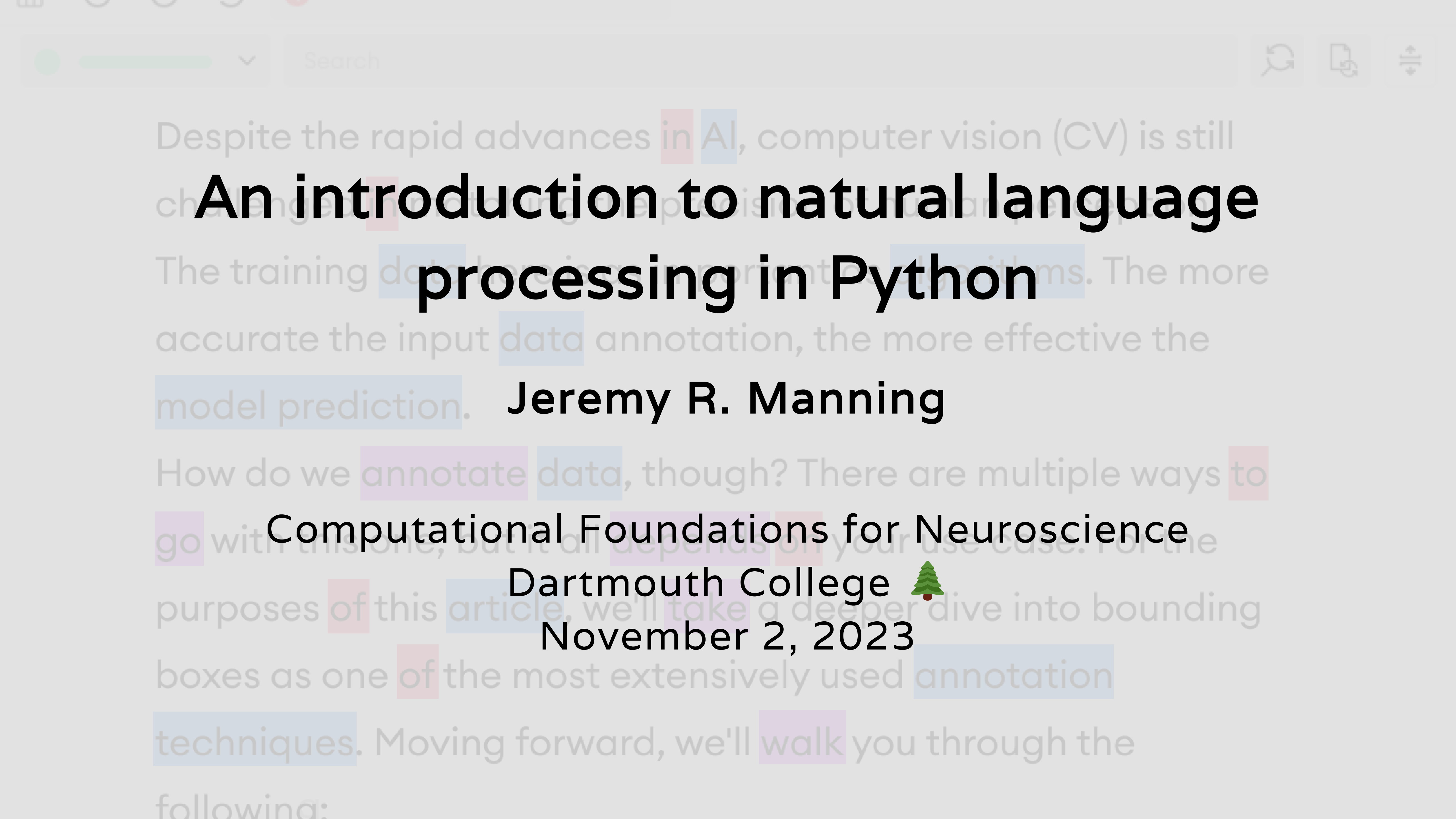
An introduction to natural language processing in Python

Jeremy R. Manning

Computational Foundations for Neuroscience
Dartmouth College 🌲
November 2, 2023



Despite the rapid advances in AI, computer vision (CV) is still challenge [illegible] precision [illegible]
The training data [illegible] algorithms. The more accurate the input data annotation, the more effective the model prediction.

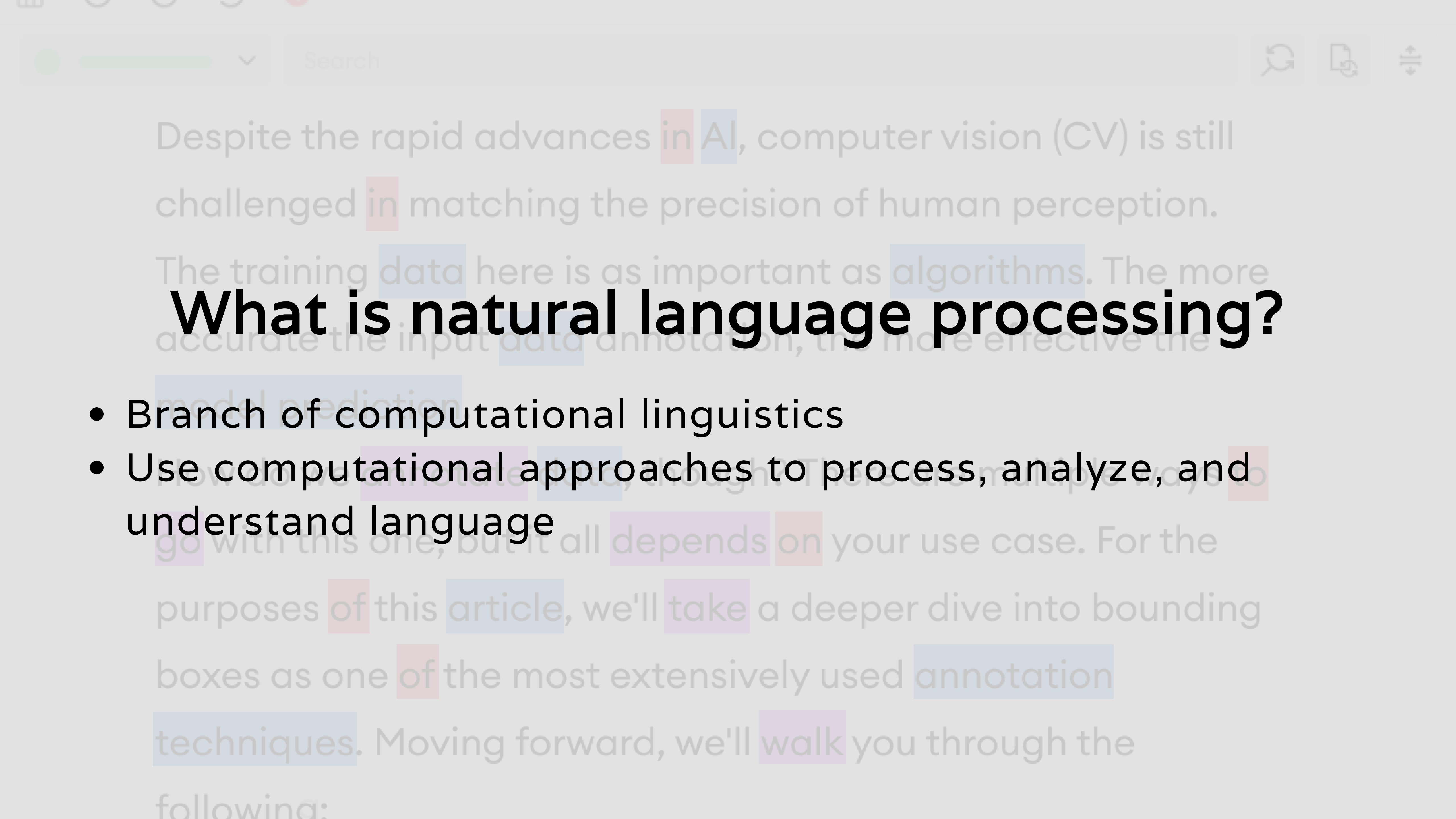
How do we annotate data, though? There are multiple ways to go with [illegible] depends on your use case. For the purposes of this article, we'll take a deeper dive into bounding boxes as one of the most extensively used annotation techniques. Moving forward, we'll walk you through the following:

