
Jacob *loves* reading
Jacob *enjoys* reading
Jacob *hates* reading

ChatGPT

So that is GPT, what is ChatGPT?

ChatGPT

InstructGPT with RLHF

ChatGPT

RLHF (Reinforcement Learning from Human Feedback)

1. Supervised fine-tuning from human inputs – basically let GPT generate responses and pick a good option (or suggest a new one)
2. Humans rank responses for quality and train model to favor higher quality responses

ChatGPT

InstructGPT

1. Build on the principles of GPT and RLHF but input queries are now instructions
2. Humans label data to generate answers that best “follow” the instructions

ChatGPT

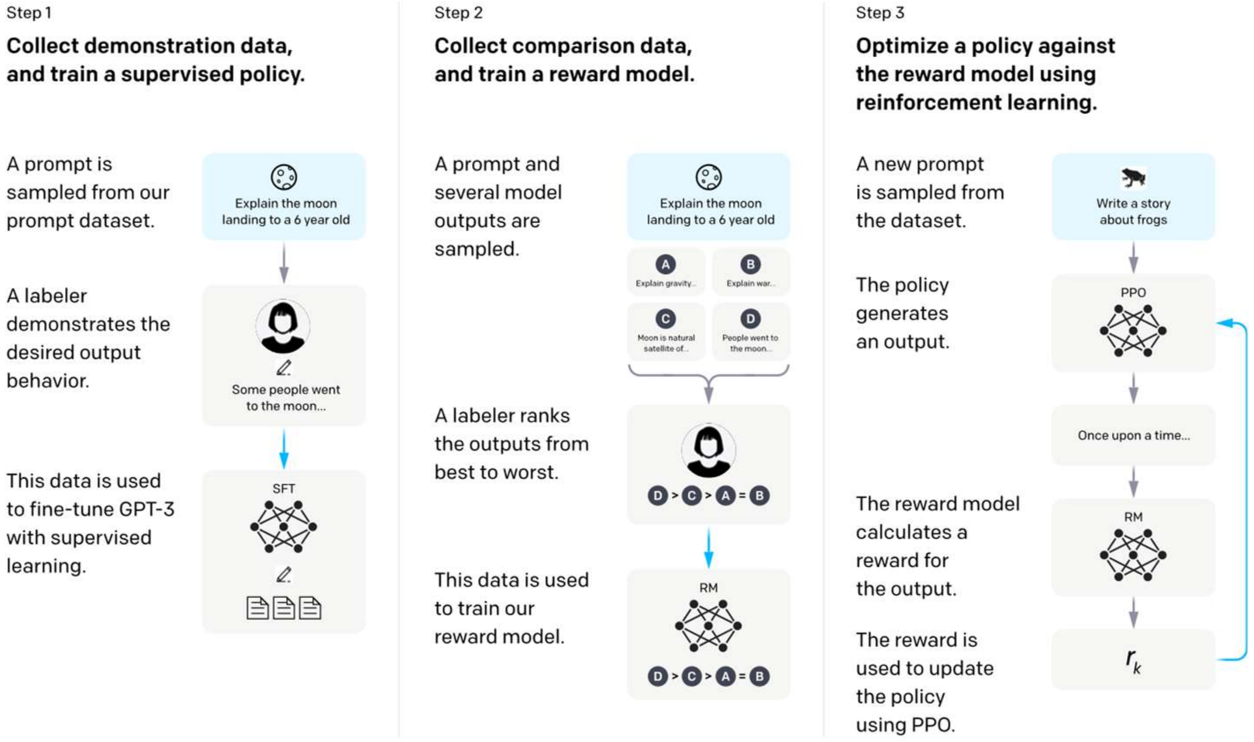
ChatGPT tries to disambiguate **questions** from **instructions**

ChatGPT

ChatGPT tries to disambiguate **questions** from **instructions**

And RLHF + InstructGPT make sure the answers are contextually relevant

ChatGPT


Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

ChatGPT

See also

https://openai.com/research/instruction-following

Attention models

So is ChatGPT (or LLMs) the answer to everything?

Attention models

What about text classification?

Attention models

What about text classification?

Well maybe with some prompt enineering

FUTURE CLUB

Attention models

What about text classification?

Well maybe with some prompt enineering

Speaking of which…

Attention models

Back to Transformers

So how do **<u>we</u>** use this in practice?

