Language Models in the Wild

•••

Deep Learning Sessions Lisboa

Telmo Pessoa Pires

Disclaimer

Opinions here are my own, not my employer’s.

Any mistakes are mine.

Overview

- What is a Language Model?
- Classical Language Modeling
- Deep Learning Language Modeling
- In practice: Multilingual Language Modeling

About me

- Machine Learning Researcher/Engineer @ Apple
- Before:
  - AI Research Resident @ Google
  - AI Researcher @ Unbabel
  - Researcher @ IT
  - Student @ IST

What is a language model?

A probability distribution over sentences:

$p(w_1, \dots, w_n)$ meaning $\sum_{(w_1,\dots,w_n)\in V*} p(w_1, \dots, w_n) = 1$

Tokens: words, punctuation, ...

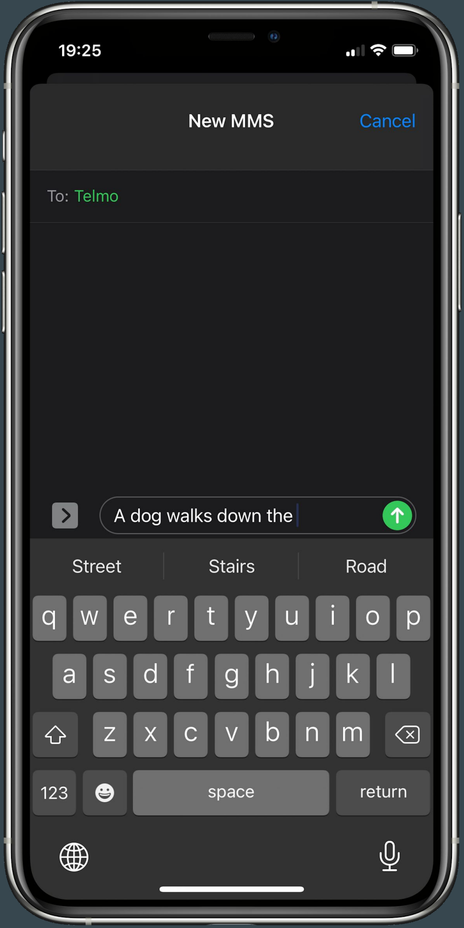
All sentences (including random)

Hopefully, likely ≈ plausible

$$p(This, is, a, talk) > p(talk, is, a, This)$$

Why do we care?

Most obvious example: autocorrect on smartphones.



To: Telmo

A dog walks down the

Street | Stairs | Road

Why?

Useful when generating text: translation, speech recognition, …

Transfer Learning.

Classical Language Modeling

Classical Language Models

The basic idea is to count how frequent a sentence is in a corpus.

$$p(w_1, \ldots, w_n) = \frac{count(w1, \ldots, w_n)}{N}$$

Number of sentences

Problem: Unseen sentences have zero probability.

N-grams

Applying the chain rule of probabilities:

$$p(w_1, \ldots, w_n) = \prod_{i=1}^{n} p(w_i | \underbrace{w_1, \ldots, w_{i-1}}_{\text{Unbounded context}})$$

$$= p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)...$$

Simplifying assumption: the ith word depends only on a few words before it.

N-grams

Unigram: words don’t depend on context.

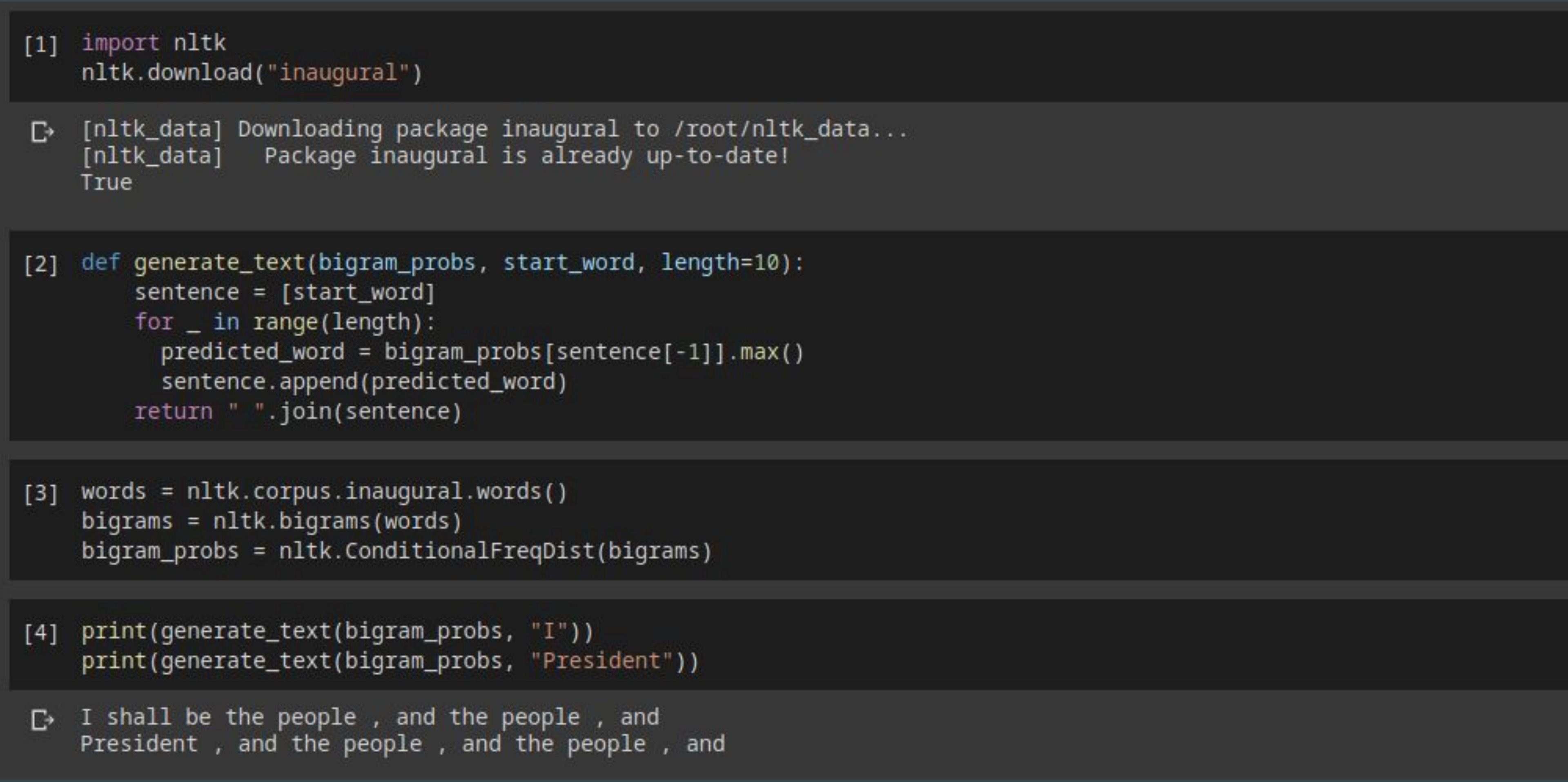
$$p(w_i|w_1,\ldots,w_{i-1}) = p(w_i) = \frac{count(w_i)}{\sum_j count(w_j)}$$

Total count of all words

Bigram: words depend only on the previous word.

$$p(w_i|w_1,\ldots,w_{i-1}) = p(w_i|w_{i-1}) = \frac{count(w_{i-1},w_i)}{count(w_{i-1})}$$