What’s covered in this tutorial?

- Today’s NLP tutorial provides some background and shows how to apply several different text embedding models to a conversations dataset.
- The tutorial also shows how to implement a simple chatbot using `langchain`.
- Suggestion: open the tutorial in Google Colaboratory and select "Run All" from the "Runtime" menu, while I’m going through the slides. Then you can play around with the different pieces once everything has loaded.

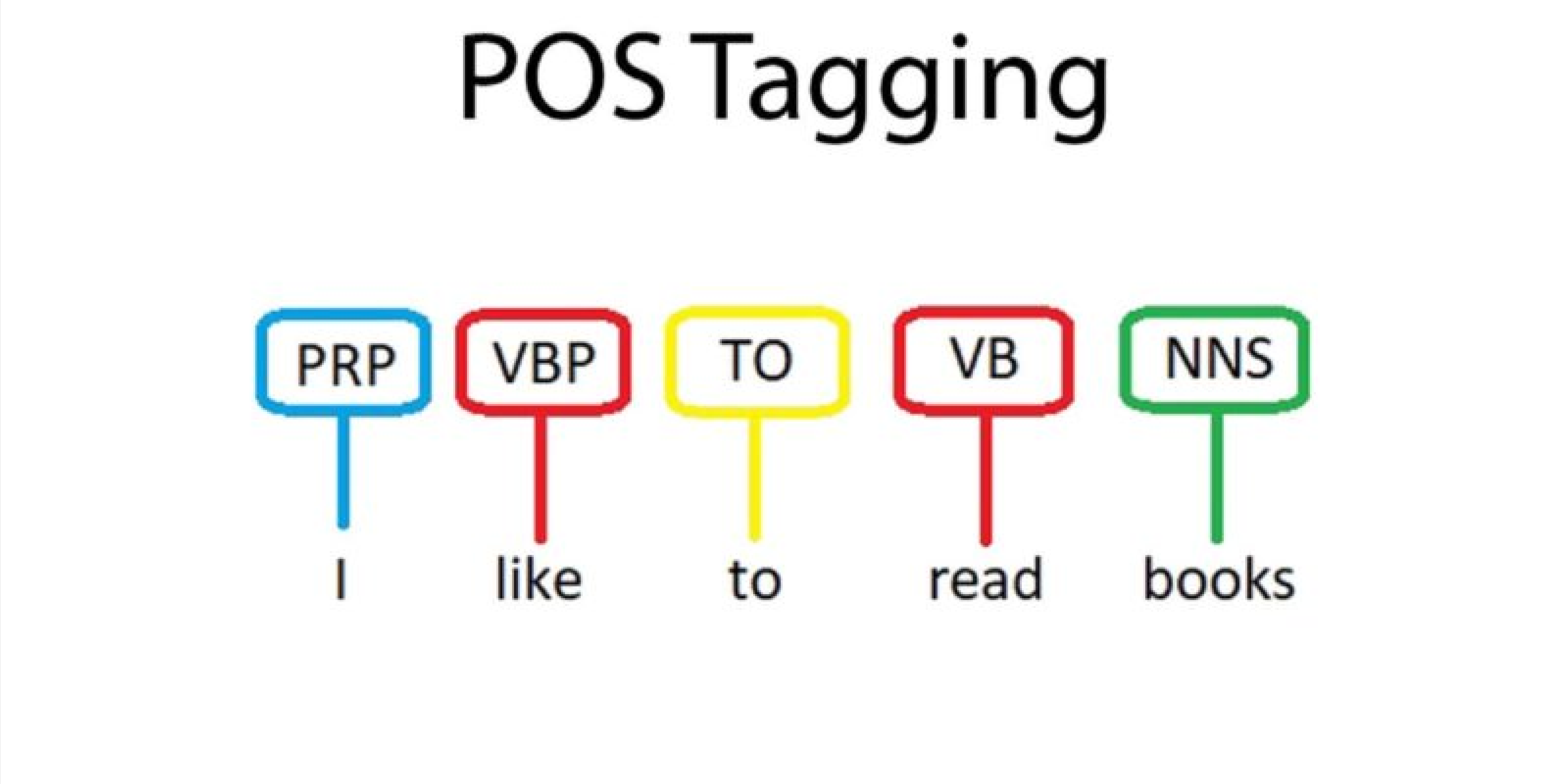
What is natural language processing?