Attention models

One option

Go to chat.openai.com and voila ☺

Attention models

Or, go over to https://huggingface.co/ and start coding!

Attention models

Example for classification

Attention models

```python
from transformers import AutoTokenizer, AutoModelForSequenceClassification,
 TrainingArguments, Trainer, DataCollatorWithPadding
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")

def preprocess_function(examples):
    return tokenizer(examples["text"], truncation=True)

tokenized_imdb = imdb.map(preprocess_function, batched=True)
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
model = AutoModelForSequenceClassification.from_pretrained("distilbert-base
-uncased", num_labels=2)

training_args = TrainingArguments(
    output_dir="./results",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=5,
    weight_decay=0.01,
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_imdb["train"],
    eval_dataset=tokenized_imdb["test"],
    tokenizer=tokenizer,
    data_collator=data_collator,
)

trainer.train()
```

Attention models

Example for generation

Attention models

```python
from transformers import (GPT2LMHeadModel, GPT2Tokenizer)
MAX_LENGTH = int(10000)
args = obj({
    'model_type': 'gpt2',
    'model_name_or_path': 'gpt2',
    'length': 50,
    'temperature': 1.0,
    'repetition_penalty': 1.0,
    'k': 50,
    'p': 0.95,
    'seed': 1337,
    'num_return_sequences': 1,
    'device': None,
    'n_gpu': None
})
model_class = GPT2LMHeadModel
tokenizer_class = GPT2Tokenizer
tokenizer = tokenizer_class.from_pretrained(args.model_name_or_path)
model = model_class.from_pretrained(args.model_name_or_path)
model.to(args.device)
```

```python
args.length = adjust_length_to_model(args.length,
 max_sequence_length=model.config.max_position_embeddings)
encoded_prompt = tokenizer.encode(input_text, add_special_tokens=False,
 return_tensors="pt")
encoded_prompt = encoded_prompt.to(args.device)
if encoded_prompt.size()[-1] == 0:
    input_ids = None
else:
    input_ids = encoded_prompt
output_sequences = model.generate(
    input_ids=input_ids,
    max_length=args.length + len(encoded_prompt[0]),
    temperature=args.temperature,
    top_k=args.k,
    top_p=args.p,
    repetition_penalty=args.repetition_penalty,
    do_sample=True,
    num_return_sequences=args.num_return_sequences,
)
```

Attention models

Example for topic modelling

Attention models

There is KeyBERT and BERTopic

https://github.com/MaartenGr/KeyBERT

https://maartengr.github.io/BERTopic/index.html

Attention models

But, really, just check the docs, they are quite good.

https://huggingface.co/docs/transformers/index

Program

- ~~Basics of Deep Learning (for NLP)~~
- ~~Vectorization models~~
- ~~Auto-encoders~~
- ~~Recurrent neural nets~~
- ~~Recursive neural nets~~
- ~~LSTMs~~
- ~~Attention models~~
- Deep learning for NLP in practice
- t-SNE
- Google Colab

DL NLP in practice

So this is all cool stuff!

Let's go ahead and train some models ourselves ☺

Or not?

DL NLP in practice

There are many challenges with DL NLP in practice!

We’ll check out a couple and how to remedy them.

DL NLP in practice

- Cost
- Language support
- Knowledge on usage
- Explainability (ethical AI)

DL NLP in practice

Cost

Speeds, Sizes, Times

Training logs: Tensorboard link

- Dates:
  - Started 11th March, 2022 11:42am PST
  - Estimated end: 5th July, 2022
- Checkpoint size:
  - Bf16 weights: 329GB
  - Full checkpoint with optimizer states: 2.3TB
- Training throughput: About 150 TFLOP per GPU per second
- Number of epochs: 1
- Estimated cost of training: Equivalent of $2-5M in cloud computing (including preliminary experiments)
- Server training location: Île-de-France, France

Environmental Impact

The training supercomputer, Jean Zay (website), uses mostly nuclear energy. The heat generated by it is reused for heating campus housing.