And so on...



```
[1] import nltk
    nltk.download("inaugural")
```

```
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data]   Package inaugural is already up-to-date!
True
```

```
[2] def generate_text(bigram_probs, start_word, length=10):
        sentence = [start_word]
        for _ in range(length):
          predicted_word = bigram_probs[sentence[-1]].max()
          sentence.append(predicted_word)
        return " ".join(sentence)
```

```
[3] words = nltk.corpus.inaugural.words()
    bigrams = nltk.bigrams(words)
    bigram_probs = nltk.ConditionalFreqDist(bigrams)
```

```
[4] print(generate_text(bigram_probs, "I"))
    print(generate_text(bigram_probs, "President"))
```

```
I shall be the people , and the people , and
President , and the people , and the people , and
```

Use case: Phrase-based Machine Translation

Goal: find the most likely translation: $\underset{e_1,\ldots,e_n}{\operatorname{argmax}} \quad p(\underbrace{e_1,\ldots,e_n}_{\text{English sentence}}|\underbrace{f_1,\ldots,f_m}_{\text{Portuguese sentence}})$

Apply Bayes' rule:

English language model

$$p(e_1,\ldots,e_n|f_1,\ldots,f_m) = \frac{p(f_1,\ldots,f_m|e_1,\ldots,e_n)\cdot p(e_1,\ldots,e_n)}{p(f_1,\ldots,f_m)}$$

constant

Use case: Phrase-based Machine Translation

How do you say “De nada” in English?

1. “Of nothing”?

$$\left.\begin{array}{l} p(de|of) = 0.99 \\ p(nada|nothing) = 0.99 \end{array}\right\} p(de, nada|of, nothing) = 0.98$$

$$p(of, nothing) = 0.00001$$

Then $p(of, nothing|de, nada) \propto 0.98 \cdot 0.00001 = 0.0000098$

Use case: Phrase-based Machine Translation

2. “You are welcome”?

$$p(de, nada|you, are, welcome) = 0.90$$
$$p(you, are, welcome) = 0.01$$

Then $p(you, are, welcome|de, nada) \propto 0.90 \cdot 0.01 = 0.009 > 0.0000098$

The language model steered us to a better translation!

Automatic Speech Recognition

We can apply the same trick to distinguish between similar sounding sentences:

Recognize speech

Wreck a nice beach

Much more likely

Some problems

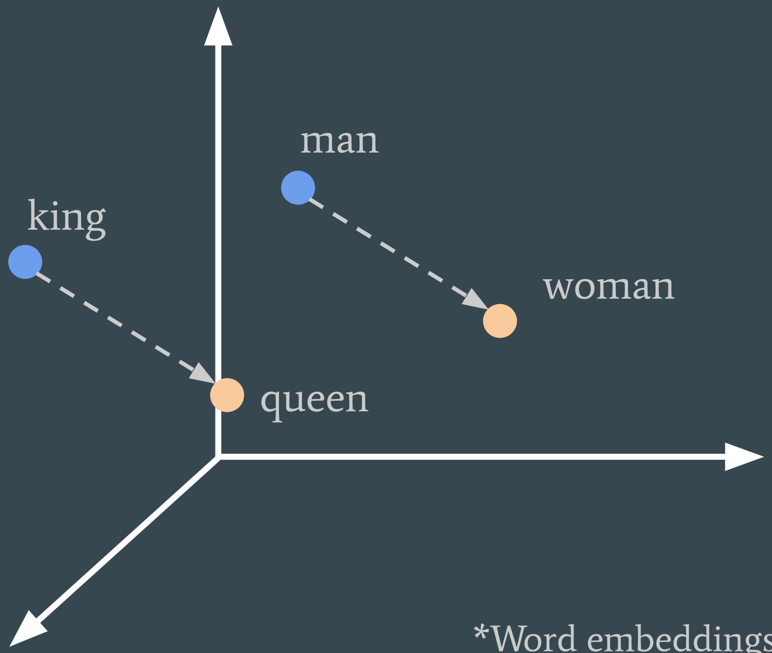
Hard to train with big context, due to data sparsity: smoothing and interpolation!

No semantics: can’t transfer knowledge between similar words.

Deep Learning Language Modeling

An aside: Word Embeddings

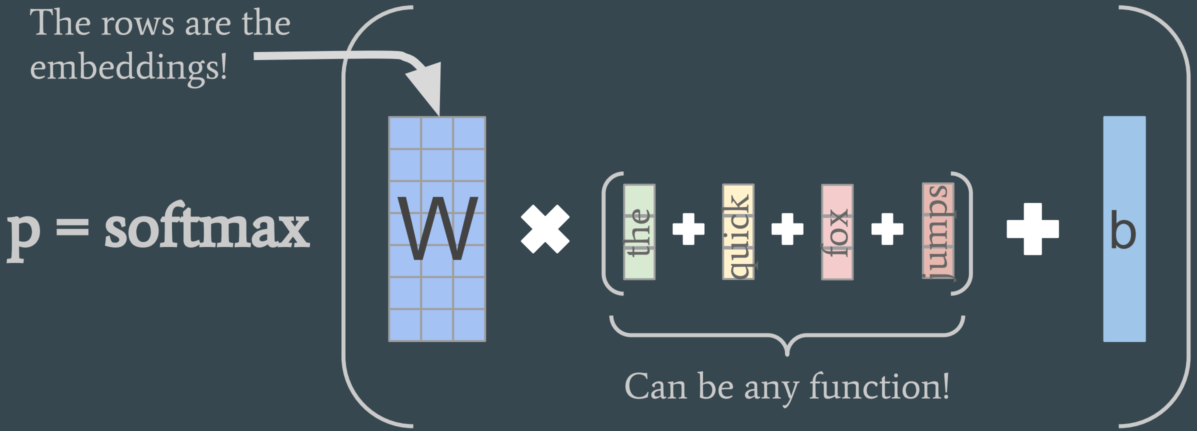
Words as vectors. Similar words, “close” vectors.



*Word embeddings were already used before DL!

Learning Embeddings

Predict a missing word given the context: "<u>the quick</u> brown <u>fox jumps</u>".



The rows are the embeddings!

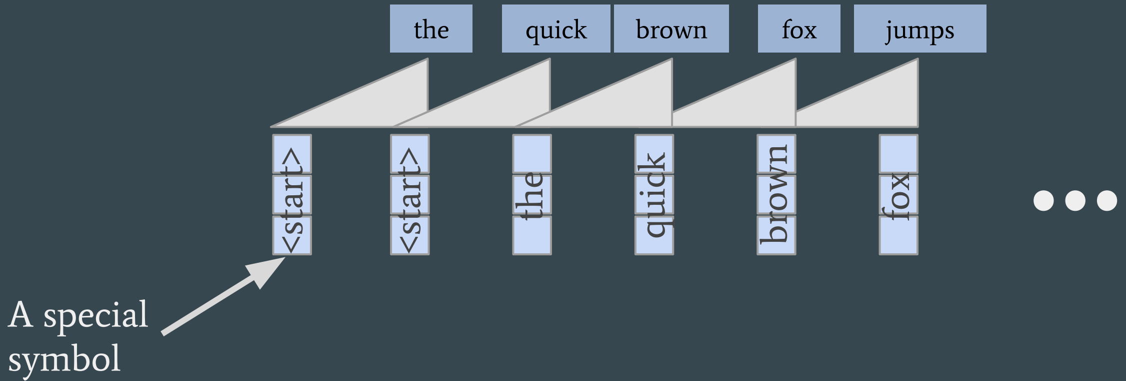
Can be any function!

Maximize the probability of the missing word.

