



MORPHOLOGY

Luís Coheur

Overview

- Learning objectives
- Topics
 - Linguistic Knowledge
 - Morphology
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Grasp fundamental concepts
- Learn several ways to perform Part-of-Speech (PoS) tagging

TOPICS

Overview

- Learning objectives
- Topics
 - Linguistic Knowledge
 - Morphology
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
- Key takeaways
- Suggested readings

LINGUISTIC KNOWLEDGE

- Phonetic knowledge: relates words to sounds
 - Example: meme. How to pronounce it? ;-)
 - (in Portuguese: Eu almoço o almoço. How to pronounce “almoço”)
- Morphological knowledge: related to the study of the constituents of words
 - Example: if “almoço” is tagged as a verb, you will already know how to pronounce it...

**This class is going to be
dedicated to
Morphology**



LINGUISTIC KNOWLEDGE

- Syntactic: determines how words can be combined to form a sentence
- Semantic Knowledge: used to assign a meaning to each word and to a sentence (literal meaning)
 - Before: Logic
 - Now: vectors (and neural embeddings)
 - Problem: how to represent NL? How to map NL in this representation?

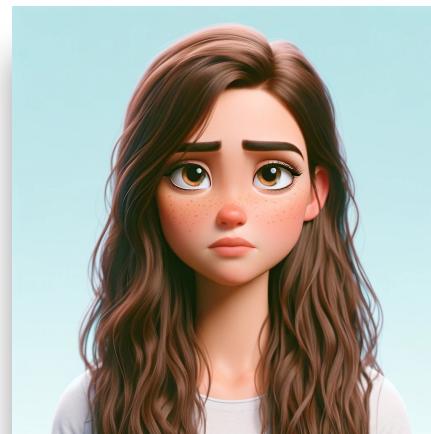
**Next week we
will have a class
dedicated to
Syntax and
another one to
Semantics**



LINGUISTIC KNOWLEDGE

- Pragmatic Knowledge: takes into account the context in the interpretation of a sentence (non-literal meaning).
 - It is so dark! (maybe I want you to turn on the light)

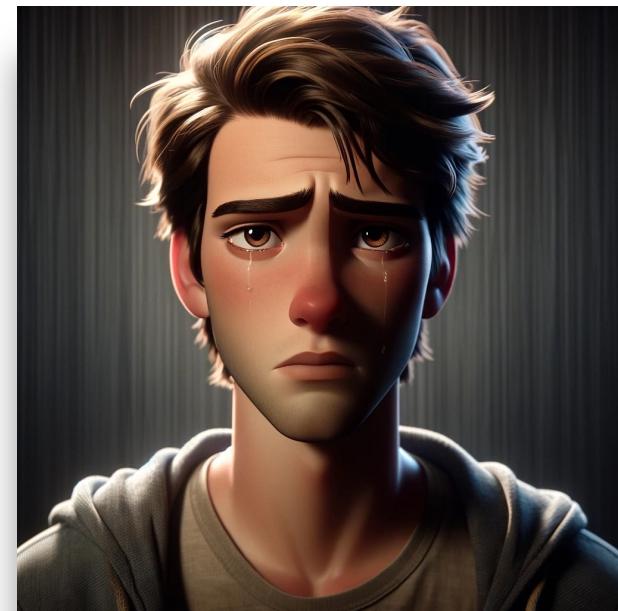
No class
dedicated to
Pragmatic
Knowledge



LINGUISTIC KNOWLEDGE

- Discourse knowledge: used to determine the influence of the preceding sentences on the interpretation of the current sentence (e.g., pronouns and temporal information)
 - John loves **his sister**. She **is so nice**.

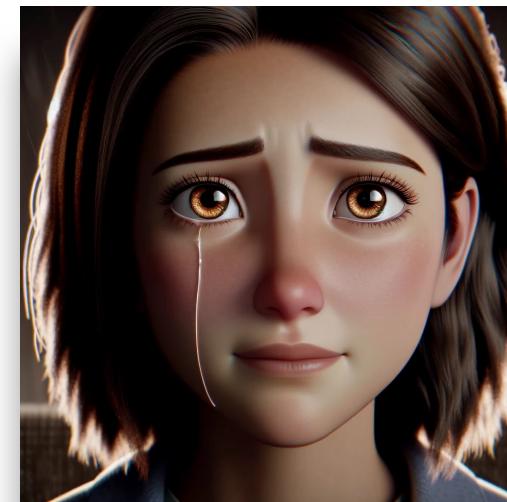
**No Discourse
Knowledge**



WORLD KNOWLEDGE (common sense)

- John was shot in the eye, and his brain came out of his ear
- => he died

No World
Knowledge (but
check Open
Mind Common
Sense... or
ChatGPT)



Overview

- Learning objectives
- Topics
 - Linguistic Knowledge
 - Morphology
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
- Key takeaways
- Suggested readings

WORD CLASSES (or PART-OF-SPEECH)

- A [Word Class](#) is a category into which words are grouped considering their function within a sentence

WORD CLASSES

- Nouns: name people, places, things, ideas, or concepts
- Pronouns: take the place of nouns
- Verbs: express actions, states, or occurrences
 - Action verb: She **writes** a letter.
 - State verb: He **knows** the answer. She **is** tired.
 - Occurrence verb: The sun **rose** early today.
- Adjectives: describe or modify nouns or pronouns
- Adverbs: modify verbs, adjectives, or other adverbs

WORD CLASSES (cont.)

- Prepositions: typically indicating location, direction, time, or manner
 - Examples:
 - Location: The book is **on** the table.
 - Direction: He walked **to** the park.
 - Time: We will meet **at** 5 PM.
 - Manner: She spoke **with** confidence.
- Conjunctions: join words, phrases, clauses, or sentences
- Interjections: express emotions

Remember?



ACTIVE LEARNING MOMENT



EXERCISE

- Consider the sentence (by ChatGPT):

"Wow", Sarah carefully hands her friend the red book from the shelf, and smiles.

- Find:

- Adjective(s):
- Adverb(s):
- Conjunction(s):
- Common noun(s):
- Determiner(s):
- Interjection(s):
- Preposition(s):
- Proper noun(s):
- Pronoun(s):
- Verb(s):

EXERCISE

- Consider the sentence (by ChatGPT):

"Wow", Sarah carefully hands her friend the red book from the shelf, and smiles.

- Find:

- Adjective(s): "red"
- Adverb(s): "carefully"
- Conjunction(s): "and"
- Common noun(s): "friend", "book" and "shelf"
- Determiner(s): "the"
- Interjection(s): "Wow"
- Preposition(s): "from"
- Proper noun(s): "Sarah"
- Pronoun(s): "her"
- Verb(s): "hands" and "smiles"

Overview

- Learning objectives
- Topics
 - Linguistic Knowledge
 - Morphology
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
- Key takeaways
- Suggested readings

PART-OF-SPEECH TAGGING

- Part-of-Speech (PoS) tagging is the task of assigning a word class to each word in a text
- By the way: not to confuse with Morphological Analysis
 - we will talk about this next

PART-OF-SPEECH TAGGING

- Challenges:
 - There is not a single tag set for part-of-speech tags
 - Words are ambiguous (Example: book)
- Good News:
 - Just a small percentage of words are ambiguous!
- Bad news:
 - Ambiguous words are the most frequent words!
- Example from the Brown Corpus:
 - 11,5% of the words are ambiguous (form)
 - 40% of the words in the corpus are ambiguous.

SOME APPROACHES TO PART-OF-SPEECH TAGGING

- Rule-based
- Stochastic
- Deep Learning

Where have I seen this before?



RULE-BASED PART-OF-SPEECH TAGGING

- The algorithms operate in two steps:
 - STEP 1: with the help of a dictionary, a list of potential tags is assigned to each word
 - STEP 2: the tag to be assigned is chosen based on sets of rules usually designed by humans (otherwise automatically extracted from data annotated by humans)

RULE-BASED PART-OF-SPEECH TAGGING

- Example: He had a book
 - STEP 1:
 - He he/pronoun
 - Had have/verb
 - A a/article
 - Book book/noun book/verb
 - STEP 2
 - "Rule XPTO: If the previous tag is an article, then eliminate the verb tag."
 - Thus: Book book/noun

SOME APPROACHES TO PART-OF-SPEECH TAGGING

- ~~Rule-based~~
- Stochastic
- Deep Learning

STOCHASTIC PART-OF-SPEECH TAGGING

- Goal: choose the best sequence of tags

$$T = t_1 t_2 \dots t_n$$

for a given sentence (sequence of words)

$$W = w_1 w_2 \dots w_n$$

- That is: calculate the most likely (highest probability) tag sequence for a sequence of words

STOCHASTIC PART-OF-SPEECH TAGGING

- We will estimate $P(T | W)$ with an HMM tagger
- HMM taggers make two simplifying assumptions:

$$P(T|W) \approx \prod_{i=1}^n \underline{P(t_i|t_{i-1})} \times \underline{P(w_i|t_i)}$$

Transition: bigram
assumption

Emission: the probability
of a word depends only
on its own tag

STOCHASTIC PART-OF-SPEECH TAGGING

What is
an HMM
tagger?



STOCHASTIC PART-OF-SPEECH TAGGING

- A Hidden Markov Model (HMM) is a statistical model that describes a system with unobservable (hidden) states (in our case, the tags) through observable sequences (in our case the words), with the transitions among states being characterized by certain probabilities.



Oh, it is raining outside!

STOCHASTIC PART-OF-SPEECH TAGGING

- To calculate transitions' probabilities, we use:

$$P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \rightarrow \begin{array}{l} \text{Number of times } t_i \\ \text{follows } t_{i-1} \end{array}$$

- To calculate the emissions' probabilities, we use:

$$P(w_i | t_i) = \frac{C(t_i, w_i)}{C(t_i)} \rightarrow \begin{array}{l} \text{Number of times } t_i \text{ is} \\ \text{the tag of word } W_i \end{array}$$

ACTIVE LEARNING MOMENT



EXERCISE

- In the Brown corpus:
 - The DT tag occurs 116,454 times and appears before an NN 56,509 times. Then:
 - $P(NN | DT) =$
 - The VBZ tag occurs 21,627 times, and VBZ is the tag for “is” 10,073 times. Then:
 - $P(is | VBZ) =$

EXERCISE

- In the Brown corpus:
 - The DT tag occurs 116,454 times and appears before an NN 56,509 times. Then:
 - $P(NN | DT) = C(DT, NN)/C(DT) = 56,509/116,454 = 0.49$
 - The VBZ tag occurs 21,627 times, and VBZ is the tag for “is” 10,073 times. Then:
 - $P(is | VBZ) = C(VBZ, is)/C(VBZ) = 10,073/21,627 = 0.47$

STOCHASTIC PART-OF-SPEECH TAGGING

- After the counts, HMMs usually take advantage of the Viterbi algorithm for decoding

VITERBI



**What kind of algorithm?
What does it do?**

STOCHASTIC PART-OF-SPEECH TAGGING

- Viterbi:
 - uses **dynamic programming**
 - seeks the **best path** for a given observation, using:
 - the probability of the previous path
 - the transition probability

VITERBI

```
i ← 1
while i < N do
    SS(i, 1) = P(w1 | Li) * P(Li | < s >)
    BP(i, 1) = 0
    i ++
end while
t ← 2
while t < n do
    i ← 1
    while i < N do
        SS(i, t) = maxj=1,...,N SS(j, t-1) * P(Li | Lj) * P(wt | Li)
        BP(i, t) = j that resulted in the maximum score
        i ++
    end while
    t ++
end while
C(n) = i that maximizes SS(i, n)
i ← n - 1
while i > 1 do
    i --
    C(n) = BP(C(i+1), i+1)
end while
```

- N = number of tags
- n = number of words in the sequence
- Data structures:
 - SS (sequence score) – records the score of the best sequence found up to a given position with category L.
 - BP (Back Pointer) – records the previous state to a given state
 - C – records the best sequence of tags.