Estimated carbon emissions: *(Forthcoming.)*

Estimated electricity usage: *(Forthcoming.)*

FUTURE CLUB

DL NLP in practice

Cost

DL NLP in practice

Cost

OpenAI recently published GPT-3, the largest language model ever trained. GPT-3 has 175 billion parameters and would require 355 years and $4,600,000 to train - even with the lowest priced GPU cloud on the market.[1]

Source https://lambdalabs.com/blog/demystifying-gpt-3/

DL NLP in practice

Cost

OpenAI recently published GPT-3, the largest language model ever trained. GPT-3 has 175 billion parameters and would require 355 years and $4,600,000 to train - even with the lowest priced GPU cloud on the market.[1]

Source https://lambdalabs.com/blog/demystifying-gpt-3/

DL NLP in practice

GPU RAM required (for inference!)

- LLaMA 65B **260GB**
- Falcon 40B **160GB**
- LLaMA 33B **132GB**
- Falcon 7B **28GB**
- LLaMa 7B **14GB** (with some tricks)

DL NLP in practice

But the bigger the context window, the more vRAM you need.

DL NLP in practice

But the bigger the context window, the more vRAM you need.

Claude 2.1 has a 200k window!

DL NLP in practice

But the bigger the context window, the more vRAM you need.

Claude 2.1 has a 200k window!

DL NLP in practice

Note that all these LLMs (Large Language Models) are highly subsidized and sponsored by governments, industry and NVIDIA

DL NLP in practice

Solution?

DL NLP in practice

Solution?

Don’t train these yourelf. Use pre-trained models off the bat or finetune only!

DL NLP in practice

- ~~Cost~~
- Language support
- Knowledge on usage
- Explainability (ethical AI)

DL NLP in practice

So now you want to use this stuff for Dutch! Great, go ahead.

DL NLP in practice

So now you want to use this stuff for Dutch! Great, go ahead.

But wait a minute…

DL NLP in practice

No Dutch in BLOOM ☹

No Dutch for GPT3 ☹

No Dutch for GPT2 ☹

No Dutch in LLaMa2 ☹

DL NLP in practice

Or even worse: real-life use-case – Kirundi, main language in Burundi.

There isn't even a wikipedia in Kirundi

DL NLP in practice

Why is this?

FUTURE CLUB

DL NLP in practice

These are the most widely used datasets to train (unsupervised) on

Dataset	# Tokens (Billions)
Total	499
Common Crawl (filtered by quality)	410
WebText2	19
Books1	12
Books2	55
Wikipedia	3

DL NLP in practice

These are the most widely used datasets to train (unsupervised) on

This is skewed to world languages!

English

Spanish

Chinese

Arabic

Dataset	# Tokens (Billions)
Total	499
Common Crawl (filtered by quality)	410
WebText2	19
Books1	12
Books2	55
Wikipedia	3

DL NLP in practice

Reality check

Dutch is not a world language ☹

DL NLP in practice

Solution?

- ❑ Don’t model small languages?
- ❑ Join a tech giant?
- ❑ Pray someone trains your language?
- ❑ Use transfer learning
  - ❑ Take pre-trained mixed language models and apply them to your language
  - ❑ Use machine translation (there’s a model for that) to your language

DL NLP in practice

- ~~Cost~~
- ~~Language support~~
- Knowledge on usage
- Explainability (ethical AI)

DL NLP in practice

Knowledge on usage – que es eso?

FUTURE CLUB

DL NLP in practice

Question

Do you have a GPU?

Which one?

Does it run CUDA?

Tensorflow, cuDNN, torch, cuBLAS?

DL NLP in practice

Will you install all of this?

Maintain it?

DL NLP in practice

What about all these parameters?

Epochs, mini-batch size, sliding window, vectorization

DL NLP in practice

What layers would you use when?

When to use dropout, with which thresholds?

DL NLP in practice

Bottom line: deep learning is pretty complex

And this is not just about understanding what a neural network does!