- Branch of computational linguistics
- Use computational approaches to process, analyze, and understand language

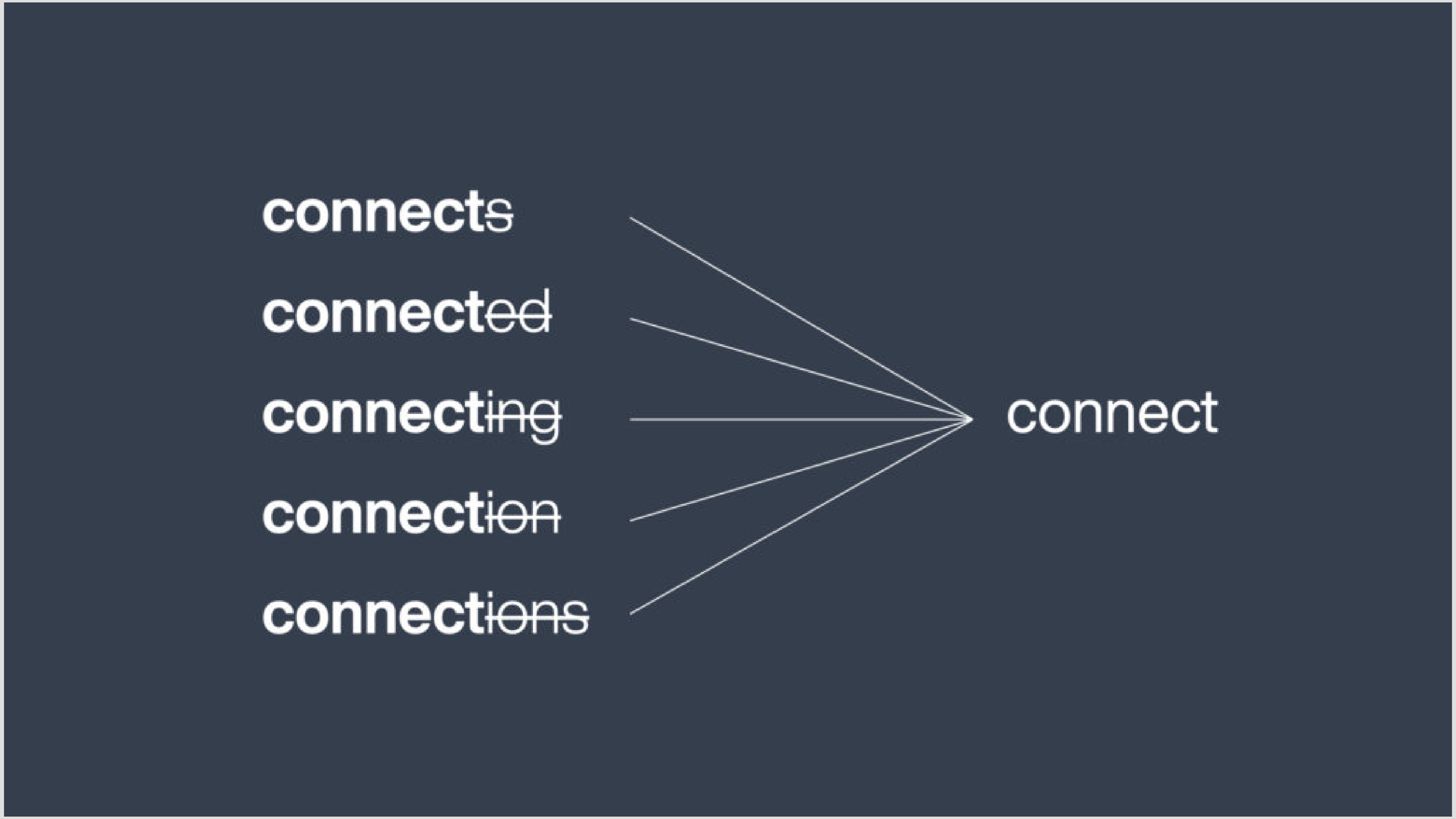


Despite the rapid advances in AI, computer vision (CV) is still challenged in matching the precision of human perception. The training data here is as important as algorithms. The more accurate the input data annotation, the more effective the model prediction. How do we annotate data, though? There are multiple ways to go with this one, but it all depends on your use case. For the purposes of this article, we'll take a deeper dive into bounding boxes as one of the most extensively used annotation techniques. Moving forward, we'll walk you through the following:

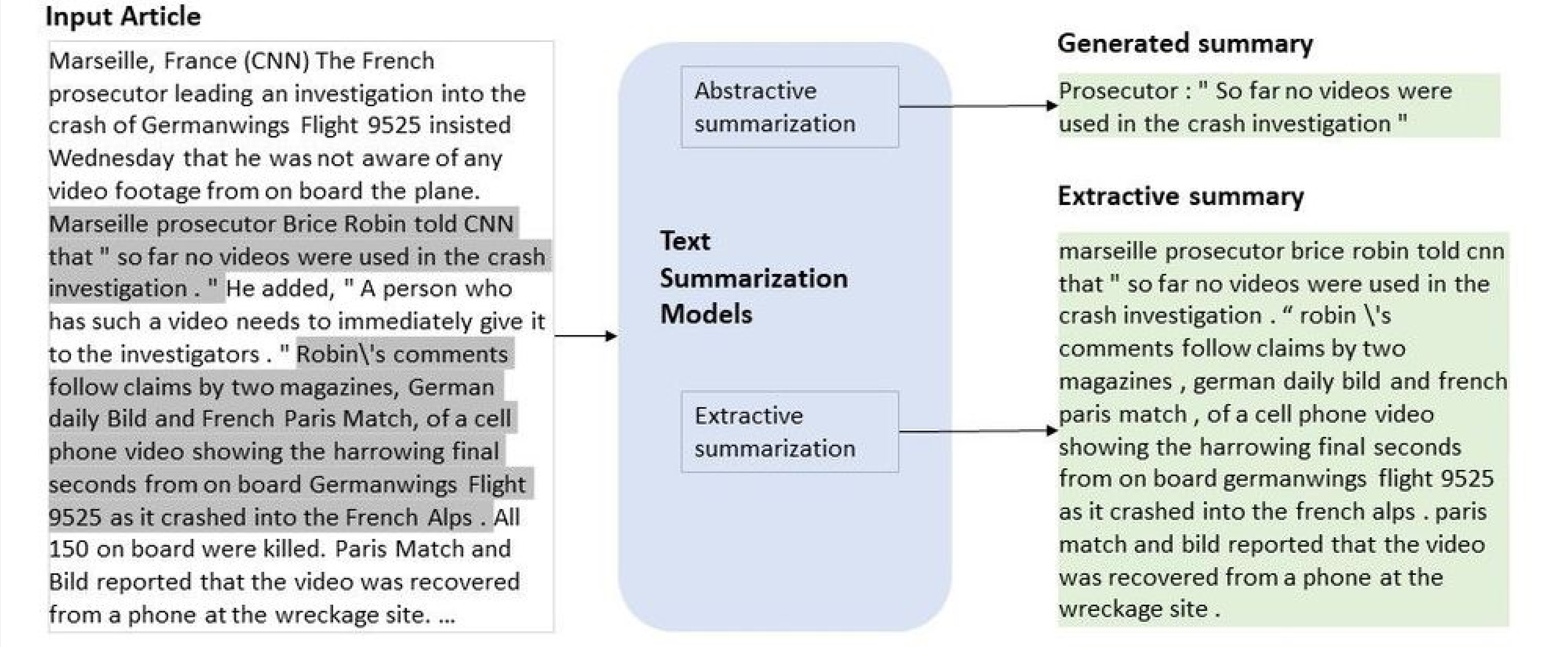
Early tasks: part of speech tagging



Early tasks: stemming/lemmatization



Early tasks: automatic summarization



Input Article

Marseille, France (CNN) The French prosecutor leading an investigation into the crash of Germanwings Flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane. Marseille prosecutor Brice Robin told CNN that " so far no videos were used in the crash investigation . " He added, " A person who has such a video needs to immediately give it to the investigators . " Robin\'s comments follow claims by two magazines, German daily Bild and French Paris Match, of a cell phone video showing the harrowing final seconds from on board Germanwings Flight 9525 as it crashed into the French Alps . All 150 on board were killed. Paris Match and Bild reported that the video was recovered from a phone at the wreckage site. ...

Generated summary

Prosecutor : " So far no videos were used in the crash investigation "

Extractive summary

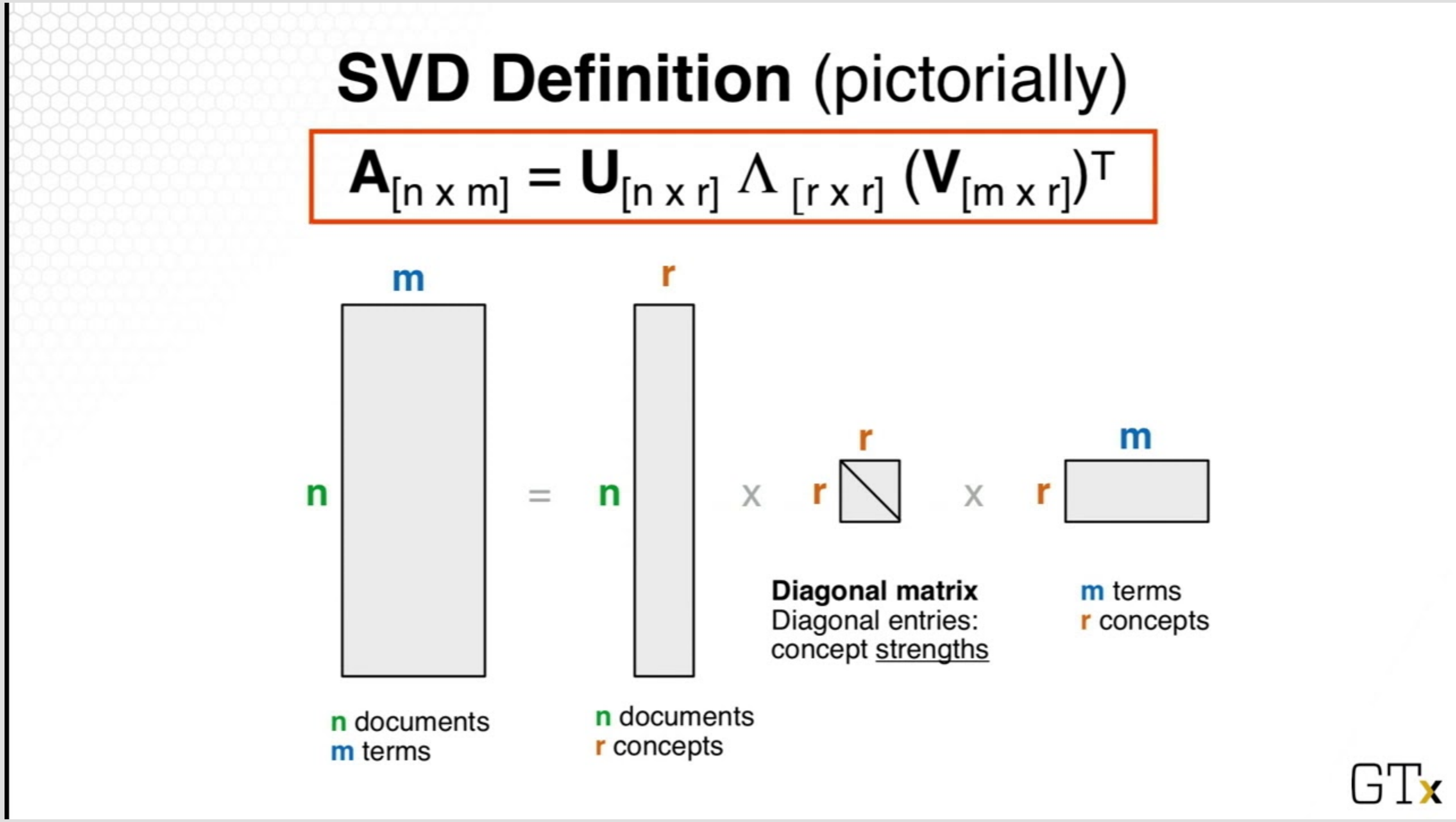
marseille prosecutor brice robin told cnn that " so far no videos were used in the crash investigation . " robin \'s comments follow claims by two magazines , german daily bild and french paris match , of a cell phone video showing the harrowing final seconds from on board germanwings flight 9525 as it crashed into the french alps . paris match and bild reported that the video was recovered from a phone at the wreckage site .

But how can we get at the *meaning* of natural language?