A quick look at Viterbi

Note:
Fictitious
values

SS	John	likes	Mary
Noun	0.6	0.015	0.6
Verb	0.3	0.045	0.0735

BP			
Noun	0	1 or 2	2
Verb	0	1	1

- Soon, in a lab near you



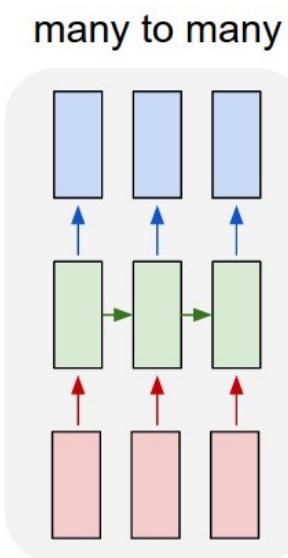
SOME APPROACHES TO PART-OF-SPEECH TAGGING

- ~~Rule-based~~
- ~~Stochastic~~
- Deep Learning

DEEP LEARNING PART-OF-SPEECH TAGGING

- PoS is a **sequence labelling** task
- RNNs, LSTMs and other architectures can be used to train a PoS model:
 - The model is trained on a labelled dataset
 - Each word is tagged with its correct PoS
 - The model learns to predict the tag of each word based on the context and in the word itself

Remember?



BY THE WAY...

- You can also use ChatGPT (and friends)
 - Some people consider that these “old” NLP tasks are good for evaluating current LLMs

Hum...
interesting...





Overview

- Learning objectives
- Topics
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
 - Morphemes
 - Building Words
 - Morphological analysis
- Key takeaways
- Suggested readings

MAIN CONCEPTS

- Morphology is the linguistics field dedicated to the study of the internal structure of words (morph = shape, logos = word)
- Words are constituted by (meaningful) units called morphemes

MAIN CONCEPTS

- There are two types of morphemes:
 - **Stems** (or lexical morphemes): carry the primary meaning of words
 - **Affixes** (or grammatical morphemes): change stems meaning and/or have grammatical functions
 - Example:

REUSABLE

Stem: Use
Affixes:

- re- (meaning again or back)
- -able (indicating capability)

MAIN CONCEPTS

- Affixes' types:
 - Prefixes: beginning of the word
 - Examples:
 - Adding “un-” to “happy” creates “unhappy”
 - Adding “re-” to “write” creates “rewrite”
 - Suffixes: end of the word
 - Example:
 - Adding “-ness” to “happy” creates “happiness”
 - Adding “-ly” to “quick” creates “quickly”

MAIN CONCEPTS

- More affixes' types:
 - Infixes: inserted inside the stem
 - Example:
 - Inserting “-[freaking](#)-” into “unbelievable” as in “un-freaking-believable” (English slang example by ChatGPT)
 - Editor-in-chief + s -> Editors-in-chief

MAIN CONCEPTS

- More affixes' types:
 - Circumfixes: precede and follow the stem; inserted at the same time
 - Examples:
 - In Portuguese: entardecer, amanhecer, embelezar
 - German (example by ChatGPT):
 - Root: lieb ("love" or "dear")
 - Circumfix: ge- ... -t
 - Word: geliebt ("loved")
 - Clitics: function like a word, but do not appear alone
 - Example:
 - Eu contei-os. (I counted them)

MAIN CONCEPTS

- In some languages words can contain an impressive number of morphemes
 - Example: Turkish, for instance, has many words with 9 or 10 morphemes

EXAMPLE

(from Prof. Nuno Mamede slides)

Turkish has lots of affixes

Avrupa	Europe
Avrupalı	of Europe / European
Avrupalılış	become European
Avrupalılıştı́r	Europeanise
Avrupalılıştı́rama	be unable to Europeanise
Avrupalılıştı́ramadık	we couldn't Europeanise
Avrupalılıştı́ramadık	one that is unable to be...
Avrupalılıştı́ramadıklar	unable to be Europeanised ones
Avrupalılıştı́ramadıkları́mız	they, who we couldn't manage to...
Avrupalılıştı́ramadıkları́mızdan	of them who we couldn't manage to Europeanise
Avrupalılıştı́ramadıkları́mızdanmış	is reportedly of ours that we were unable to Europeanise
Avrupalılıştı́ramadıkları́mızdanmışsınız	you are reportedly of ours that we were unable to Europeanise
Avrupalılıştı́ramadıkları́mızdanmışsınızcasına	as if you were reportedly of ours that we were unable to Europeanise

EXAMPLE

(generated by ChatGPT – use edit distance to compare with Wikipedia example and check its veracity)

- Turkish Word:

Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmişsinizc
esine

- Meaning: As if you are among those whom we may not be able to easily make into a maker of unsuccessful ones
- Breaking it down:
 - Stem: Muvaffak (successful)
 - Affixes: -iyet (noun-forming suffix related to doing the action), -siz (without), -leş (become), -tir (cause/make), -ici (agent or doer), -leş (become), -tir (cause/make), -iver (sudden action), -eme (cannot), -yebil (ability), -ecek (future tense), -ler (plural), -imiz (our), -den (from), -miş (past participle), -siniz (you are), -cesine (as if)

ACTIVE LEARNING MOMENT



EXERCISE

- Find (if possible) words in your own native language in which one of the morphemes is a:
 - prefix
 - suffix
 - infix (inside the stem)
 - circumfix (inserted at the same time before and after the stem)
 - clitic

Overview

- Learning objectives
- Topics
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
 - Morphemes
 - [Building Words](#)
 - Morphological analysis
- Key takeaways
- Suggested readings

MAIN CONCEPTS

- Building words from a word stem:
 - **Inflection:** Doesn't change the word class or the meaning of the word, considering the original stem
 - Examples:
 - eats from eat (both verbs) and gatas (female cats) from gato (cat) (both nouns).
 - **Derivation:** Results in a word from a different word class or with a “different meaning”
 - Examples:
 - do from undo (opposite meanings) and amigável (friendly) from amigo (friend) (adjective and noun).

MAIN CONCEPTS

- More ways of building words from a word stem:
 - **Compounding:** Combination of multiple word stems
 - Examples:
 - doghouse (dog + house) and chapéu-de-chuva (chapéu + de + chuva) (umbrella – something like “hat for rain”)
 - **Cliticization:** Words with clitics
 - Example: apagou-o (erased it or turned it off).
 - ...

ACTIVE LEARNING MOMENT



EXERCISE

- Find (if possible) words in your own native language that are formed based on:
 - Inflection (no changes in the word class/meaning, considering the stem)
 - Derivation (changes)
 - Compounding (multiple word stems)
 - Cliticization (words with clitics)

Overview

- Learning objectives
- Topics
 - Word Classes
 - Part-of-speech tagging
 - Morphology Main Concepts
 - Morphemes
 - Building Words
 - Morphological analysis
- Key takeaways
- Suggested readings

MORPHOLOGICAL ANALYSIS

- Analyzes the **structure of a word** to understand its components (stem, prefixes, etc.), and how these contribute to the word's meaning and grammatical function.
- Subtasks:
 - Lemmatization: finding the **dictionary form** of a word
 - Example:
 - has → have
 - running → run
 - Stemming: **reducing a word to its stem/root form** (may not be a valid word on its own)
 - Example:
 - "running" → runn
 - ...

MORPHOLOGICAL ANALYSIS

- We will not study any form of Morphological Analysis



KEY TAKEAWAYS

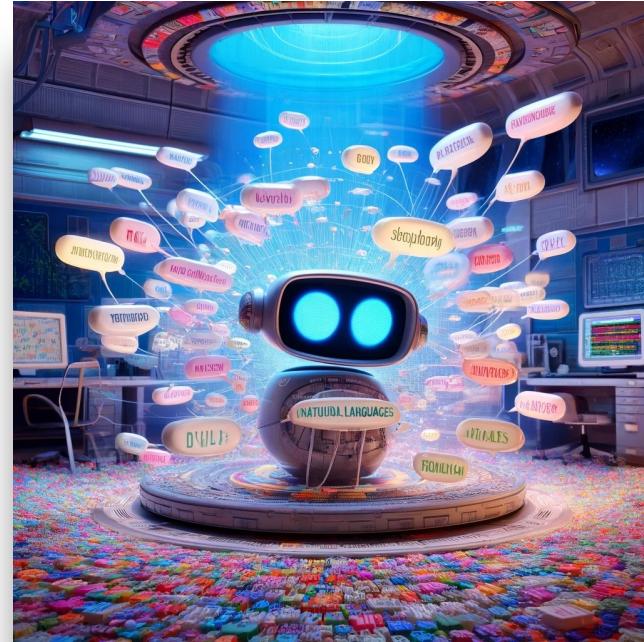
KEY TAKEAWAYS

- Define Part-of-Speech
- Explain how the following can be performed:
 - Rule-based part-of-speech tagging
 - HMM part-of-speech tagging
- Define morpheme and stem
- Identify different types of affixes in a word
- Explain the difference between inflectional and derivational morphology
- Understand the difference between stemming and lemmatization

SUGGESTED READINGS

READINGS

- Sebenta:
 - Morphology? What is it? Is there a cure?
- Jurafsky:
 - Chapter 8 (Sequence Labelling for ...):
 - 8.1-8.4



DENSE VECTORS

Luís Coheur

OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embeddings (general concept)
 - “Classic” dimensionality reduction methods
 - Neural Word Embeddings
 - Neural Sentence Embeddings
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to explain:
 - The general concept of dimensionality reduction
 - How the SVD method can be applied to create dense vectors
 - How to create neural word embeddings, namely: skip-gram and BERT
 - How we can create sentence embeddings
 - How to evaluate neural word embeddings
 - Several concepts related to dense vectors

TOPICS

THE ROAD SO FAR

Before

Sparse vectors representation
of language

Now

Dense vectors representation
of language

THE ROAD SO FAR

- REMEMBER:
 - Distributional hypothesis: “you shall know a word by the company it keeps” (Firth 1957)
 - Distributional semantics: meaning can be represented by numerical vectors rather than symbolic structures

THE ROAD SO FAR

- Up to now: very high-dimensional space
 - Words represented as sparse vectors
 - Long vectors: length = $|V|$, 10.000-50.000
 - Sparse vectors: most entries = 0
 - => need for dimensionality reduction
- Now we will see that:
 - words can be represented as dense vectors
 - Short vectors: 50-1.000 dimensions
 - Dense vectors: most entries $\neq 0$

OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embedding (general concept)
 - “Classic” dimensionality reduction methods
 - Neural Word Embeddings
 - Neural Sentence Embeddings
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

WORD EMBEDDING (GENERAL CONCEPT)

- A word embedding is a representation of a word in a vector space
 - words with similar meanings are (hopefully) represented by vectors that are close to each other in the vector space

cat →

0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
-----	-----	-----	-----	------	------	------

kitten →

0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
-----	-----	------	-----	------	------	------

dog →

0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
-----	------	-----	-----	------	------	------

houses →

-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8
------	------	------	-----	------	-----	-----

man →

0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
-----	------	-----	-----	------	------	------

woman →

0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
-----	-----	-----	------	-----	------	------

king →

0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
-----	------	-----	-----	-----	------	------

queen →

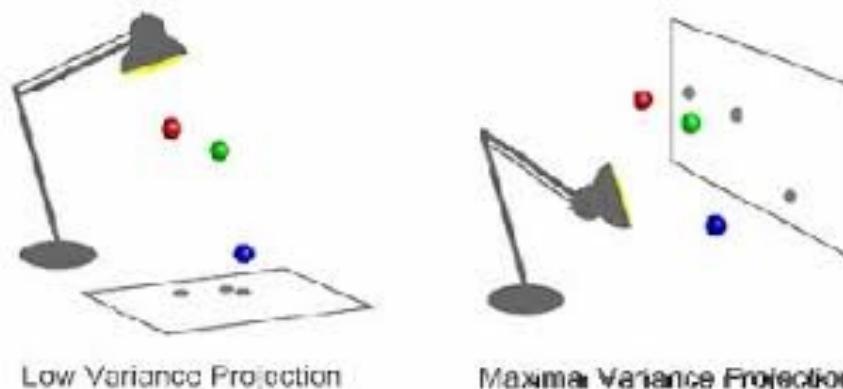
0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9
-----	------	-----	------	-----	------	------

OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embedding (general concept)
 - “Classic” dimensionality reduction methods
 - Neural Word Embeddings
 - Neural Sentence Embeddings
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

DIMENSIONALITY REDUCTION METHODS

- Goal: reduce the number of dimensions (variables) in a data set without significant loss of information.