DL NLP in practice

Fun anecdote

Geoff Hinton and his crew started winning a lot of Kaggle competitions

Without being specialized in niches like NLP, Image, Sensor data

DL NLP in practice

Fun anecdote

That means that

1. Deep learning can help tackle problems without deep domain knowledge

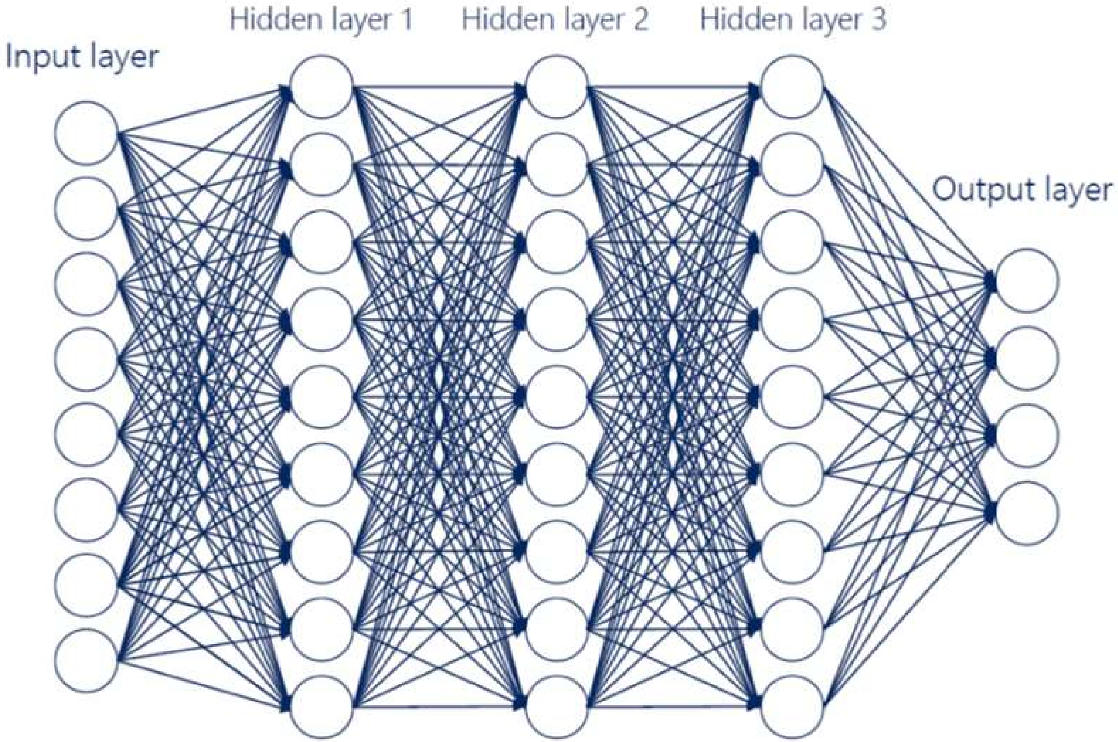
2. Doing it, is somewhat of a black art (tho we are democratizing it)

DL NLP in practice

- ❑ ~~Cost~~
- ❑ ~~Language support~~
- ❑ ~~Knowledge on usage~~
- ❑ Explainability (ethical AI)