Back to Language Models: Convolutional Neural Networks

We want to model: $p(w_i|w_1, \ldots, w_{i-1})$

How to condense a variable context into a fixed size vector? Clip it!



Recurrent Neural Networks

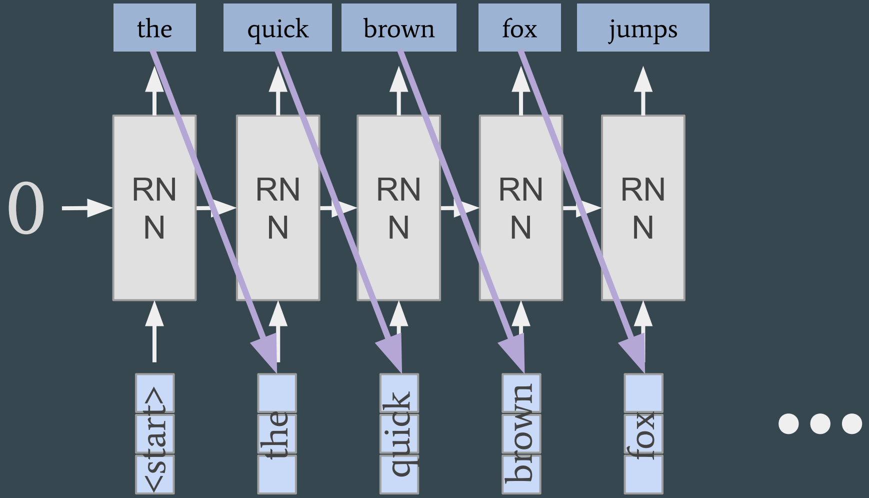
How to get unlimited context: recursivity.

$$p(w_i|w_1, \ldots, w_{i-1}) = f(w_i, f(w_{i-1}, \ldots))$$

Current word

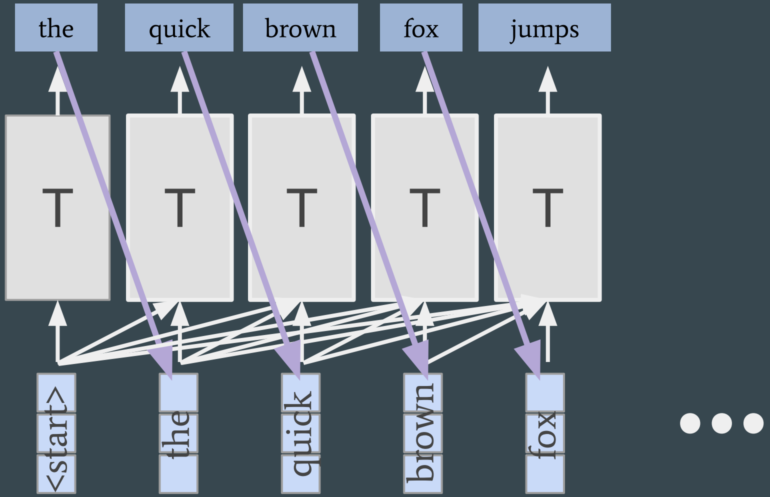
Function of the previous timestep

RNNs step by step



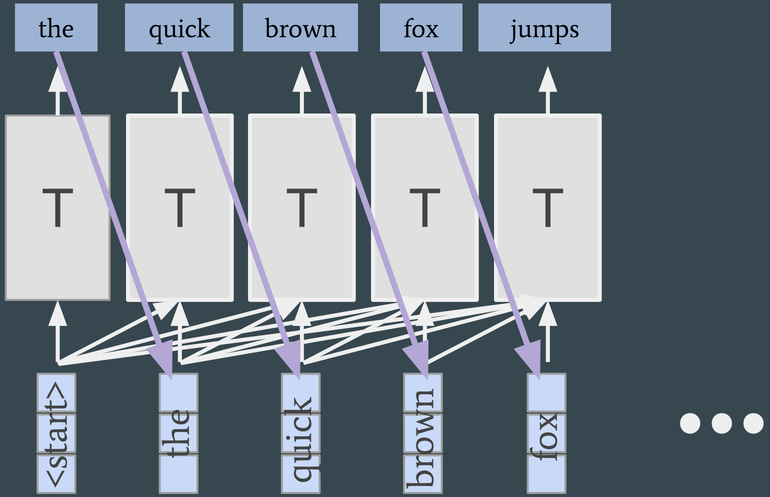
Transformers

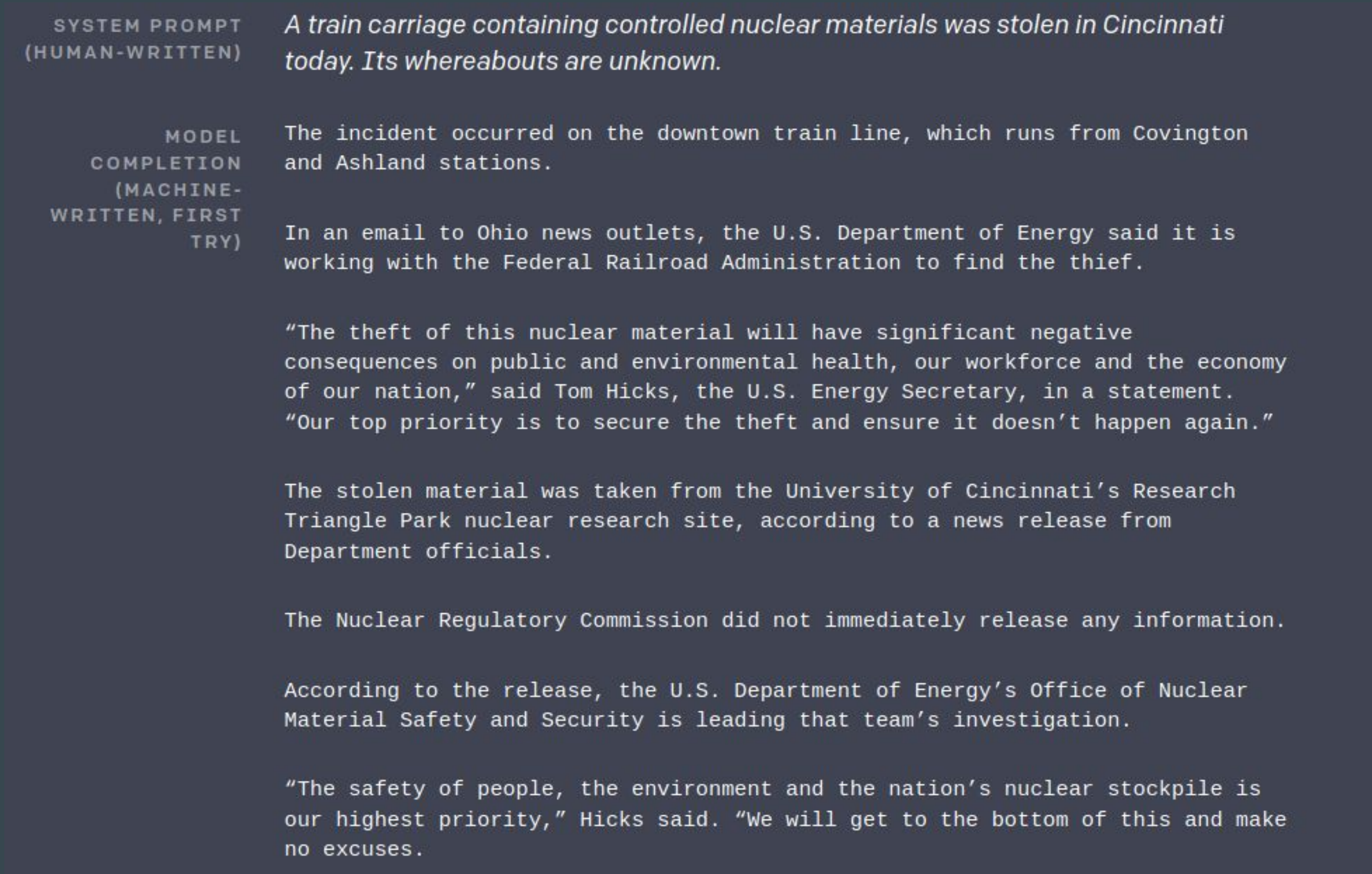
Directly attend at all previous words at each step.