Text embedding models

- Preprocess some text to make a "training corpus"
- Train a model to parse the documents in the corpus
- Goal: generate "feature vectors" (for words, phrases, documents, etc.) that capture semantic properties of the text

Latent semantic analysis



SVD Definition (pictorially)

$$\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \, \Lambda_{[r \times r]} \, (\mathbf{V}_{[m \times r]})^T$$

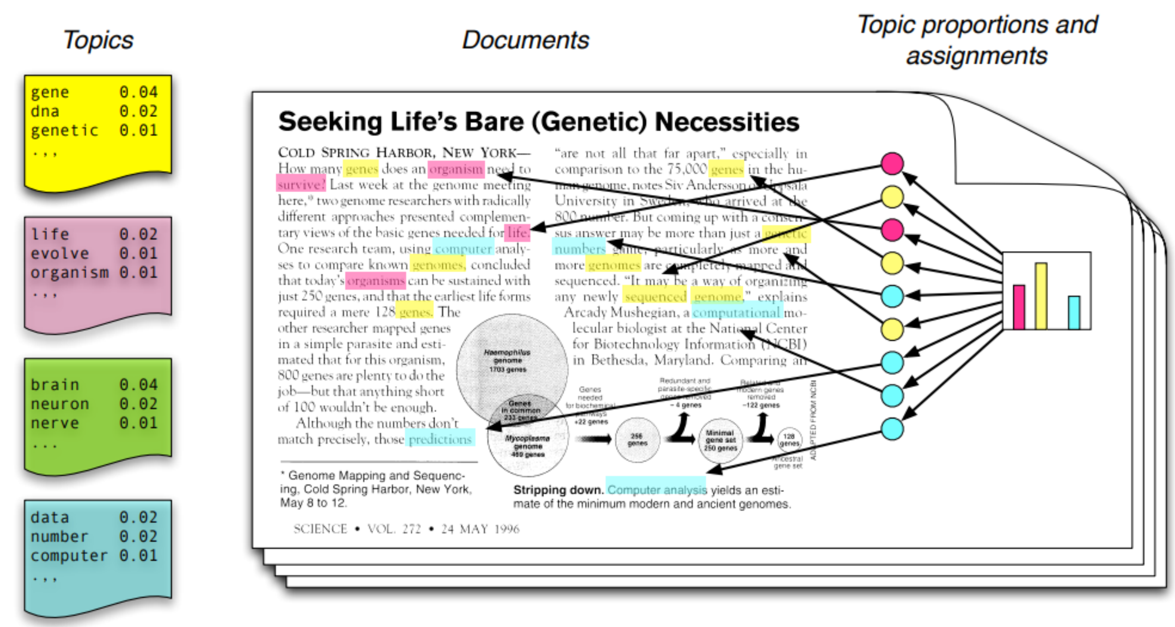
n documents
m terms

n documents
r concepts

Diagonal matrix
Diagonal entries:
concept strengths

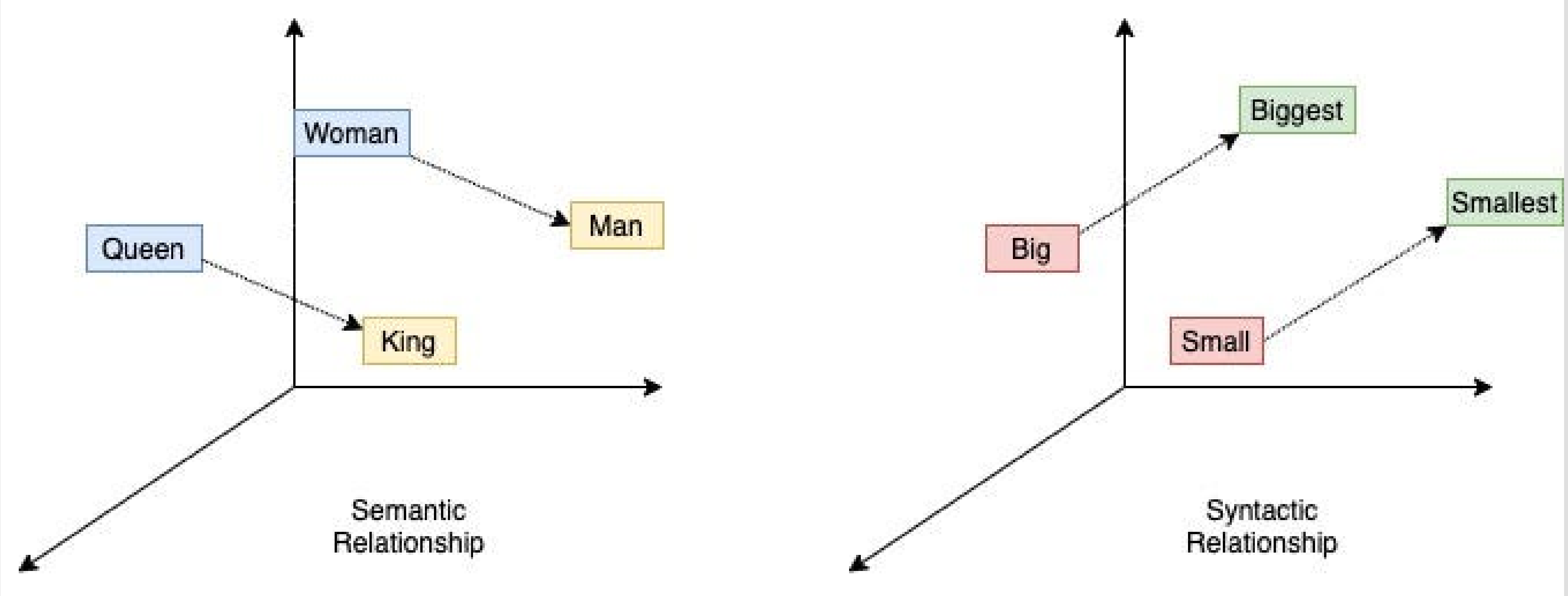
m terms
r concepts

Latent Dirichlet Allocation (LDA)



- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Word2vec


Woman
Man
Queen
King
Semantic Relationship
Biggest
Smallest
Big
Small
Syntactic Relationship

Consider the following phrases:

"My dog ate my homework"