SINGULAR VALUE DECOMPOSITION

- Singular Value Decomposition (SVD): method (Algebra) for finding the most important dimensions of a data set (those dimensions along which the data varies the most)

$$A = U D V^T$$

Diagram illustrating the Singular Value Decomposition (SVD) of a matrix A into three components:

- A is an $m \times n$ matrix.
- $=$ is followed by three matrices:
 - U is an $m \times r$ matrix.
 - D is a diagonal matrix of size $r \times r$.
 - V^T is an $r \times n$ matrix.

Annotations for the SVD formula:

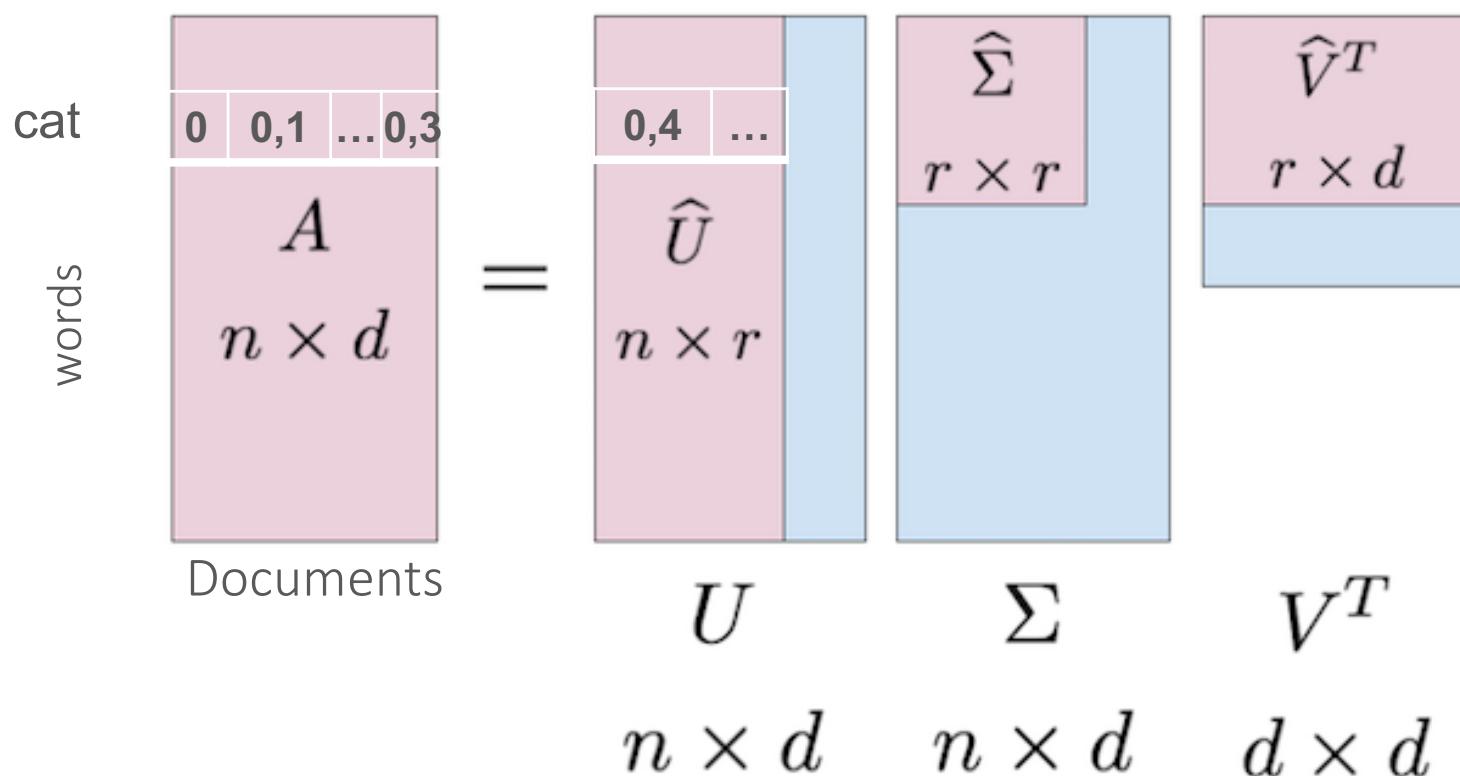
- U is labeled "Left singular vectors".
- D is labeled "Singular values".
- V^T is labeled "Right singular vectors".

<https://adel.ac/singular-value-decomposition-in-computer-vision/>

Applied to NLP since
1988

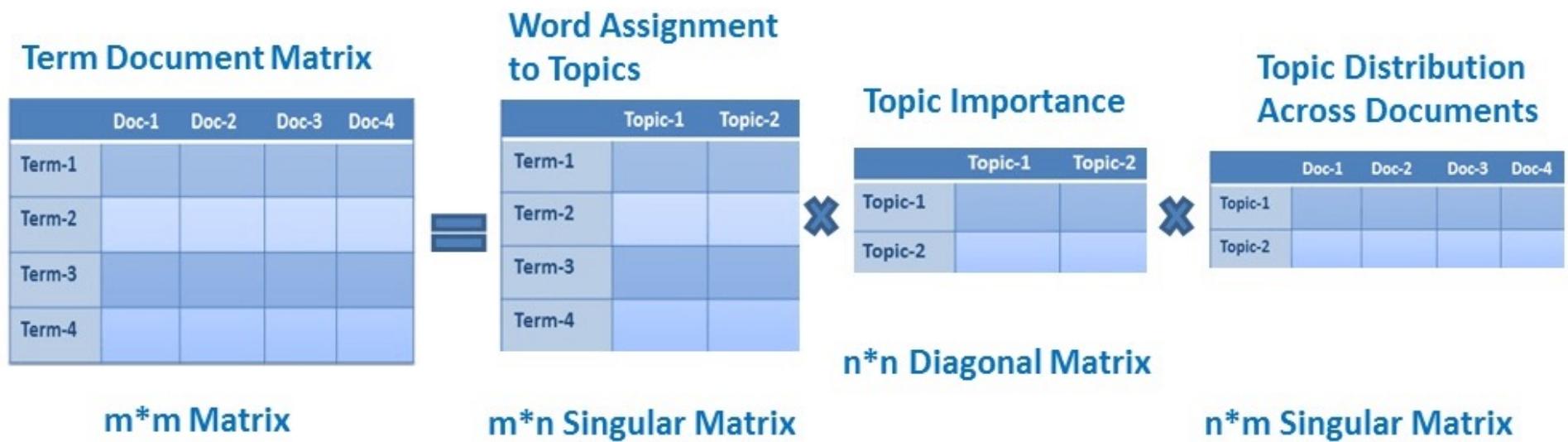
LATENT SEMANTIC ANALYSIS (OR INDEXING)

- Latent Semantic Analysis (or Indexing) (Landauer & Dumais, 1997): reduces the high-dimensional vector space by collapsing the original representation into a smaller (size r) set of latent dimensions. This is done based on the Singular Value Decomposition (SVD) method.



LATENT SEMANTIC ANALYSIS (OR INDEXING)

- Idea (suppose we only want to keep two topics)
 - Intuition: emergence of latent topics



<https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python>

LATENT SEMANTIC ANALYSIS (OR INDEXING)

Latent Semantic Indexing (LSI) An Example

(taken from Grossman and Frieder's *Information Retrieval, Algorithms and Heuristics*)

A “collection” consists of the following “documents”:

- d1: *Shipment of gold damaged in a fire.*
- d2: *Delivery of silver arrived in a silver truck.*
- d3: *Shipment of gold arrived in a truck.*

Suppose that we use the term frequency as term weights and query weights. The following document indexing rules are also used:

- stop words were not ignored
- text was tokenized and lowercased
- no stemming was used
- terms were sorted alphabetically

LATENT SEMANTIC ANALYSIS (OR INDEXING)

Problem: Use Latent Semantic Indexing (LSI) to rank these documents for the query *gold silver truck*.

Step 1: Set term weights and construct the term-document matrix **A** and query matrix:

Terms	d1	d2	d3
a	↓	↓	↓
arrived	1	1	1
damaged	0	1	1
delivery	1	0	0
fire	0	1	0
gold	1	0	0
in	1	0	1
of	1	1	1
shipment	1	1	1
silver	1	0	1
truck	0	2	0
	0	1	1

LATENT SEMANTIC ANALYSIS (OR INDEXING)

Step 2: Decompose matrix **A** matrix and find the **U**, **S** and **V** matrices, where

$$\mathbf{A} = \mathbf{USV}^T$$

$$\mathbf{U} = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix}$$

U contains the Word
Embeddings! (dim = 2)

$$\mathbf{S} = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix}$$

$$\mathbf{V}^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

LATENT SEMANTIC ANALYSIS (OR INDEXING)

Latent Semantic Analysis — Deduce the **hidden topic** from the document

$$A = \begin{bmatrix} d1 & d2 & d3 \\ \downarrow & \downarrow & \downarrow \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ \boxed{0} & \boxed{1} & \boxed{0} \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$
$$U \approx U_k = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ \boxed{-0.1576} & \boxed{-0.3046} \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix}$$

LATENT SEMANTIC ANALYSIS (OR INDEXING)

- LATENT SEMANTIC ANALYSIS
 - If applied to word-to-word matrix:
 - k = most important singular values (top k dimensions)
 - Truncated SVD

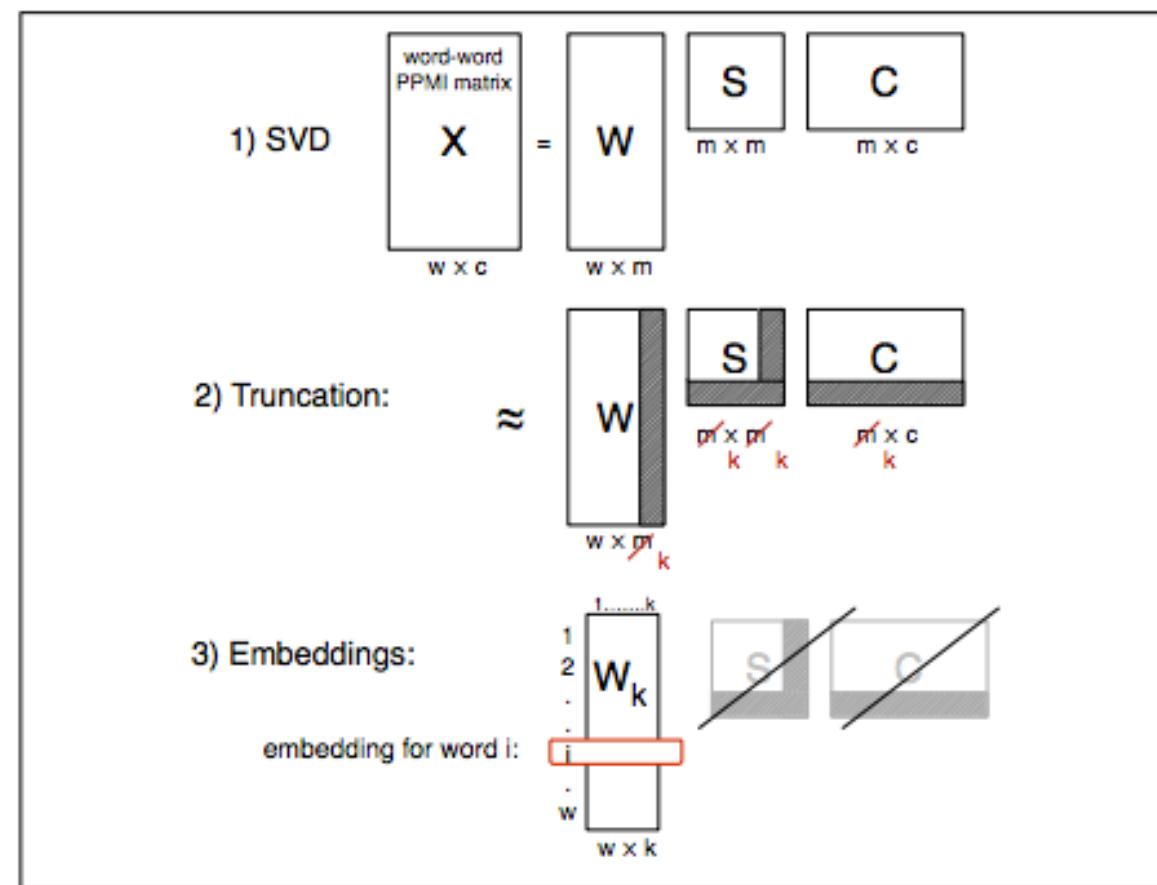


Figure from Jurafsky

RELATED TOPICS

- Topic models: type of statistical models for discovering the abstract "topics" that occur in a collection of documents.
- You have probably studied a dimensionality reduction technique called Principal Component Analysis (PCA)
 - Latent Semantic Analysis (LSA) is essentially truncated SVD, which is mathematically related to PCA.

OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embedding (general concept)
 - “Classic” dimensionality reduction methods
 - [Neural Word Embeddings](#)
 - Neural Sentence Embeddings
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

DENSE VECTORS BASED ON NEURAL NETWORKS

- Neural Word Embeddings:
 - F_θ : words $\rightarrow \mathbb{R}^n$
 - θ is randomly initialized (thus, random vectors for each word)
 - We learn (Deep Learning) to have meaningful vectors
 - Several methods to do this



HYPE

WORD2VEC

- WORD2VEC (Mikolov et al. 2013 and 2013a): group of models that are used to produce word embeddings.
 - Skip-gram model: uses the current word to predict the surrounding window of context words.
 - CBOW model: predicts the current word from a window of surrounding context words.

TEASER

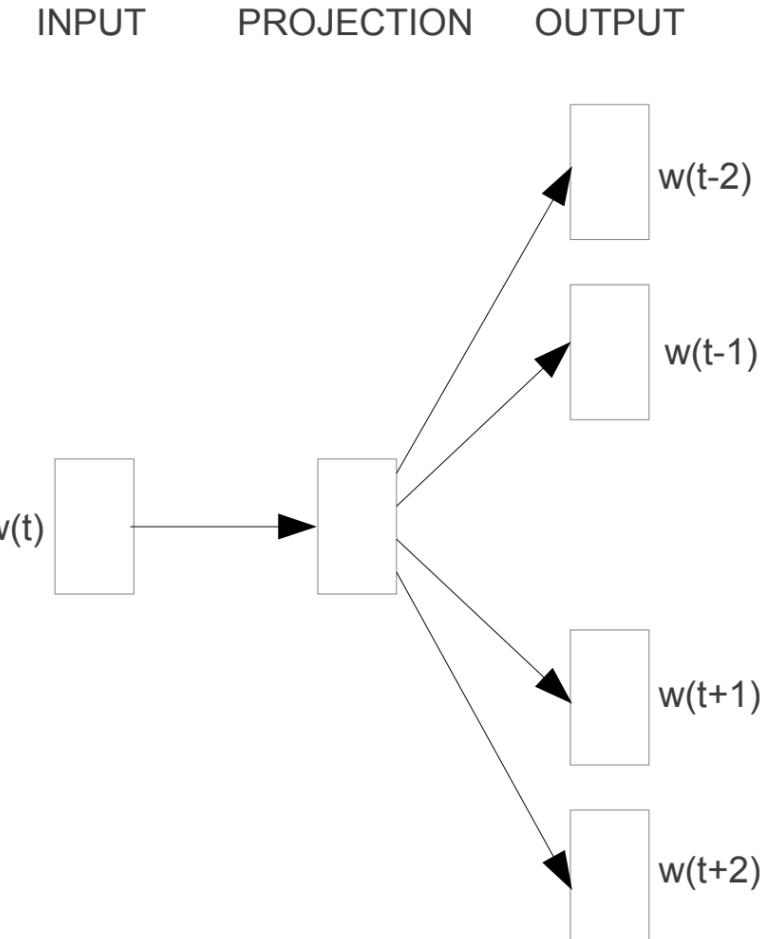
- Skip-gram and CBOW model perform “Fake” tasks



SKIP-GRAM

- Skip-gram:

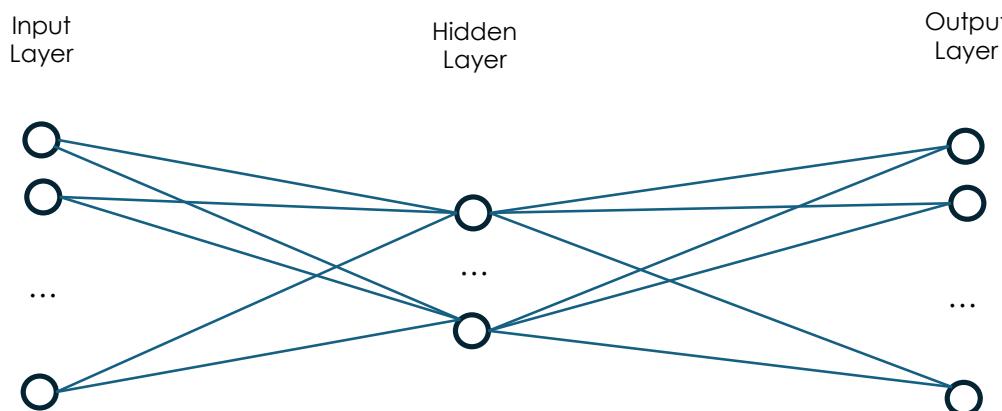
- Task intuition: given the word “Soviet”, the output probabilities are going to be higher for words like “Union” and “Russia” than for “table” or “watermelon”.



Skip-gram

SKIP-GRAM

- Vocabulary size = $|V|$ (unique words)
- Training:
 - 1-hot input vector with size $|V| \times 1$
 - 1 hidden layer
 - 1-hot output vector with size $|V| \times 1$



Dictionary D= { I; cat; dog; have; a }

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

One hot vector

Also, consider the dimension of each dense vector (ex: N = 300)

SKIP-GRAM

- Skip-gram: unsupervised (building the training data)

My cat has powers

Training samples
(window = 1)

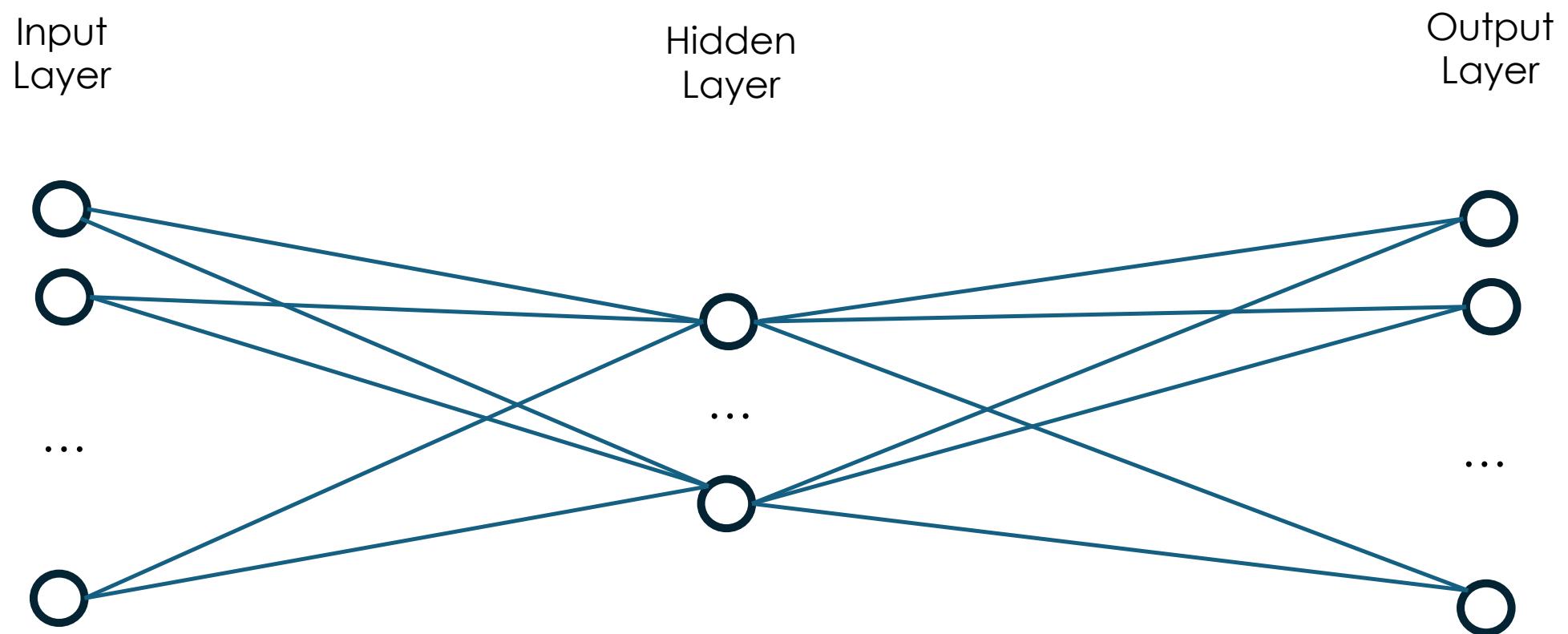
(my, cat)
(cat, my)
(cat, has)
(has, cat)
(has, powers)
(powers, has)

Training samples
(window = 2)

(my, cat)
(my, has)
(cat, my)
(cat, has)
(cat, powers)
(has, my)
(has, cat)
(has, powers)
(powers, cat)
(powers, has)