DL NLP in practice

Question – here's a neural net

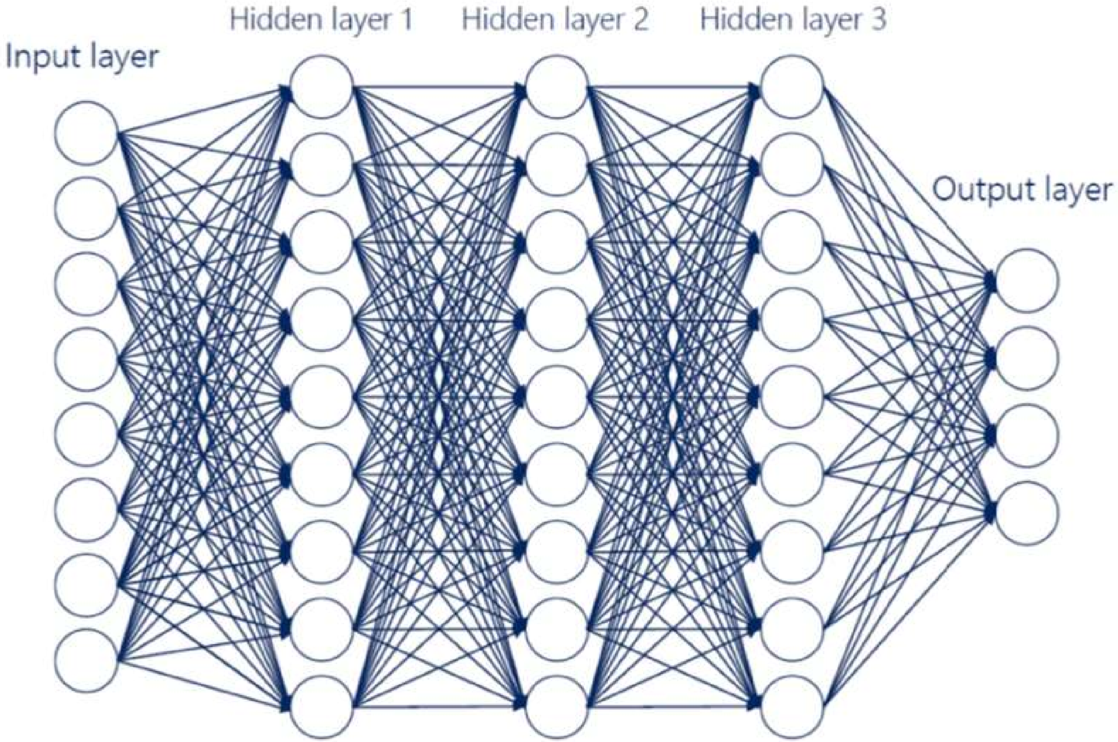


DL NLP in practice

Question – here's a neural net

Why do I get [0.2, 0.8, -0.2, 1.0]

out for my input which is a sentence?



DL NLP in practice

Or better said: how do we explain what a neural net does?

DL NLP in practice

Or better said: how do we explain what a neural net does?

And more importantly: the bias it contains

DL NLP in practice

True story

Amazon built a model that predicted if a candidate was a good match to become a software engineer within Amazon

DL NLP in practice

True story

But because the model was trained on existing software engineers, being predominantly male, the model just learned *"if you are male, you are hired"*

DL NLP in practice

Result

Big scandal


REUTERS World Business Markets Breakingviews Video More

RETAIL OCTOBER 11, 2018 / 1:04 AM / UPDATED 4 YEARS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

DL NLP in practice

In this case even, the outcome was tangible – we can see it hired men only

But what if we have complex, autonomous, automated outcomes that are not curated?

DL NLP in practice

GPT3, BLOOM, etc. can create human-like texts

DALL-E, MidJourney, Stable Duffision can create human-like art, images

Deepfake methods can create human-like videos

DL NLP in practice

GPT3, BLOOM, etc. can create human-like texts

DALL-E, MidJourney, Stable Duffision can create human-like art, images

Deepfake methods can create human-like videos

DL NLP in practice

Ethical AI tries to advocate against all these practices

But developments outpace what we can monitor and manage…

DL NLP in practice

- ~~Cost~~
- ~~Language support~~
- ~~Knowledge on usage~~
- ~~Explainability (ethical AI)~~

Program

- ~~Basics of Deep Learning (for NLP)~~
- ~~Vectorization models~~
- ~~Auto-encoders~~
- ~~Recurrent neural nets~~
- ~~Recursive neural nets~~
- ~~LSTMs~~
- ~~Attention models~~
- ~~Deep learning for NLP in practice~~
- t-SNE
- Google Colab

t-SNE

You: Hey Erik, I got a really cool vectorization model that captures semantics, syntax, irony ánd sarcasm in one go!

t-SNE

Me: prove it

t-SNE

Meet: t-SNE

(t-Distributed Stochastic Neighbor Embedding)

A dimension reduction method developed at TU Delft

t-SNE

What does it do?