GPT

A big transformer trained on LOTS of data.





SYSTEM PROMPT (HUMAN-WRITTEN)

A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)

The incident occurred on the downtown train line, which runs from Covington and Ashland stations.

In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.

“The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.”

The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials.

The Nuclear Regulatory Commission did not immediately release any information.

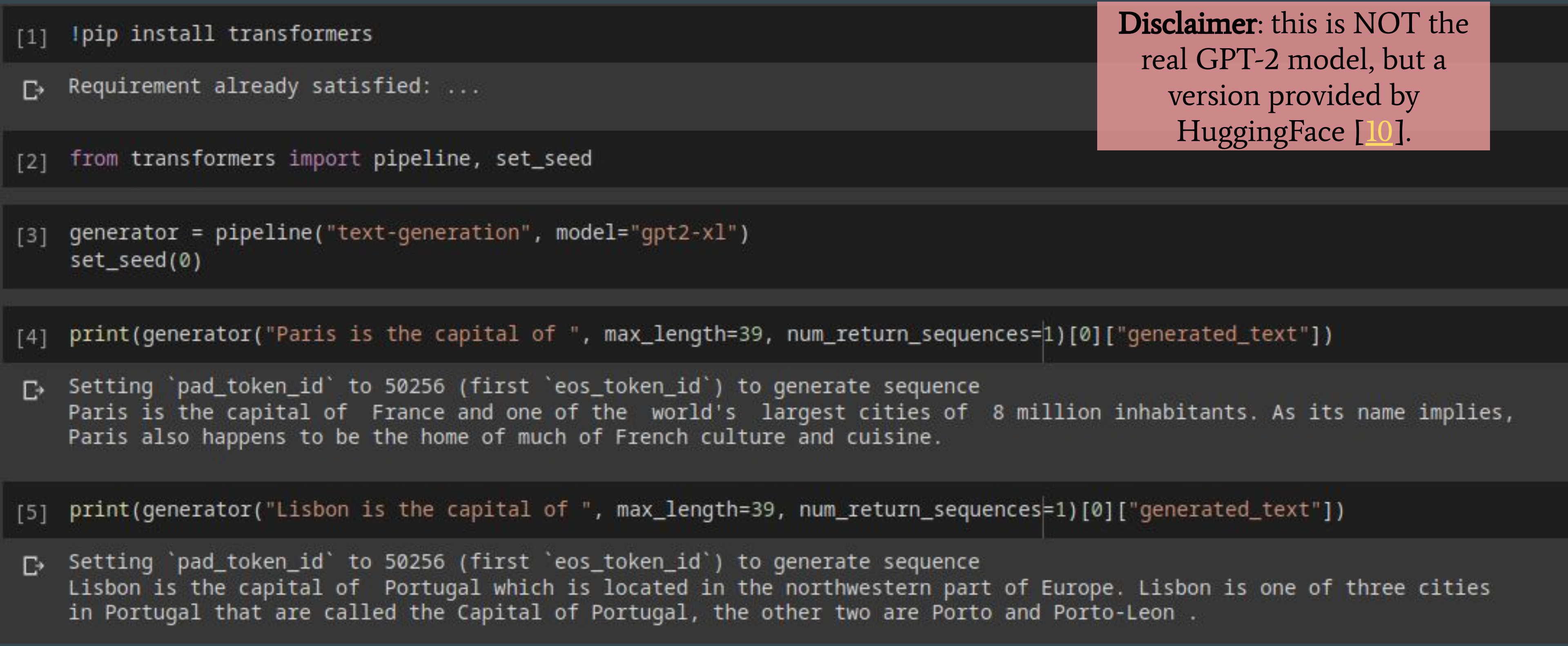
According to the release, the U.S. Department of Energy’s Office of Nuclear Material Safety and Security is leading that team’s investigation.

“The safety of people, the environment and the nation’s nuclear stockpile is our highest priority,” Hicks said. “We will get to the bottom of this and make no excuses.

Taken from [4].

GPT learns common knowledge

Disclaimer: this is NOT the real GPT-2 model, but a version provided by HuggingFace [10].



```
[1] !pip install transformers
```

```
Requirement already satisfied: ...
```

```
[2] from transformers import pipeline, set_seed
```

```
[3] generator = pipeline("text-generation", model="gpt2-xl")
    set_seed(0)
```

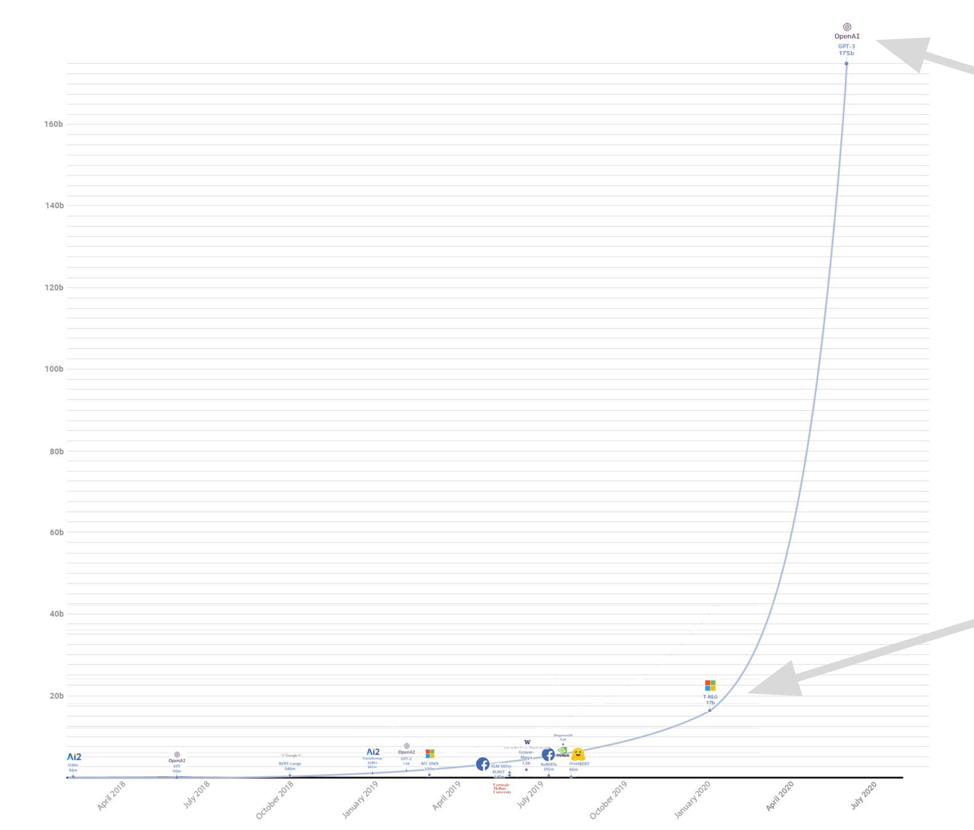
```
[4] print(generator("Paris is the capital of ", max_length=39, num_return_sequences=1)[0]["generated_text"])
```

```
Setting `pad_token_id` to 50256 (first `eos_token_id`) to generate sequence
Paris is the capital of  France and one of the  world's  largest cities of  8 million inhabitants. As its name implies,
Paris also happens to be the home of much of French culture and cuisine.
```

```
[5] print(generator("Lisbon is the capital of ", max_length=39, num_return_sequences=1)[0]["generated_text"])
```

```
Setting `pad_token_id` to 50256 (first `eos_token_id`) to generate sequence
Lisbon is the capital of  Portugal which is located in the northwestern part of Europe. Lisbon is one of three cities
in Portugal that are called the Capital of Portugal, the other two are Porto and Porto-Leon .
```