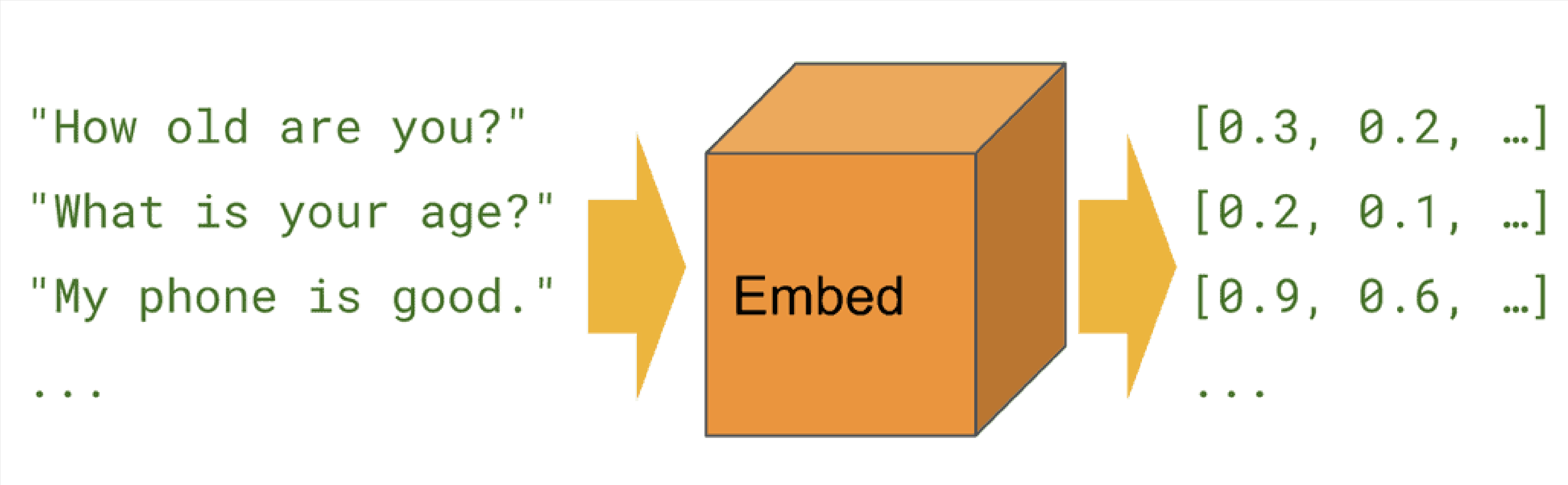
vs.

"My homework ate my dog"

Bag of words vs. context-sensitive models

- BoW models (e.g., LSA, LDA, word2vec) don’t care about word order
- C-S models "care" about *context* and *grammar* by picking up on word order effects

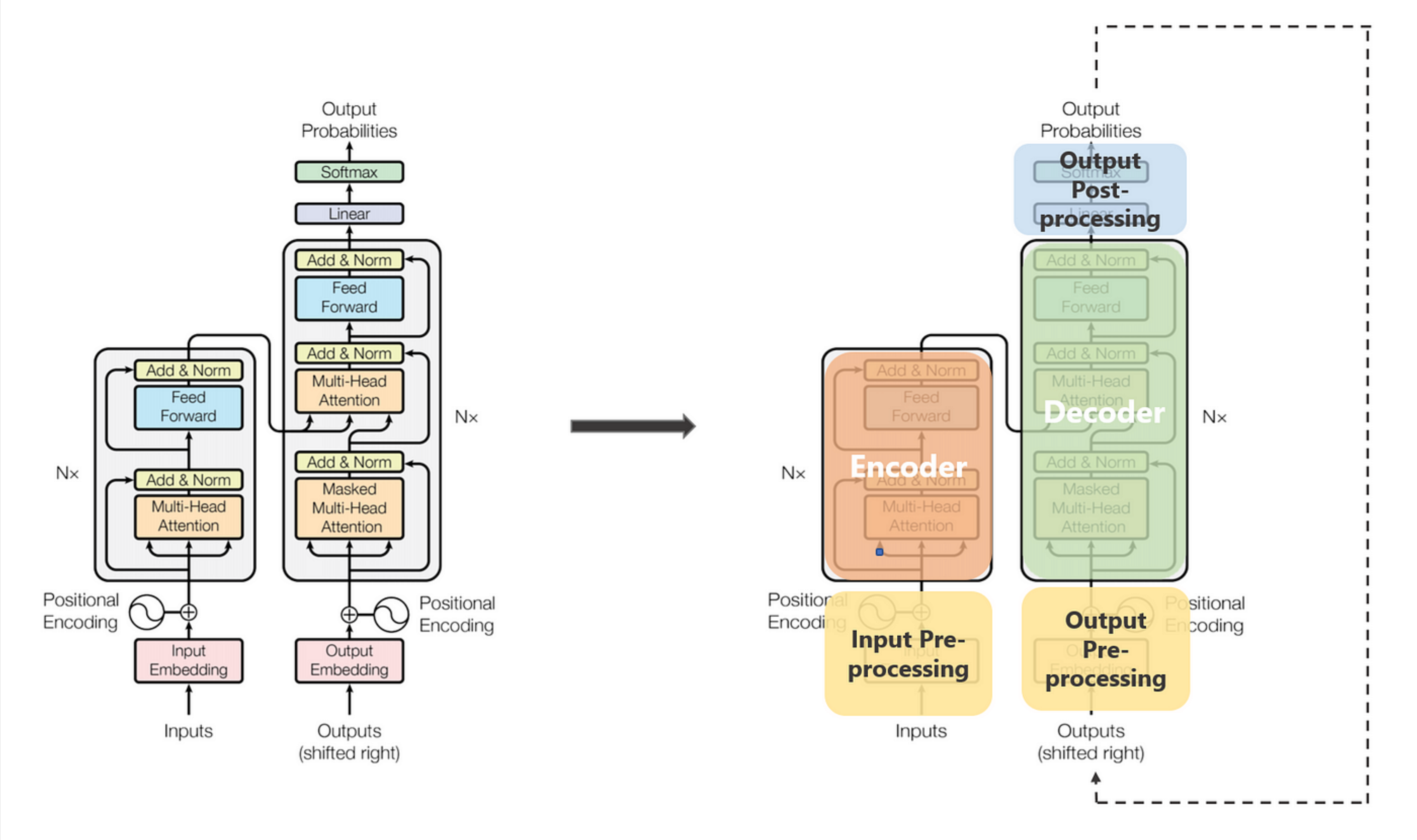
Universal Sentence Encoder



```
"How old are you?"
"What is your age?"
"My phone is good."
...
```

```
[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...
```

Transformers



Transfomers

1. *Tokenize* text into smaller units (words/sub-words)
2. *Embed* tokens (ignoring order)
3. Update the embeddings to include *position* information (add features)
4. Update the embeddings using the *encoder and decoder* layers:
   a. The encoder processes the inputs
   b. The decoder generates outputs