Compress your vectors into less dimensions – normally 2

Capturing as much information as possible in those 2 dimensions

t-SNE

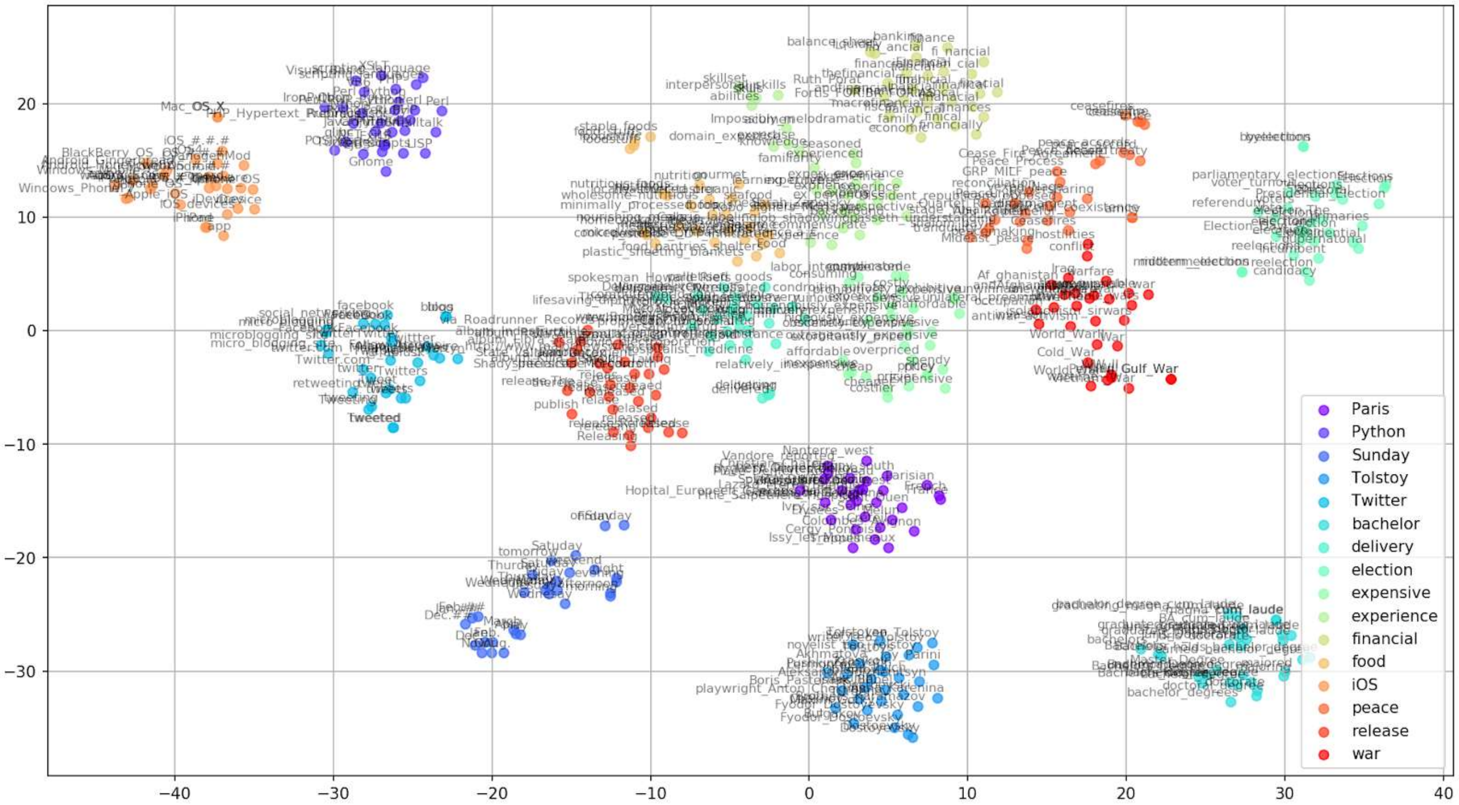
What does it do?

Compress your vectors into less dimensions – normally 2

Capturing as much information as possible in those 2 dimensions

Which means we can plot it (our word vectors!)

t-SNE


Paris
Python
Sunday
Tolstoy
Twitter
bachelor
delivery
election
expensive
experience
financial
food
iOS
peace
release
war
−40
−30
−20
−10
0
10
20
30
40
20
10
0
−10
−20
−30

t-SNE

```
from gensim.models import word2vec
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

model = word2vec.Word2Vec(corpus, size=100, window=20, min_count=200, workers=4)

labels = []
tokens = []
for word in model.wv.vocab:
  tokens.append(model[word])
  labels.append(word)

  tsne_model = TSNE(perplexity=40, n_components=2, init='pca', n_iter=2500, random_state=23)
  new_values = tsne_model.fit_transform(tokens)

x = []
y = []
for value in new_values:
    x.append(value[0])
    y.append(value[1])

plt.figure(figsize=(16, 16))
for i in range(len(x)):
    plt.scatter(x[i],y[i])
    plt.annotate(labels[i], xy=(x[i], y[i]), xytext=(5, 2), textcoords='offset points', ha='right', va='bottom')

plt.show()
```

https://www.kaggle.com/code/jeffd23/visualizing-word-vectors-with-t-sne/notebook

Program

- ~~Basics of Deep Learning (for NLP)~~
- ~~Vectorization models~~
- ~~Auto-encoders~~
- ~~Recurrent neural nets~~
- ~~Recursive neural nets~~
- ~~LSTMs~~
- ~~Attention models~~
- ~~Deep learning for NLP in practice~~
- ~~t-SNE~~
- Google Colab

Google Colab

Remember out points about DL being hard?