GPT-3: scaling it up

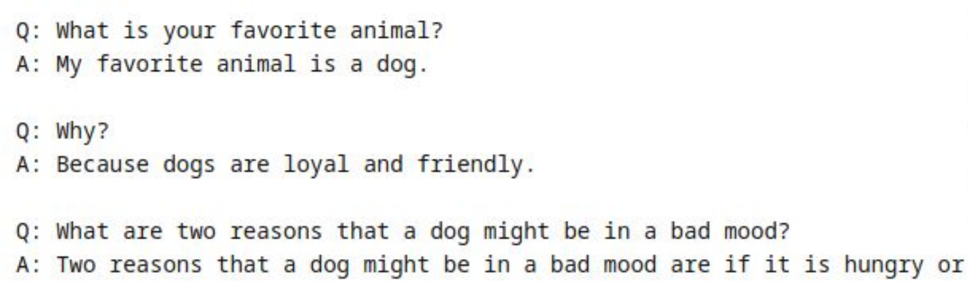


GPT-3: 175B...

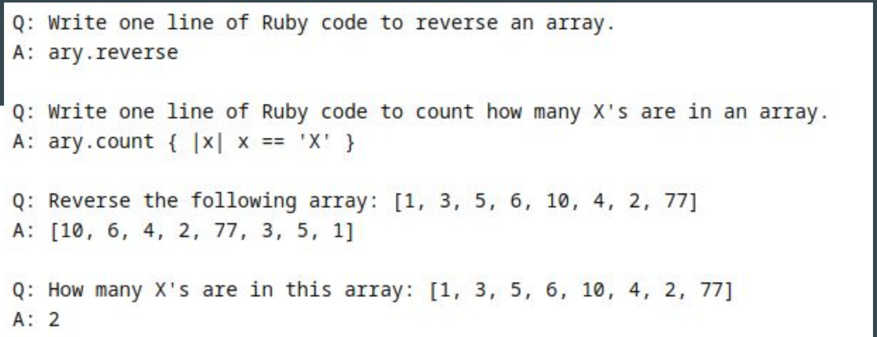
Previous biggest LM:
17B parameters

Image taken from [7].

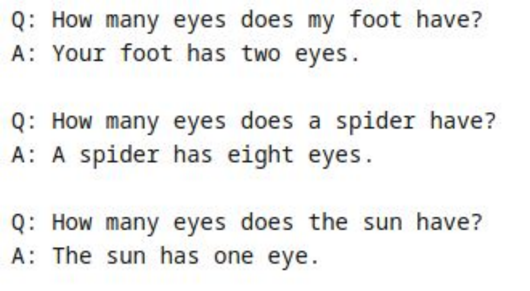
GPT-3 learns all sorts of things



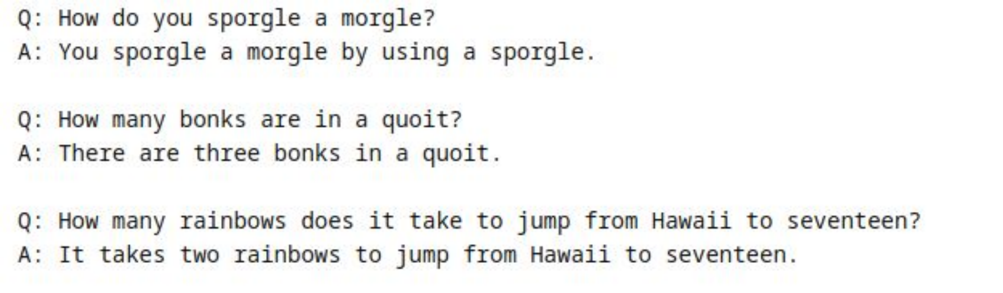
```
Q: What is your favorite animal?
A: My favorite animal is a dog.

Q: Why?
A: Because dogs are loyal and friendly.

Q: What are two reasons that a dog might be in a bad mood?
A: Two reasons that a dog might be in a bad mood are if it is hungry or if it is hot.
```



```
Q: Write one line of Ruby code to reverse an array.
A: ary.reverse

Q: Write one line of Ruby code to count how many X's are in an array.
A: ary.count { |x| x == 'X' }

Q: Reverse the following array: [1, 3, 5, 6, 10, 4, 2, 77]
A: [10, 6, 4, 2, 77, 3, 5, 1]

Q: How many X's are in this array: [1, 3, 5, 6, 10, 4, 2, 77]
A: 2
```



```
Q: How many eyes does my foot have?
A: Your foot has two eyes.

Q: How many eyes does a spider have?
A: A spider has eight eyes.

Q: How many eyes does the sun have?
A: The sun has one eye.
```



```
Q: How do you sporgle a morgle?
A: You sporgle a morgle by using a sporgle.

Q: How many bonks are in a quoit?
A: There are three bonks in a quoit.

Q: How many rainbows does it take to jump from Hawaii to seventeen?
A: It takes two rainbows to jump from Hawaii to seventeen.
```

Images taken from [8].

Transfer Learning

Labeled data is expensive. Can we leverage the abundant online text?

Idea: develop better word embeddings.

What is a bank?

Image taken from [9].

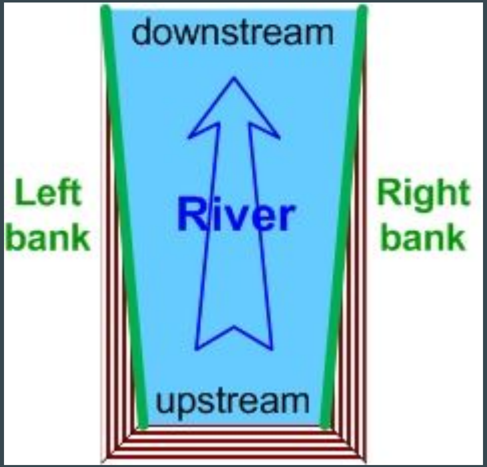
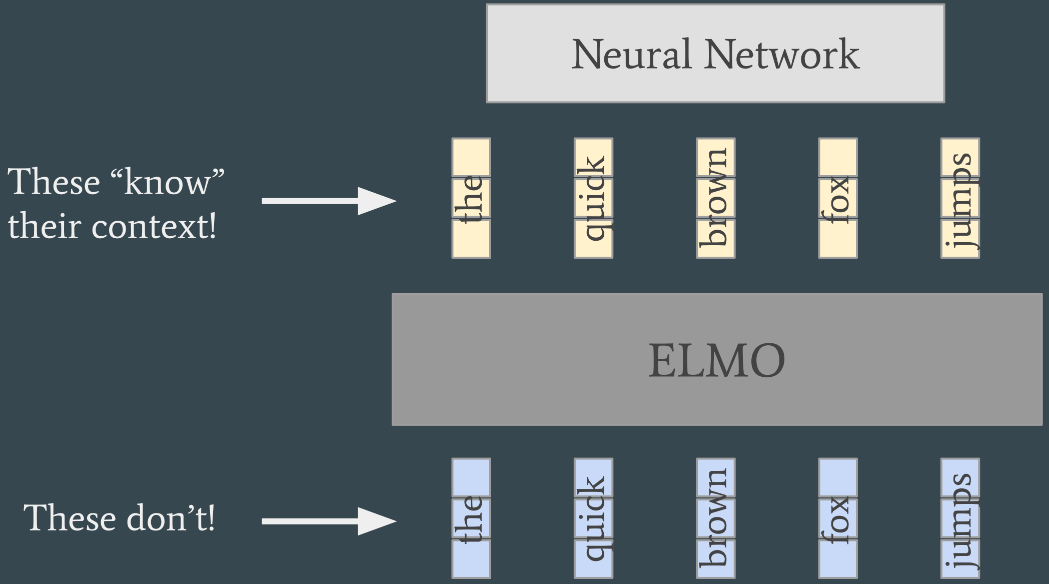


Image taken from [10].