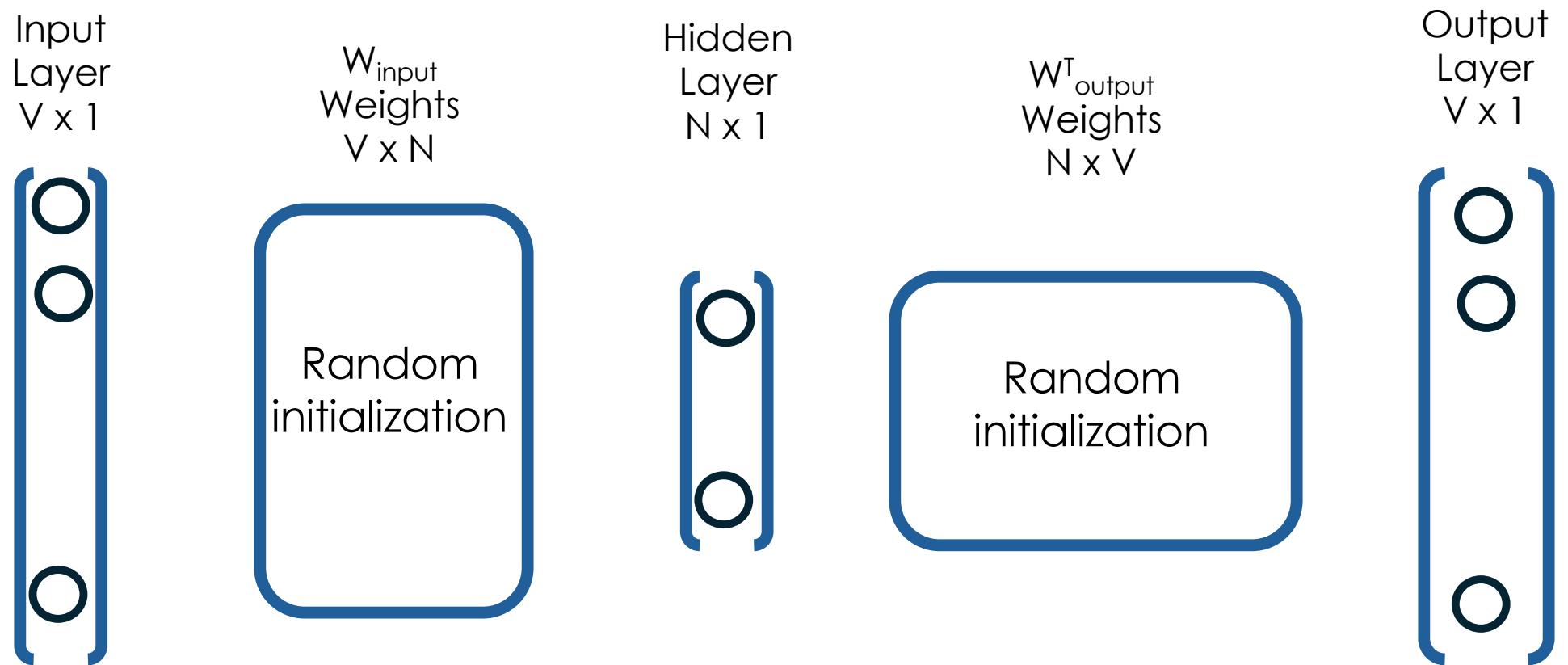
SKIP-GRAM

- Skip Gram model architecture



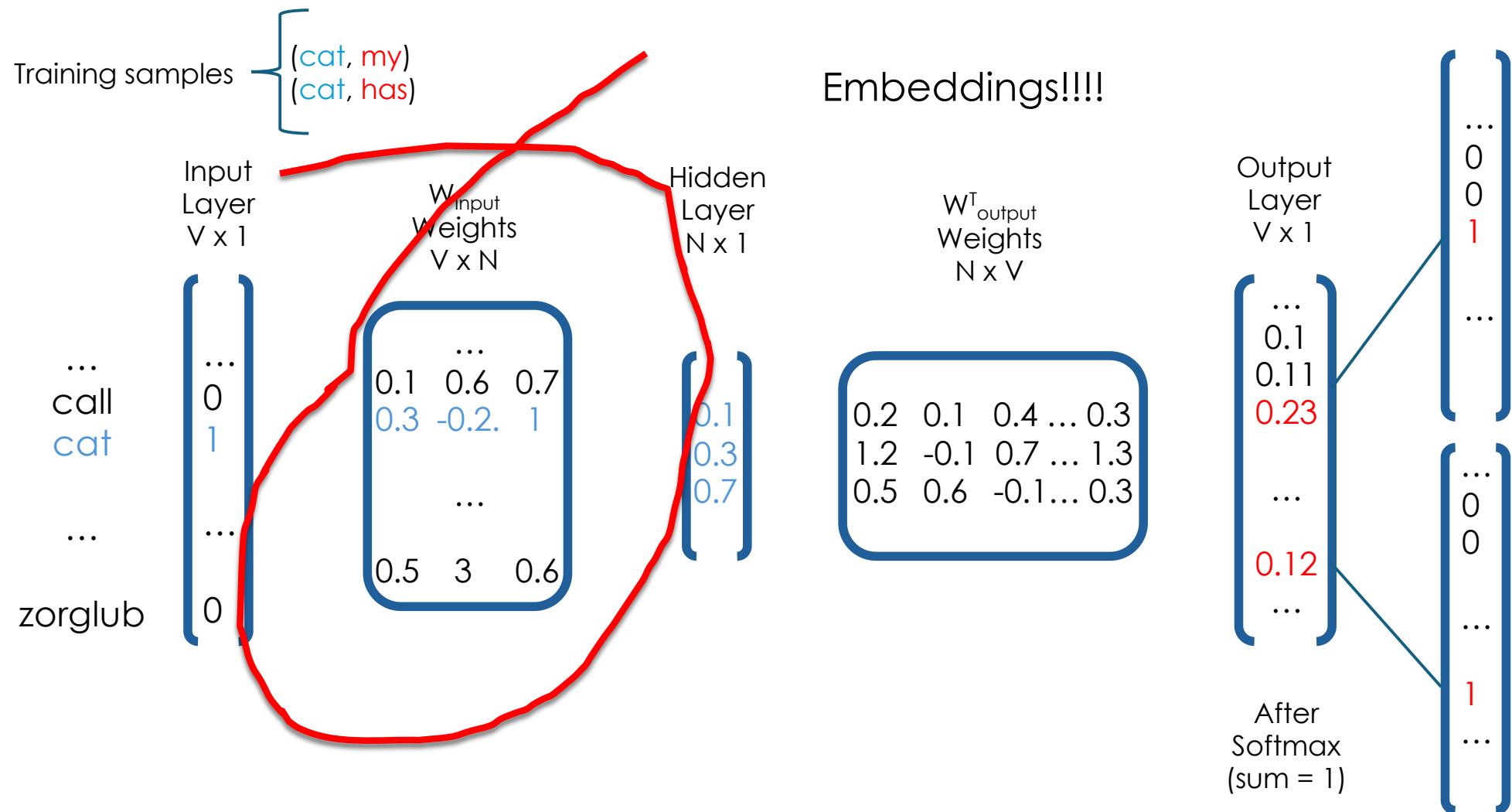
SKIP-GRAM

- Skip Gram model architecture



SKIP-GRAM

- Skip Gram model training



SKIP-GRAM

- As each word is represented by a one-hot vector, multiply that vector by the (lookup) matrix and you have the vector that corresponds to that word embedding

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{1} & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \hline 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

From Chris McCormick

SKIP-GRAM

- Now you understand why Skip-gram and CBOW model perform “fake” tasks



ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

- Write a short comment justifying the veracity/falsity of the sentence:

*“Embeddings created with word2vec
are the result of a fake task”*

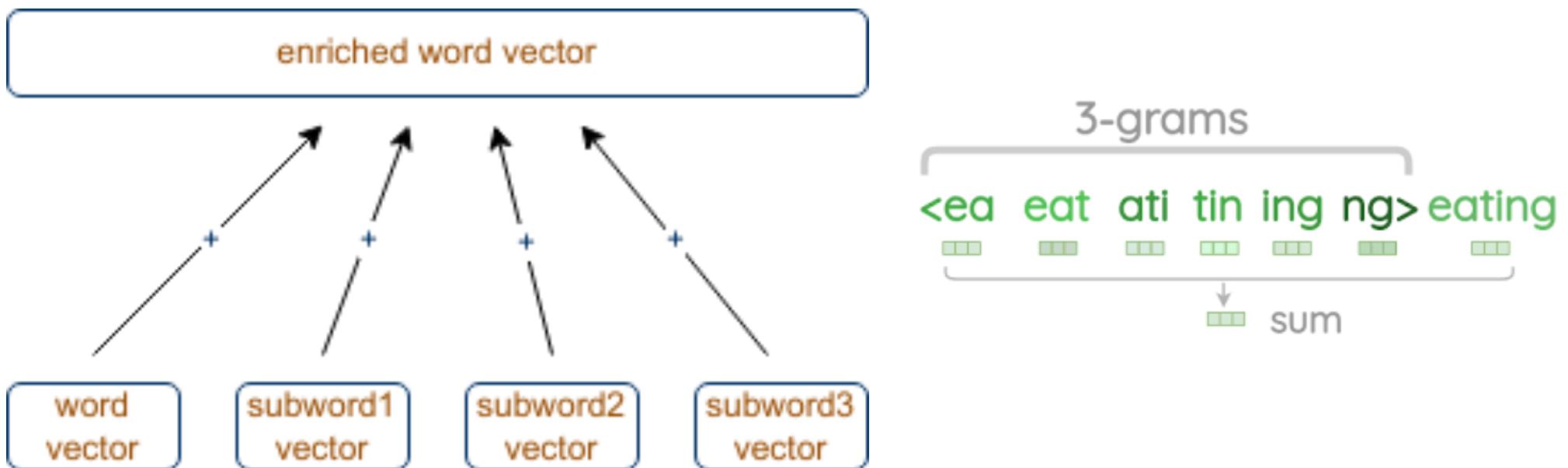
After training (word2vec), the goal is not to apply the created model, but to extract the weights that connect the input and the hidden layers. These weights are going to be the Word Embeddings.

GloVe

- After Word2Vec there were many models to create embeddings.
 - You probably heard about Global Vectors (GloVe) (Stanford, Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014)

fastText

- fastText (Facebook, P. Bojanowski, E. Grave, A. Joulin. 2016) was specifically interesting as it used the concept of subword:
 - a powerful way to handle misspelling words and unknown words.



CONTEXT-FREE MODELS

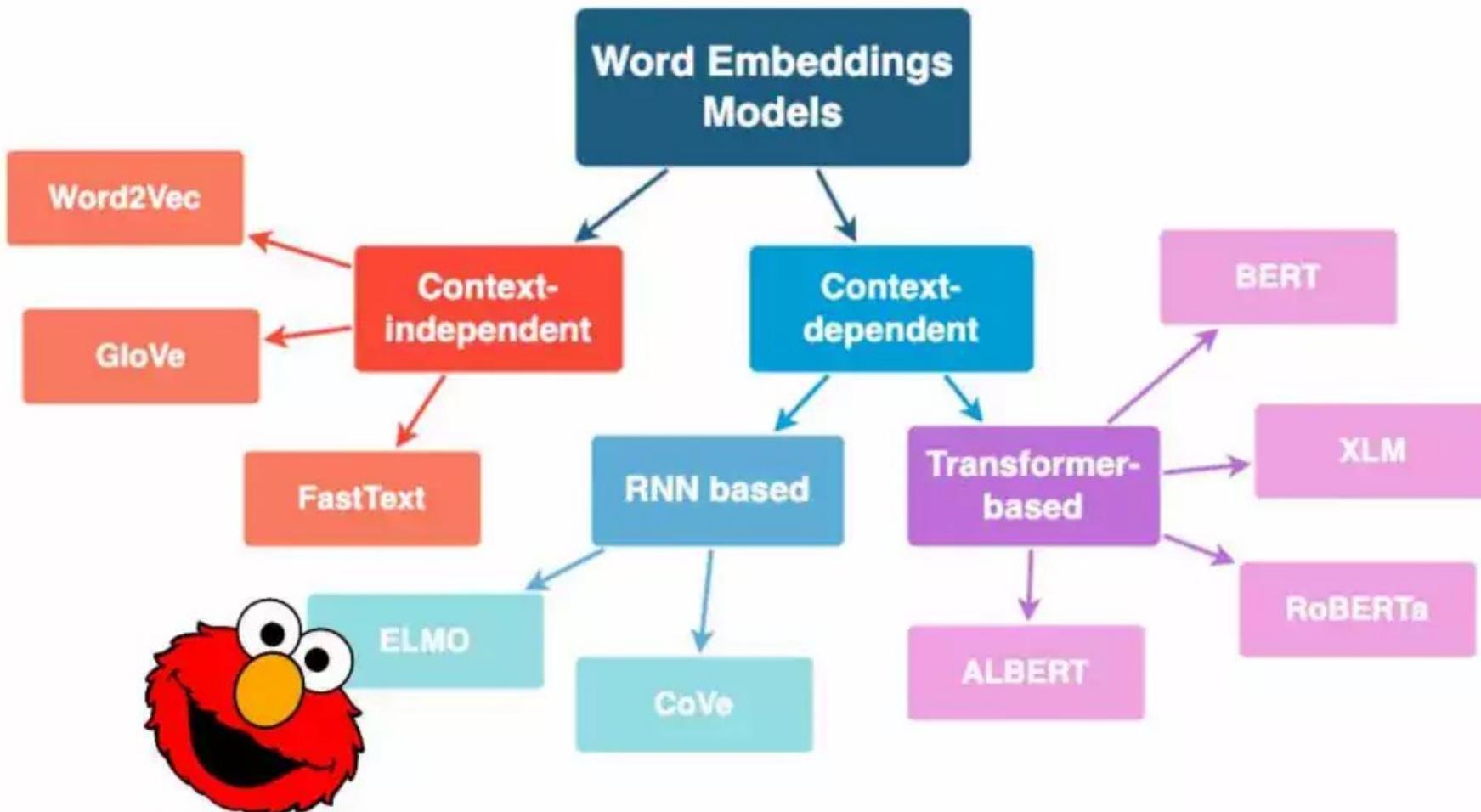
- However, the first models were context-free models: they generated a single word embedding representation for each word in the vocabulary.
 - So, bank has the same representation in bank deposit and river bank

CONTEXTUAL MODELS

- Then, the contextual models arrived: these generate a representation of each word based on the other words in the sentence
 - Examples: ELMo, ULMFit, and BERT