Google Colab

Remember out points about DL being hard?

GPU, CUDA, Tensorflow, Huggingface, compiling libraries…

Google Colab

Well, meet Google Colab!

It does all the hard for you – and you don't need to buy an expensive GPU

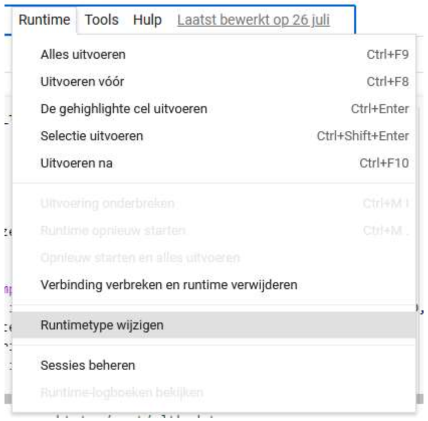
Google Colab

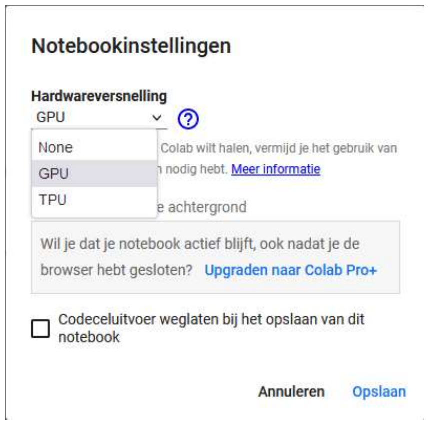
Create a Google account with your JADS e-mail address

Go to https://colab.research.google.com/

Google Colab

Create a notebook, make sure you use a GPU runtime when working with DL libraries such as Keras, Tensorflow, Huggingface





Google Colab

You will get a 12GB capable GPU that you can use for 12 hours in one go

Google Colab

You will get a 12GB capable GPU that you can use for 12 hours in one go

Tip: checkpoint your epochs in between to continue across 12h sessions!

Google Colab

You can also connect Google Drive to Colab

Google Colab

You can also connect Google Drive to Colab

Including the datasets

Google Colab

```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
df = pd.read_json("/content/drive/MyDrive/dataset.json", encoding='utf8', lines=True)
df.head
```

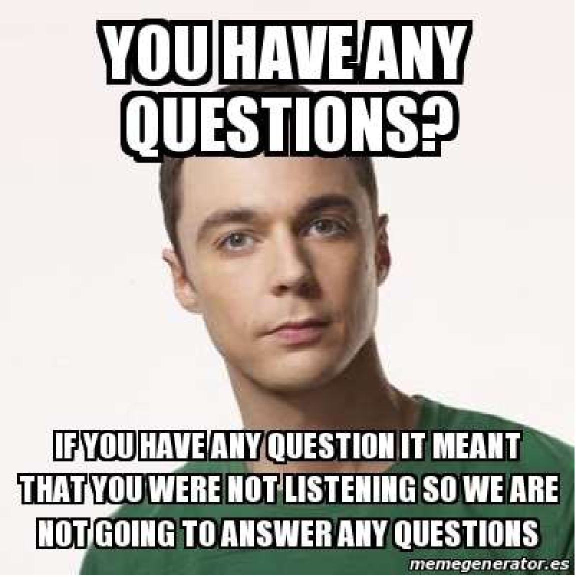
FUTURE CLUB

Program

- ~~Basics of Deep Learning (for NLP)~~
- ~~Vectorization models~~
- ~~Auto-encoders~~
- ~~Recurrent neural nets~~
- ~~Recursive neural nets~~
- ~~LSTMs~~
- ~~Attention models~~
- ~~Deep learning for NLP in practice~~
- ~~t-SNE~~
- ~~Google Colab~~

Questions?

Thanks for this DL ride



erik@futureclub.nl

https://www.futureclub.nl/

https://github.com/AI-Commandos/LLaMa2lang

https://github.com/AI-Commandos/RAGMeUp

https://www.linkedin.com/in/eriktromp/

https://www.udemy.com/course/the-definitive-intro-to-big-data-science/?referralCode=47E37006CFC5B5599874

https://www.udemy.com/course/every-data-architecture-is-the-same/?referralCode=CD1F8BF2562089D63617