We need context

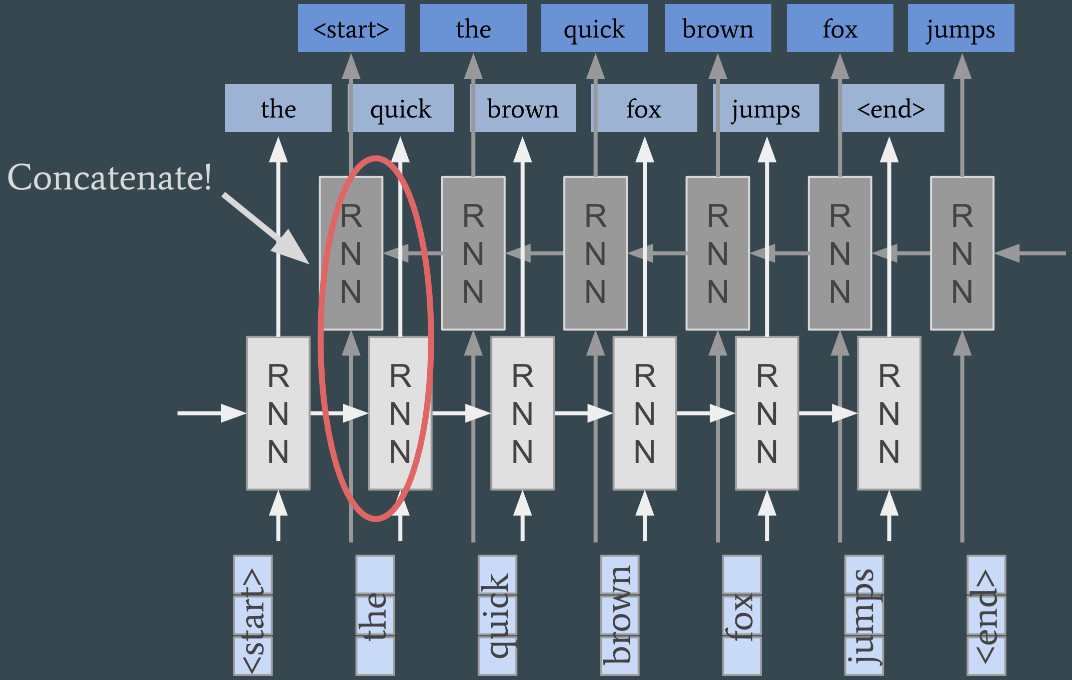
Contextualized embeddings

ELMO: take hidden states of RNN (bi-LSTM), and use them as embeddings!

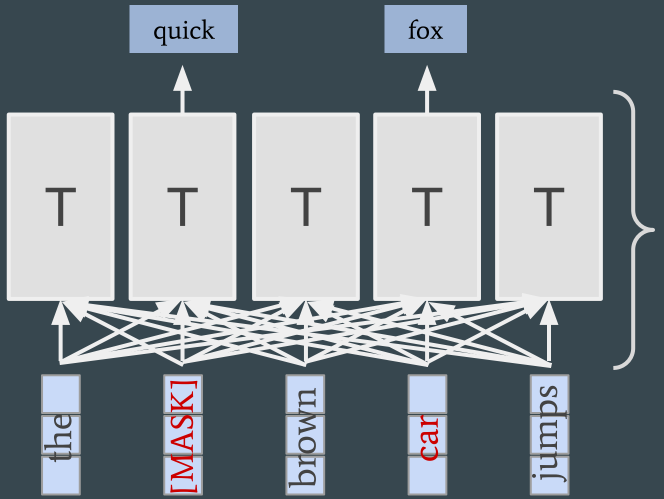


ELMO step by step

Two language models at the same time.



BERT

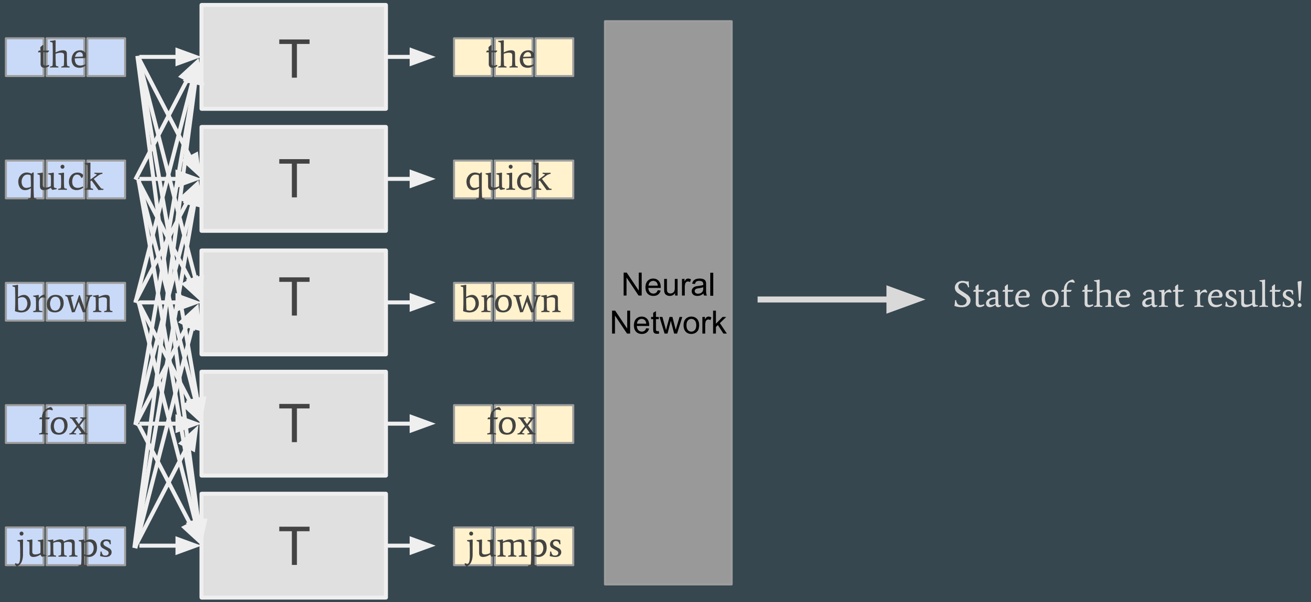


Repeat over many layers!

Not really a language model, but very useful!

(Also next sentence prediction)

BERT



State of the art results!