Transformers

- Initially used for "sequence-to-sequence" tasks (e.g., translation)
- Now the basis of most state-of-the-art NLP models

Generative Pretrained Transformer (GPT)

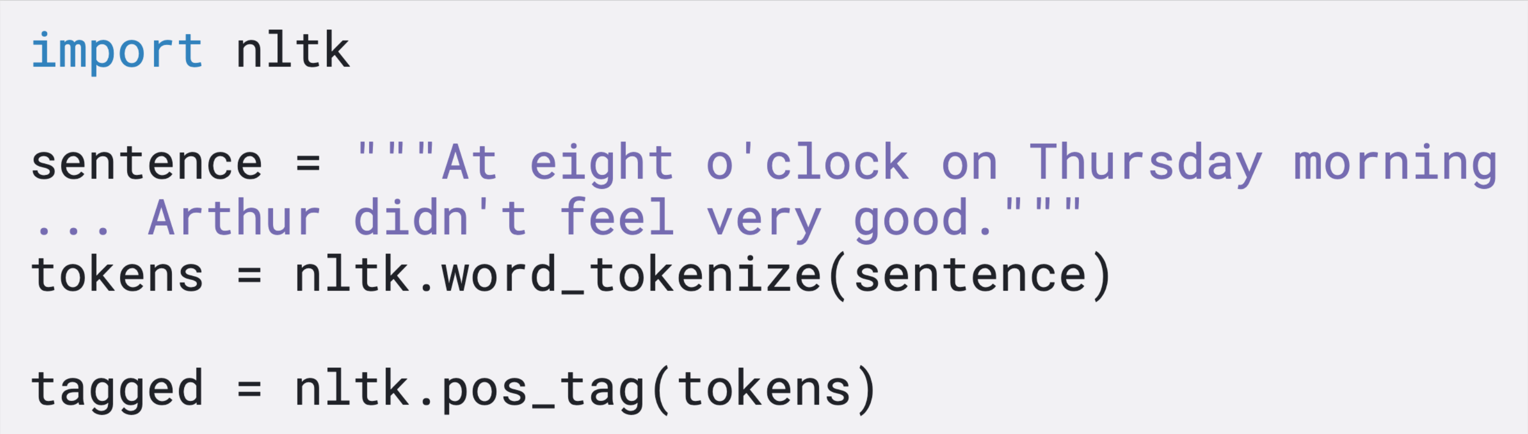
- Variant of transformers
- "Regular" transformers have both an encoder and decoder; GPT only has the decoder part
- The goal is to predict the next token in a sequence, given the previous context
- The model is "pre-trained" on a large corpus of text to learn which tokens are likely to follow which other tokens
- The model can then be "fine-tuned" on specific tasks (e.g., Q&A, following instructions, coding tasks, etc.)
- Goal: text completion, translation, summarization, writing code, etc.

NLP in practice

- Natural Language Toolkit (NLTK) implements lots of basic text processing tasks like tokenization, lemmatization, part of speech tagging, etc.
- Scikit-learn has some basic models, like LDA
- For fancier models the best place to look is Hugging Face

NLTK

Implements lots of fundemantal "traditional" computational linguistics tasks

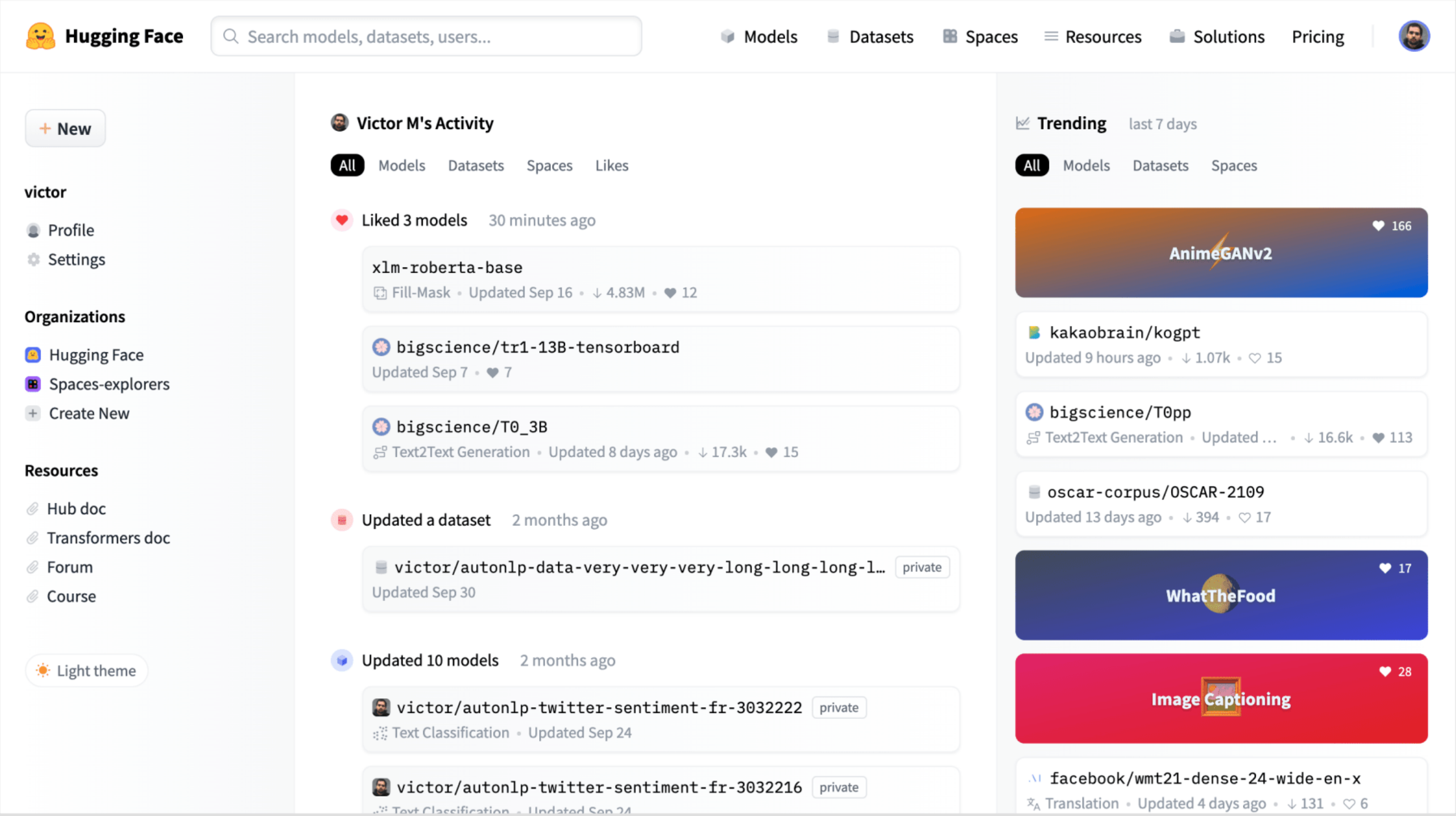


```
import nltk

sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
tokens = nltk.word_tokenize(sentence)

tagged = nltk.pos_tag(tokens)
```

Hugging Face



Hugging Face

Search models, datasets, users...