DENSE VECTORS BASED ON NEURAL NETWORKS

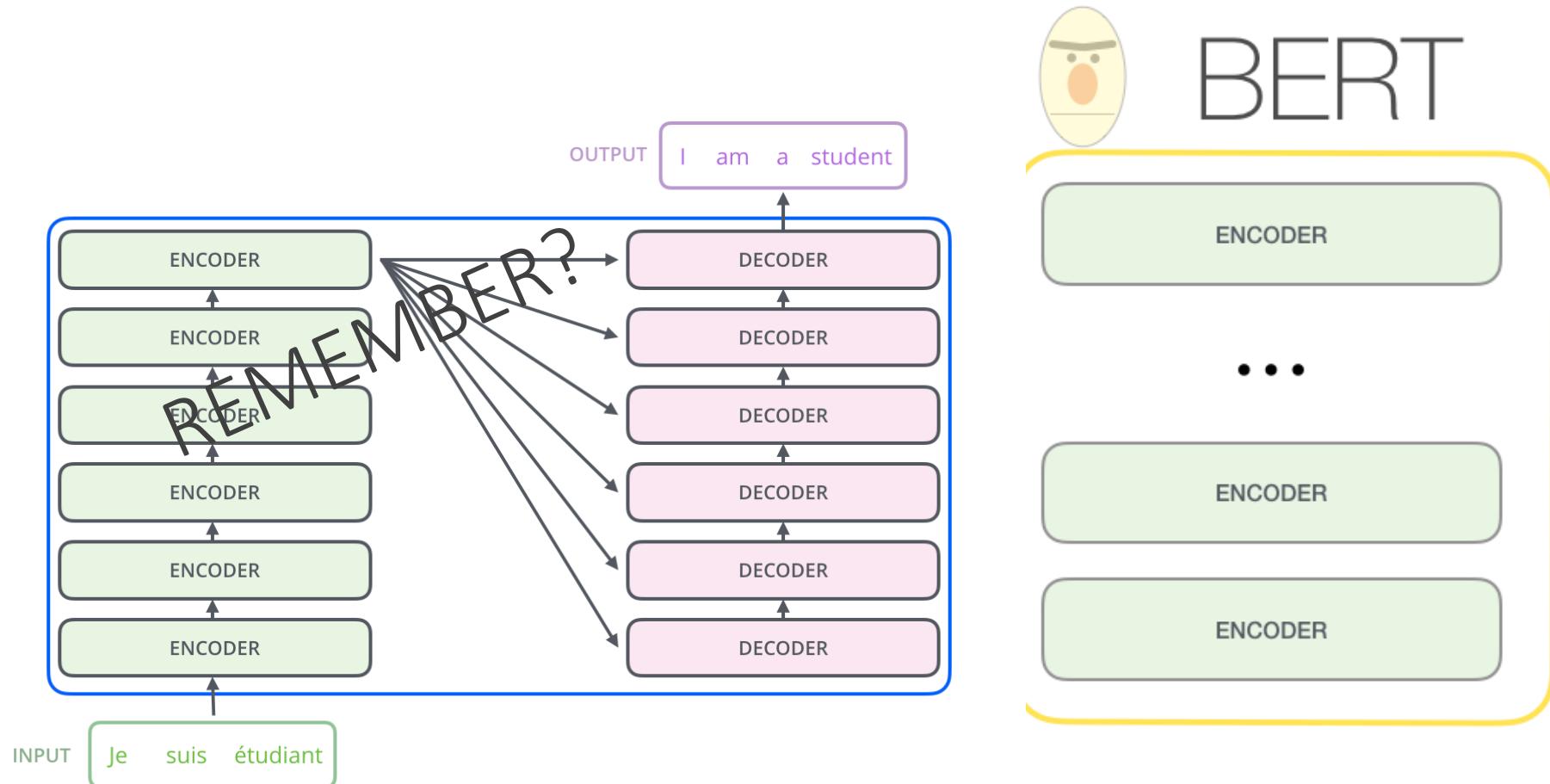
- Before we move to BERT (and just to mention a few)





BERT

- BERT (Google, late 2018)



BERT TASK 1

- BERT is trained in two tasks:
 - TASK 1 (Masked Language Model): BERT masks out 15% of the words in the input, run the entire sequence through a bidirectional Transformer encoder, and then predict only the masked words.
- Example:
 - Input: the man went to the [MASK1] . he bought a [MASK2] of milk. Labels: [MASK1] = store; [MASK2] = gallon

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



THINK-PAIR-SHARE

- What do you think bidirectional means in this context?



BERT TASK 1

- BERT Training (TASK 1 – Masked Language Model): for the 15% of selected input tokens we want to mask, what they do:

Original Sentence

BERT can see all the words in this sentence

- 1 With MASK token (80%)** BERT can see all the [MASK] in this sentence
- 2 With Incorrect word (10%)** BERT can see all the touchdown in this sentence
- 3 With Correct word (10%)** BERT can see all the words in this sentence

BERT TASK 2

- BERT Training (Task 2 – Next Sentence Prediction): given two sentences A and B, find out if B is the actual next sentence that comes after A, or just a random sentence from the corpus.

- Example:

Sentence A: the man went to the store .

Sentence B: he bought a gallon of milk .

Label: IsNextSentence

Sentence A: the man went to the store .

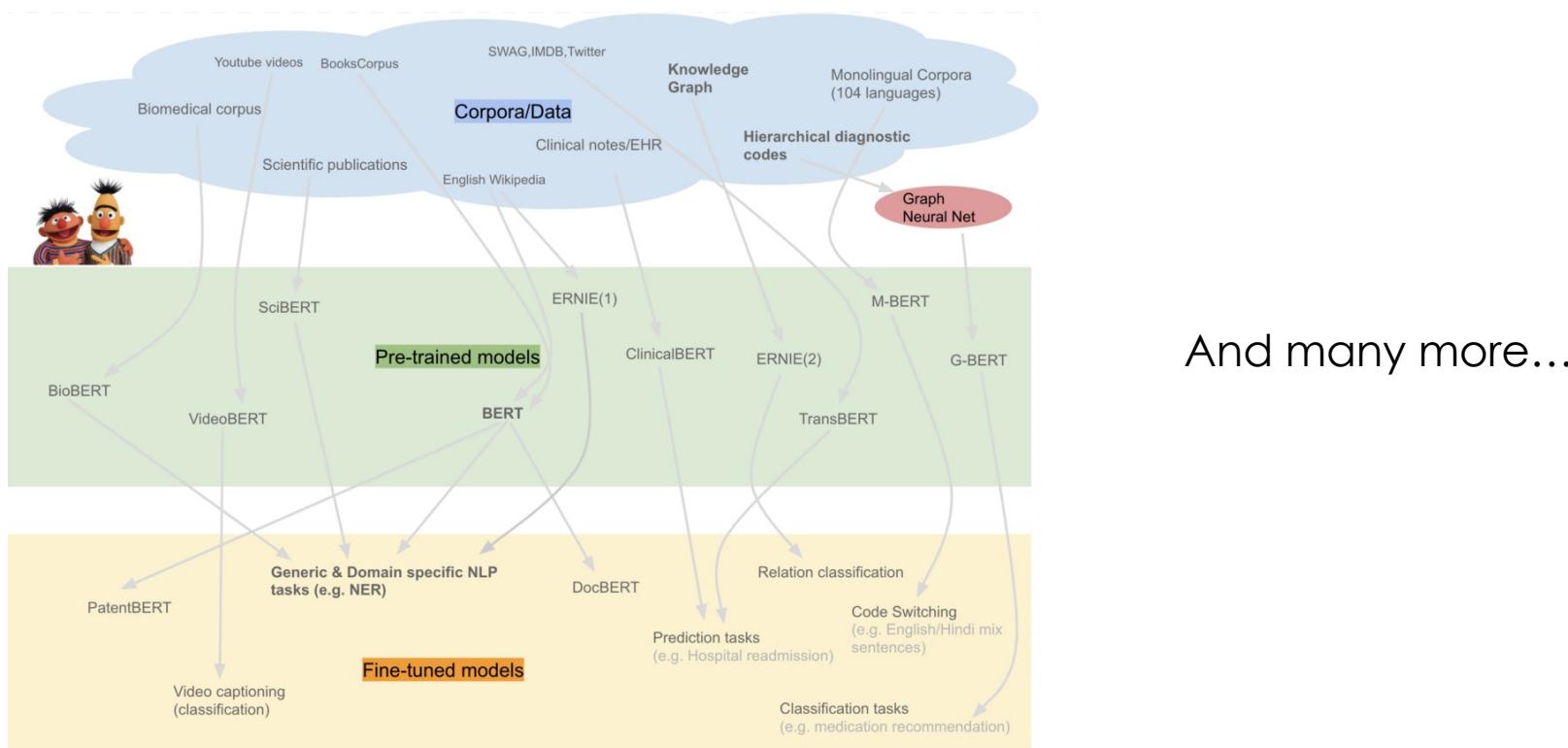
Sentence B: penguins are flightless .

Label: NotNextSentence

MULTI-TASK LEARNING IS ALSO A CURRENT TREND!!

DENSE VECTORS BASED IN NEURAL NETWORKS

- BERT train:
 - A large model is trained (12-layer to 24-layer Transformer) on a large corpus (Wikipedia + [BookCorpus](#)) for a long time.
 - There are several different versions of BERT →



DENSE VECTORS BASED IN NEURAL NETWORKS

- Using BERT:
 - BERT has two stages: Pre-training and fine-tuning
 - Pre-training is fairly expensive (four days on 4 to 16 Cloud TPUs), but is a one-time procedure
 - Fine-tuning is inexpensive
- We will talk about this a future class

OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embedding (general concept)
 - “Classic” dimensionality reduction methods
 - Neural Word Embeddings
 - [Neural Sentence Embeddings](#)
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

SENTENCE EMBEDDINGS

- And... what about sentence embeddings?
 - Sentence embedding techniques represent entire sentences and their semantic information as vectors

SENTENCE EMBEDDINGS

- There are also models that are trained to create sentence embeddings:
 - Doc2Vec (2014): adds on to the Word2Vec
 - SentenceBERT (2019, Nils Reimers, Iryna Gurevych)
 - InferSent (facebook)
 - Universal Sentence Encoder (Google): multitask learning

SENTENCE EMBEDDINGS

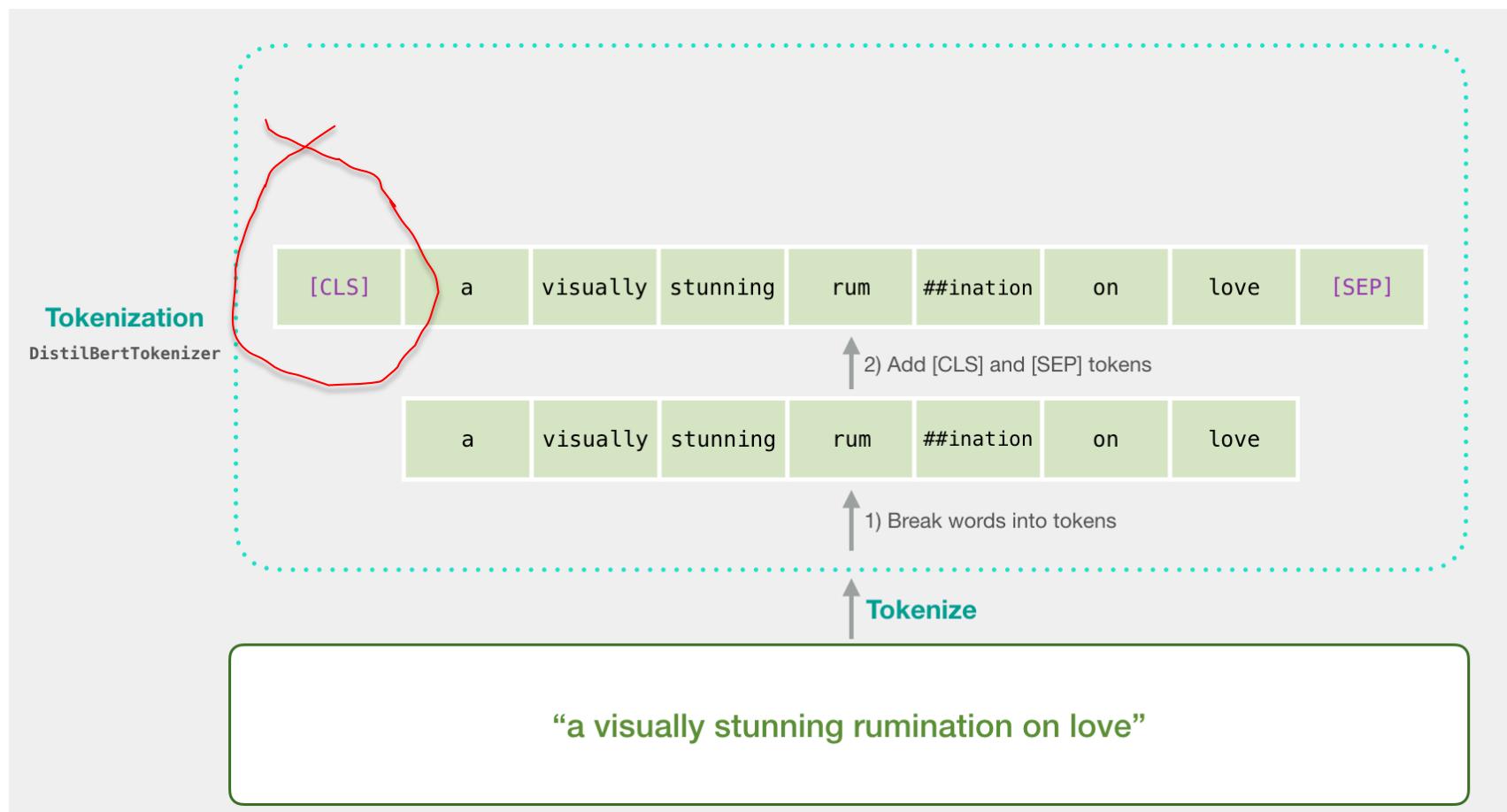
- But... the most straightforward sentence embeddings can be obtained:
 - by summing the word embeddings of the words in the sentence;
- or
- by taking an average of the word embeddings of the words in the sentence

You must be joking...