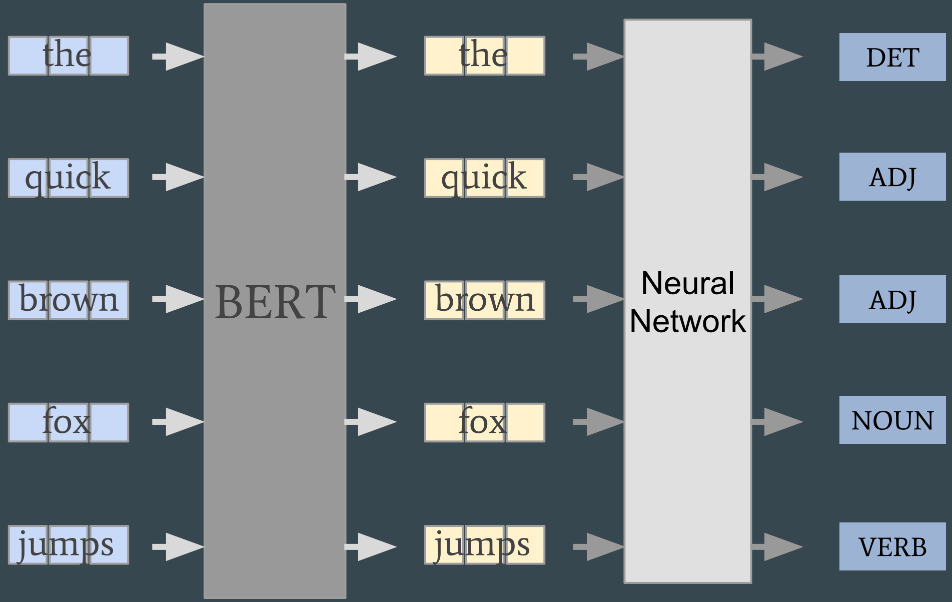
So popular that tons of variants have appeared: ALBERT, RoBERTa, ...

In practice: a part of speech system

Predicting the grammatical category. Useful in many downstream tasks.

The	quick	brown	fox	jumps	over	the	lazy	dog	.
DET	ADJ	ADJ	NOUN	VERB	ADP	DET	ADJ	NOUN	PUNCT

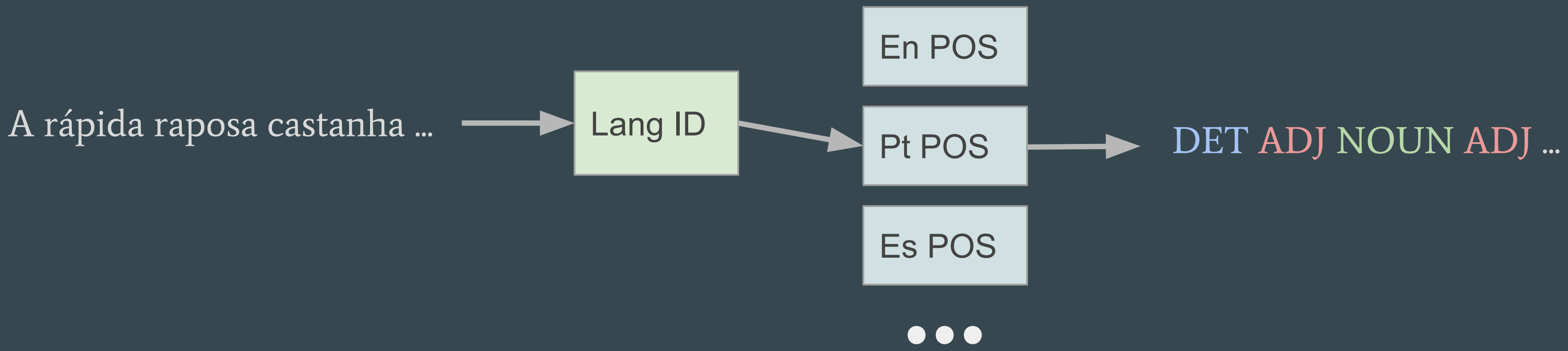
Adapting BERT



Great, but previous (cheaper) techniques already gave good results.

The traditional multilingual pipeline

What if we need to support multiple languages? Easy, just train more models...



A rápida raposa castanha ...

Lang ID

En POS

Pt POS

Es POS

• • •

DET ADJ NOUN ADJ ...

Two problems

1. Lots of models

2. Fails with code-mixing

I thought मौसम different होगाबस fog है

Example from [13].

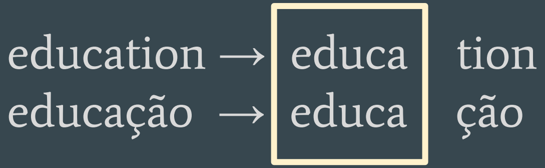
Can a single model handle multiple languages?

Multilingual BERT

It would be great if our representations were language agnostic.

Idea: train on LOTS of languages with a shared wordpiece vocabulary.

Instead of predicting full words, break them into chunks.