Models Datasets Spaces Resources Solutions Pricing

+ New

victor

Profile

Settings

Organizations

Hugging Face

Spaces-explorers

+ Create New

Resources

Hub doc

Transformers doc

Forum

Course

Light theme

Victor M's Activity

All Models Datasets Spaces Likes

Liked 3 models 30 minutes ago

xlm-roberta-base
Fill-Mask • Updated Sep 16 • ↓ 4.83M • ♥ 12

bigscience/tr1-13B-tensorboard
Updated Sep 7 • ♥ 7

bigscience/T0_3B
Text2Text Generation • Updated 8 days ago • ↓ 17.3k • ♥ 15

Updated a dataset 2 months ago

victor/autonlp-data-very-very-very-long-long-long-l... private
Updated Sep 30

Updated 10 models 2 months ago

victor/autonlp-twitter-sentiment-fr-3032222 private
Text Classification • Updated Sep 24

victor/autonlp-twitter-sentiment-fr-3032216 private
Text Classification • Updated Sep 24

Trending last 7 days

All Models Datasets Spaces

AnimeGANv2 ♥ 166

kakaobrain/kogpt
Updated 9 hours ago • ↓ 1.07k • ♡ 15

bigscience/T0pp
Text2Text Generation • Updated ... • ↓ 16.6k • ♥ 113

oscar-corpus/OSCAR-2109
Updated 13 days ago • ↓ 394 • ♡ 17

WhatTheFood ♥ 17

Image Captioning ♥ 28

facebook/wmt21-dense-24-wide-en-x
Translation • Updated 4 days ago • ↓ 131 • ♡ 6

`transformers` library (by Hugging Face)

Direct interactions with text and models

```python
from transformers import AutoTokenizer, AutoModelForMaskedLM

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
model = AutoModelForMaskedLM.from_pretrained("bert-base-uncased")
```

`pydata-wrangler` library

- Wrapper for `scikit-learn` and Hugging Face models, focused on computing text embeddings

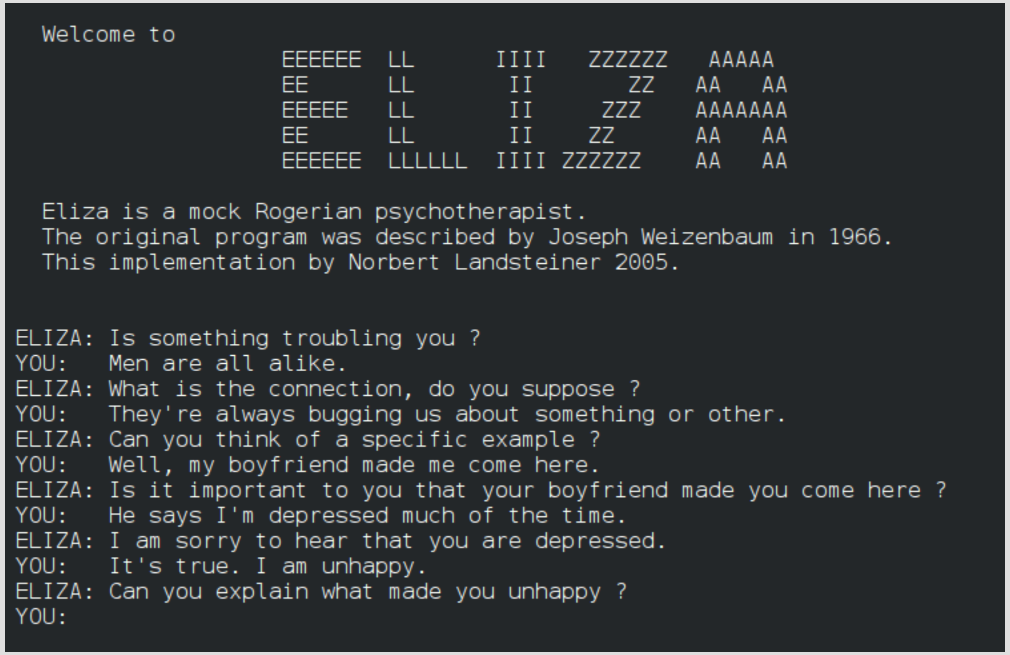
```python
import datawrangler as dw

bert = {'model': 'TransformerDocumentEmbeddings',
        'args': ['bert-base-uncased'],
        'kwargs': {}}
bert_embeddings = dw.wrangle(my_text,
                             text_kwargs={'model': bert})
```

- Lots of other useful tools for working with data (supports numpy, pandas, text, images, and more)
- Essentially the core function (`dw.wrangle`) turns messy data into Pandas DataFrames

Interactive agents (ChatBots)

- Early implementations (e.g., ELIZA) used clever string manipulation hacks



```
  Welcome to
                    EEEEEE  LL      IIII   ZZZZZZ   AAAAA
                    EE      LL       II       ZZ   AA   AA
                    EEEEE   LL       II     ZZZ    AAAAAAA
                    EE      LL       II    ZZ      AA   AA
                    EEEEEE  LLLLLL  IIII ZZZZZZ    AA   AA

  Eliza is a mock Rogerian psychotherapist.
  The original program was described by Joseph Weizenbaum in 1966.
  This implementation by Norbert Landsteiner 2005.


ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

- Modern chatbots: predictive text models (text embeddings, positional coding, etc.)

`langchain` library

- Provides some great tools for interfacing with a wide variety of language models from Hugging Face, OpenAI, Meta, and more
- Infrastructure for chaining together prompts, tasks, models, etc.

Summary

- Text embeddings: bag of words vs. context-sensitive models
- Chatbots: string manipulations vs. predictive text models
- Suggested libraries: `scikit-learn`, `pydata-wrangler`, `transformers`, and `langchain`