SENTENCE EMBEDDINGS

- Or use the CLS token (we will talk about this in a future class)

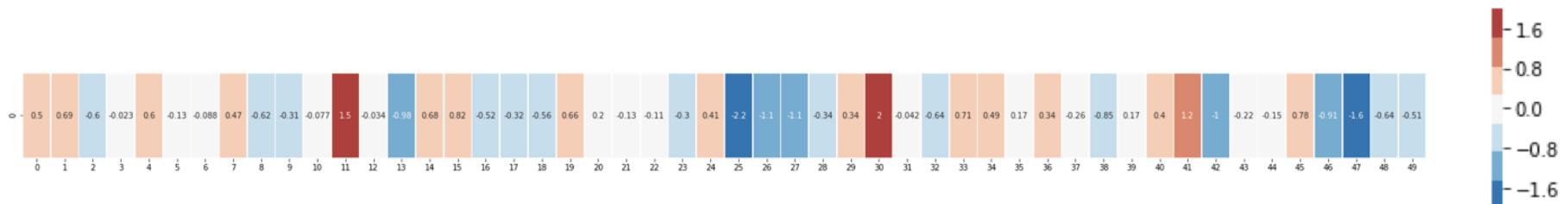


OVERVIEW

- Learning objectives
- Topics
 - The road so far
 - Word Embedding (general concept)
 - “Classic” dimensionality reduction methods
 - Neural Word Embeddings
 - Neural Sentence Embeddings
 - Representing and Evaluating Word Embeddings
- Key takeaways
- Suggested readings

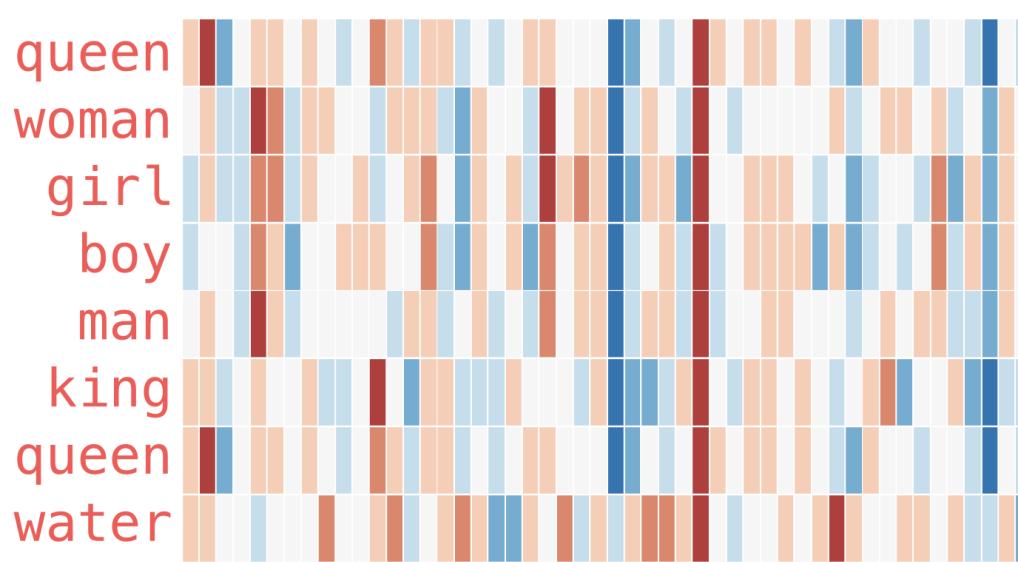
REPRESENTING WORD EMBEDDINGS

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -  
0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 ,  
-0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -  
1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 ,  
0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -  
0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

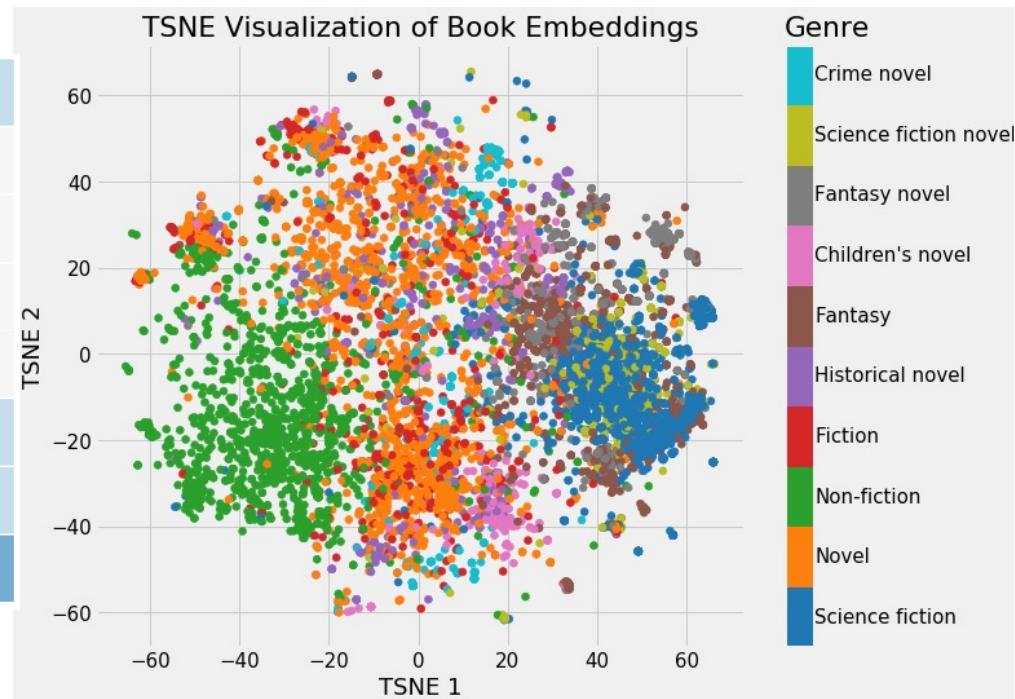


REPRESENTING WORD EMBEDDINGS

- <https://devopedia.org/word-embedding>

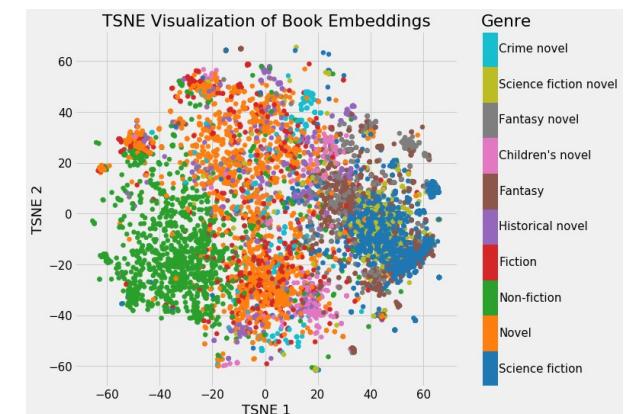


From <http://jalammar.github.io/illustrated-word2vec/>



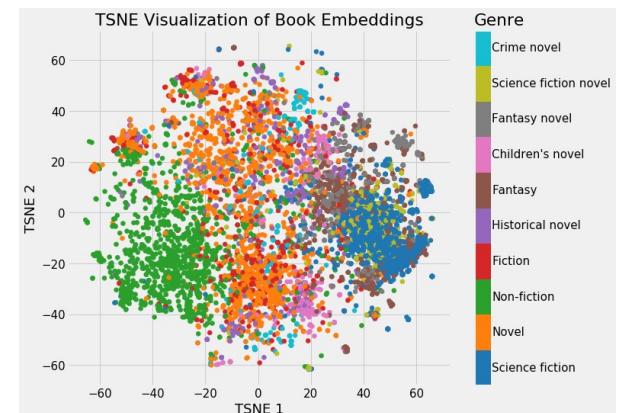
BY THE WAY...

- t-SNE is a dimensionality reduction technique designed to visualize high-dimensional data (like word embeddings) in 2D or 3D.
 - Input: High-dimensional vectors (e.g., 512-dimensional embeddings)
 - Output: 2D or 3D points that you can plot
 - Goal: Preserve the local structure of the data
 - points that are close in high dimensions remain close in the 2D/3D visualization



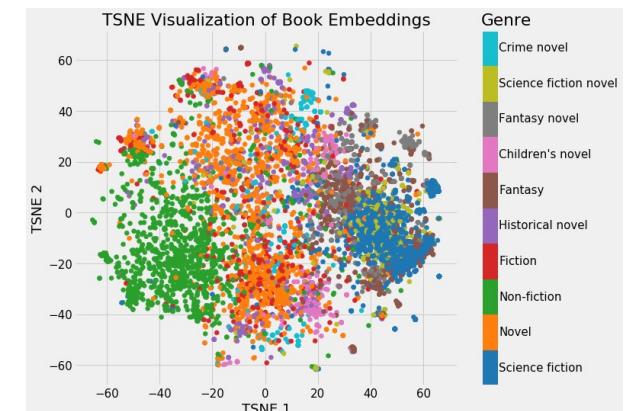
BY THE WAY...

- How t-SNE works (intuitions):
 - Measure distances between points in high dimensions
 - Convert distances to probabilities representing similarity
 - closer points → higher probability
 - Start with a random 2D layout
 - Adjust positions iteratively, so that the 2D probabilities match the high-dimensional probabilities as closely as possible
 - End result: clusters of similar points appear together, separated from dissimilar points.



BY THE WAY...

- And this takes to another concept, also named **Perplexity** (remember?):
 - Perplexity is a hyperparameter in t-SNE: it tell us how many neighbours a point cares about (each point “attends” to roughly perplexity neighbours when computing similarity)
 - Low perplexity → the algorithm focuses on very local structure (few neighbours).
 - High perplexity → it considers broader neighbourhoods (more neighbours).



EVALUATING WORD EMBEDDINGS

- And... how do we know if we have created good embeddings?
 - Use them in extrinsic evaluations (remember?)
 - QA, Spell Checking, ...
 - Evaluate them in intrinsic evaluations
 - Correlation between algorithm and human word similarity ratings
 - Example: $\text{sim}(\text{plane}, \text{car}) = 5,77$
 - TOEFL multiple-choice vocabulary tests
 - Example: “Levied” is closest in meaning to: Imposed, believed, requested, correlated

EVALUATING WORD EMBEDDINGS

- Language
 - $\Theta_{\text{france}} - \Theta_{\text{french}} \approx \Theta_{\text{mexico}} - \Theta_{\text{spanish}}$
- Gender
 - $\Theta_{\text{king}} - \Theta_{\text{man}} \approx \Theta_{\text{queen}} - \Theta_{\text{woman}}$
 - $\Rightarrow \text{King} - \text{Queen} + \text{Woman} = \text{Man}$
- Plural
 - $\Theta_{\text{car}} - \Theta_{\text{cars}} \approx \Theta_{\text{apple}} - \Theta_{\text{apples}}$

EVALUATING WORD EMBEDDINGS

- How do we know that our embeddings are not biased?
- From Jurafsky: “For example African-American names like ‘Leroy’ and ‘Shaniqua’ had a higher GloVe cosine with unpleasant words while European-American names ('Brad', 'Greg', 'Courtney') had a higher cosine with pleasant words.”