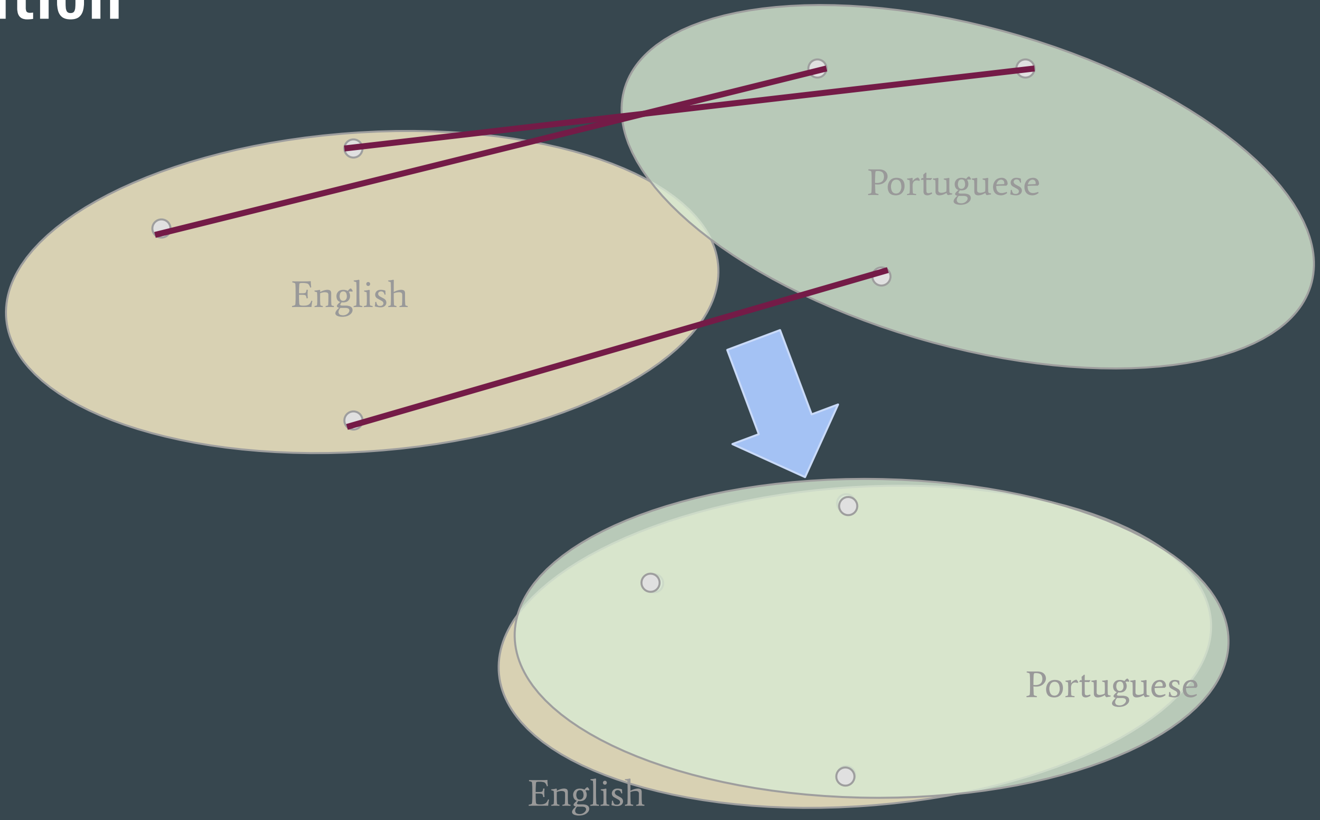


education → educa tion

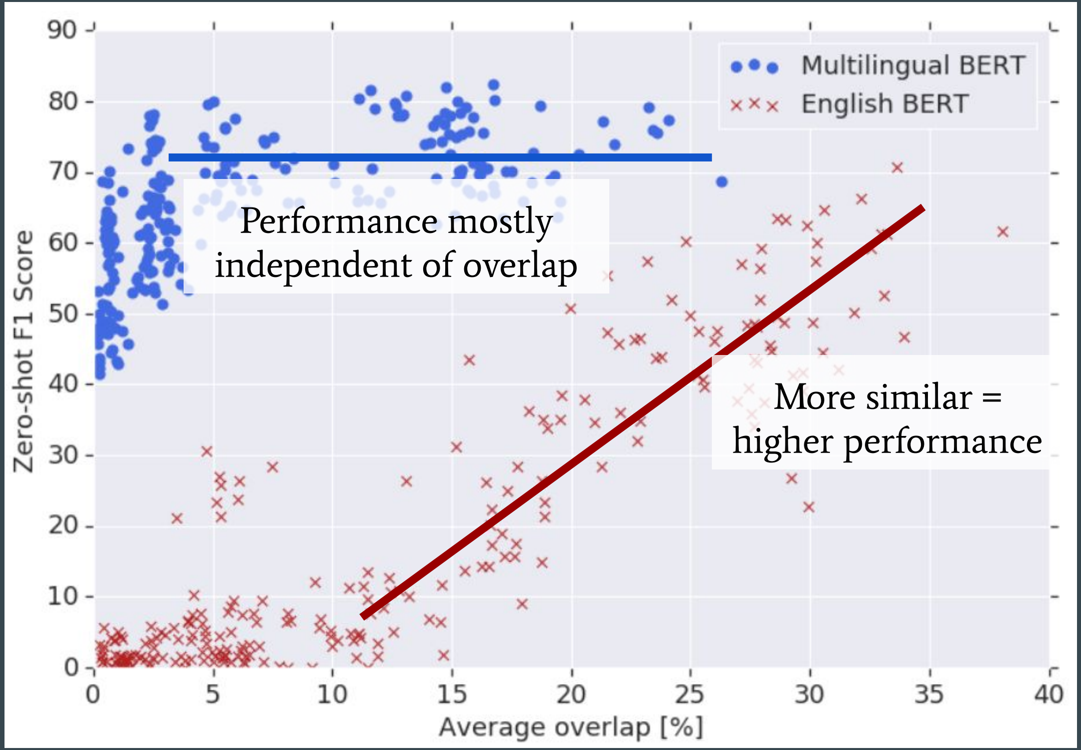
educação → educa ção

Shared for the two languages

Intuition



Strangely it works...



Performance mostly independent of overlap

More similar = higher performance

Image from [13].

Improving the representations

The representations are not perfect. Can we improve them?

Teach the model about translations.

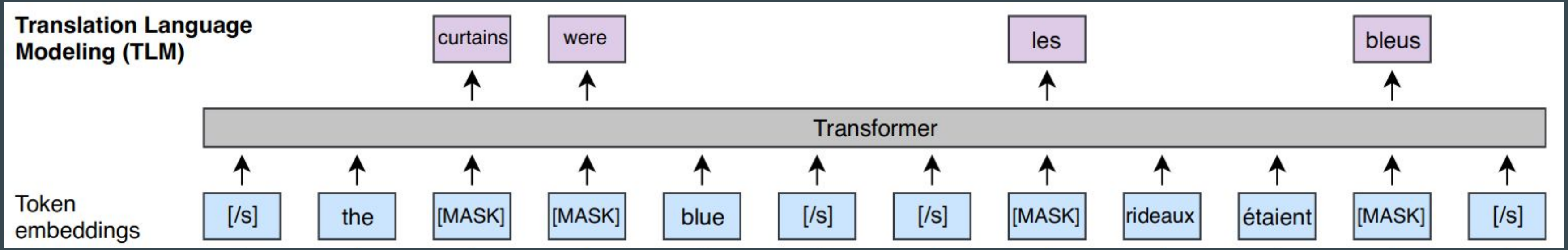
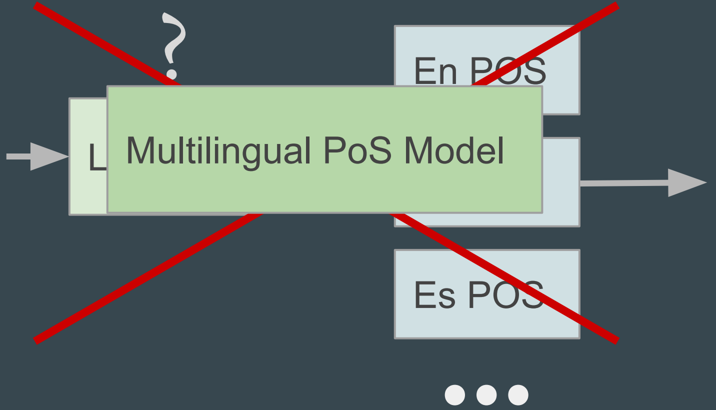


Image adapted from [14].

Back to Part-of-Speech

Ajusta o *learning rate*...



VERB DET NOUN NOUN ...

Less maintenance work, and better results: the best of both worlds!

Summary

A taste of language modeling.

Directly and indirectly useful!

Allow leveraging huge amounts of text for solving NLP tasks.

Also, they’re fun! (see AI Dungeon [15]!)

The End

telmo@apple.com

linkedin.com/in/tjppires

References

[1] Michael Collins' "Language Modeling" lecture notes: http://www.cs.columbia.edu/~mcollins/lm-spring2013.pdf

[2] "A Primer on Neural Network Models for Natural Language Processing": https://arxiv.org/abs/1510.00726

[3] https://openai.com/blog/language-unsupervised/

[4] https://openai.com/blog/better-language-models

[5] "Language Models are Few-Shot Learners": https://arxiv.org/abs/2005.14165

References

[6] https://huggingface.co/gpt2-xl

[7] https://bmk.sh/2020/05/29/GPT-3-A-Brief-Summary/

[8] https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

[9] Bank image: https://en.wikipedia.org/wiki/First_Bank_of_the_United_States

[10] River bank image https://en.wikipedia.org/wiki/Bank_(geography)

[11] “Deep contextualized word representations”: https://arxiv.org/abs/1802.05365

References

[12] “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”: https://arxiv.org/abs/1810.04805

[13] “How Multilingual is Multilingual BERT?”: https://arxiv.org/abs/1906.01502

[14] “Cross-lingual Language Model Pretraining”: https://arxiv.org/abs/1901.07291

[15] “AI Dungeon”: https://play.aidungeon.io/