ACTIVE LEARNING MOMENT



EXERCISE

- With your colleagues:
 - say how related the words in the table are (score them between -1 and 1).
 - say which is the word closest to: water, rain, person, pencil, tree, flower
 - complete the analogies: woman/girl -> man/? , woman/aunt -> man/? , uncle/boy -> aunt/? , France/Paris -> Portugal/? , water/fire -> sea/?
- Test everything in
 - http://epsilon-it.utu.fi/wv_demo/
 - Think a little bit about this

Word 1	Word 2
tiger	cat
tiger	dog
book	paper
computer	keyboar d
computer	internet
plane	car
train	car



KEY TAKEAWAYS

KEY TAKEAWAYS

- The idea of creating/using dense vectors is not new
 - SVDs can be used for dimensionality reduction
 - You should be able to explain this
- There are many neural word embeddings: everything started with word2vec
- Skip-gram is based on a neural network with a very simple architecture
- You should be able to explain:
 - Skip-gram training
 - Why Word2vec is a context-free model
 - Why BERT is a contextual model
 - BERT training (in two tasks: a masked language model and a next sentence prediction)

KEY TAKEWAYS (CONT.)

- There are several ways to obtain the embeddings of a sentence – you should be able to explain some
- There are several ways to evaluate embeddings – you should be able to describe some ways
- Be careful with bias in embeddings

SUGGESTED READINGS

READINGS

- Jurafsky: 6.8, 6.9, 6.10, 6.11, 11.1, 11.2



SYNTAX

Luísa Coheur

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Syntax
 - Grasp fundamental concepts and learn how to perform a Syntactic Analysis

TOPICS

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

MAIN CONCEPTS

- Natural Language syntax restricts the sequences of words that are part of the language, but is much more flexible than the syntax of artificial languages
- Some used tags:
 - Noun Phrases (NP)
 - Verb Phrases (VP)
 - Prepositional Phrases (PP)
 - ...

MAIN CONCEPTS

- The used tags can be more functional:
 - Subject:
 - Example:
 - [The student]_{SUBJ} took the test.
 - Direct Object/Complement:
 - Example:
 - The student is reading [the book]_{DO}.
 - Indirect Object/Complement:
 - Example:
 - Give [the book]_{DO} [to Mary]_{IO}.
 - Predicative of the Subject:
 - Examples:
 - The teacher is [tired]_{PS}.
 - Maria is [a teacher]_{PS}.

EXAMPLE

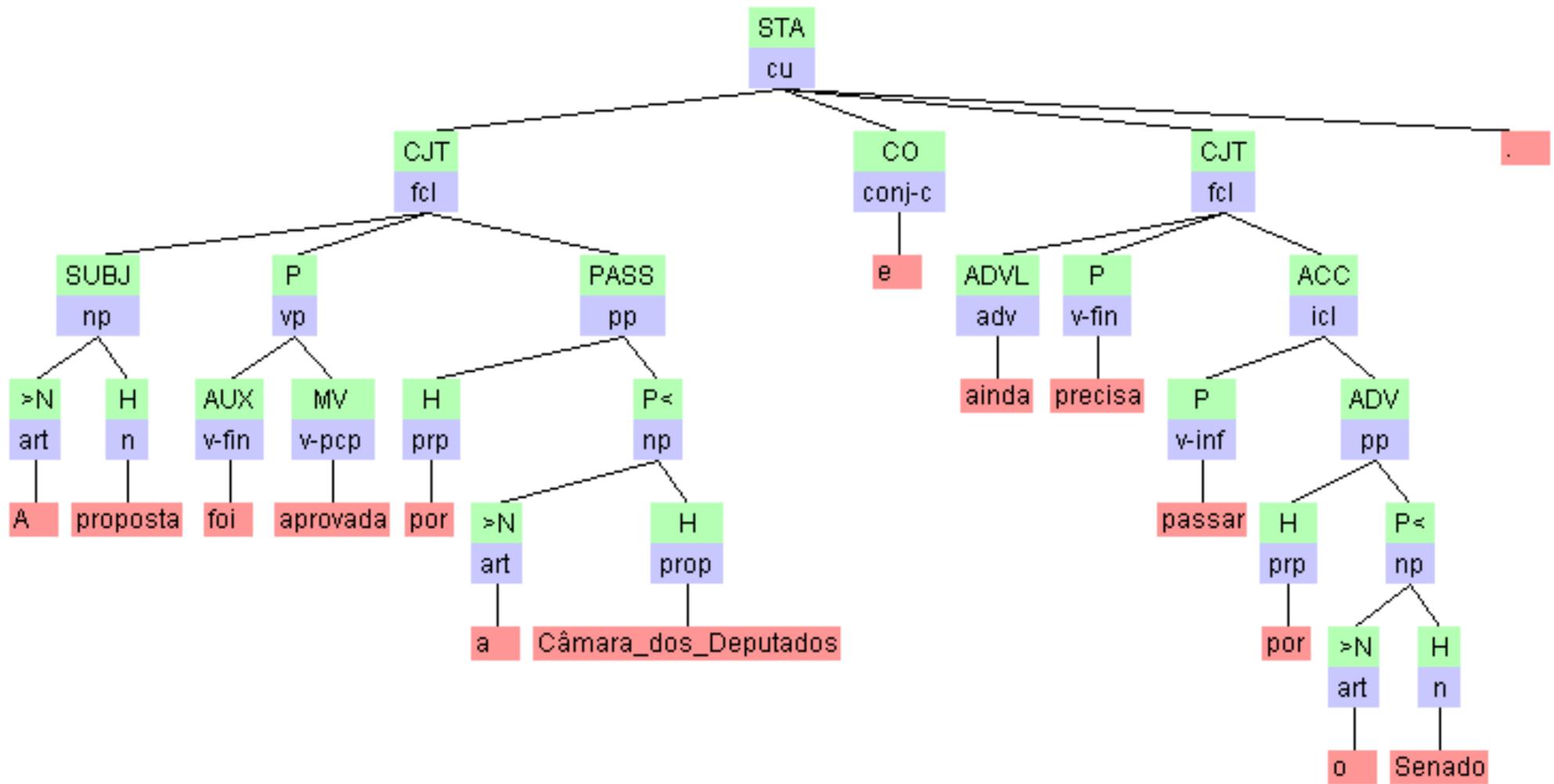
Wow, Sarah carefully hands her friend the red book from the shelf, and smiles.

- Interjection (INTJ):
- Subject (NP):
- Verb Phrase (VP):
 - Adverb (ADV):
 - Verb (V):
 - Indirect Object (NP):
 - Direct Object (NP):
 - ...
 - Prepositional Phrase (PP):
- Coordinate Clause (CC):

TREEBANKS

- Treebank:
 - A corpus where each sentence is syntactically annotated
 - Examples:
 - Penn Treebank, Prague Dependency Treebank (Czech), Negra Treebank (German), Susanne (English), Floresta Sintáctica (Linguateca) for Portuguese, ...

EXAMPLE: FLORESTA SINTÁCTICA



SYNTACTIC PARSING

- We call **syntactic analysis/parsing**¹ to the process of obtaining a **syntactic tree/structure** from an input sequence.

¹ We will use these terms interchangeably

SYNTACTIC PARSING

- There are many algorithms to perform syntactic parsing, considering the grammar in use
 - We will study several grammar formalisms (next)
 - There are some approaches that perform syntactic analysis with deep learning methods

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

CONTEXT-FREE GRAMMARS

- Groups of words can behave like single units or sentences; these groups are called “constituents”
 - Examples:
 - Noun Phrases:
 - “the princess”, “John”, “my amazing sister”, ...
- Context-Free Grammars (CFGs):
 - Capture the constituents and their order in sentences

CONTEXT-FREE GRAMMARS

But what
exactly is a
Context-Free
Grammar?



CONTEXT-FREE GRAMMARS (CFG) FORMALISM

- A CFG is a tuple (N, T, S_0, R) in which:
 - N is a set of non-terminal symbols (or tags)
 - Example: n, art, v, NP, VP, ...
 - T is a set of terminal symbols (the language tokens)
 - Example: Maria, love, peace, house, and, ...
 - S_0 is the initial symbol ($S_0 \in N$)
 - R is a set of rules of the form $A \rightarrow a$ where:
 - $A \in N$
 - a is a string of zero or more terminal and non-terminal symbols
 - Example:
 - $NP \rightarrow \text{art } n$

EXAMPLE OF A CONTEXT-FREE GRAMMAR

- $G = (N, T, S_0, R)$,
 - $N = \{S, NP, VP, Pron, Det, Noun, Verb\}$
 - $T = \{I, They, book, João, love, a, the, that\}$
 - $S_0 = S$ (S for sentence)
 - R (note: " $A \rightarrow b \mid c$ " is the same as " $A \rightarrow b$ and $A \rightarrow c$ "):
 - $S \rightarrow NP \ VP$
 - $NP \rightarrow Pron \mid Noun \mid Det \ Noun$
 - $VP \rightarrow Verb \ NP$
 - $Pronoun \rightarrow I \mid They$
 - $Noun \rightarrow book \mid João$
 - $Verb \rightarrow love$
 - $Det \rightarrow a \mid the \mid that$

SYNTACTIC PARSING

- Derivation with CFG: sequence of rule applications in which “A” is rewritten as “a” if there is a rule in the form $A \rightarrow a$
- Language derived by a CFG G:
 - $L(G) = \{w \mid w \text{ is a string of terminal symbols and } S \text{ derives } w\}$

EXAMPLE

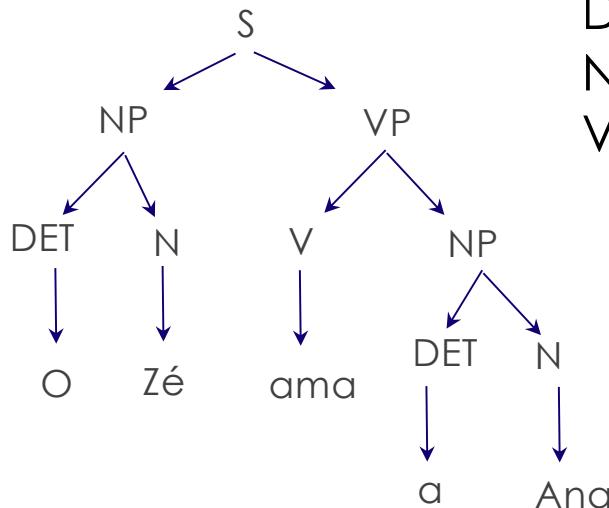
Grammar	Lexicon	POS
S -> NP VP	o, a	DET
NP -> DET N	Zé, Ana	N
VP -> V NP	ama	V

CFC G = (N, T, S₀, R)

- N = {S, NP, VP, DET, N, V}
- T = {o, a, Zé, Ana, ama}
- S₀ = S
- R = {S -> NP VP,
NP -> DET N,
VP -> V NP,
DET -> o | a,
N -> Zé | Ana,
V -> ama}



O Zé
ama a
Ana



Syntactic tree

ACTIVE LEARNING MOMENT



EXERCISE

- Give examples of sentences that belong to $L(G)$, being $G = (N, T, S_0, R)$:
 - $N = \{S, NP, VP, Pron, Det, Noun, Verb\}$
 - $T = \{I, They, book, João, love, a, the, that\}$
 - $S_0 = S$ (S for sentence)
 - $R: \{$
 - $S \rightarrow NP\ VP,$
 - $NP \rightarrow Pron \mid Noun \mid Det\ Noun,$
 - $VP \rightarrow Verb\ NP,$
 - $Pron \rightarrow I \mid They,$
 - $Noun \rightarrow book \mid João,$
 - $Verb \rightarrow love,$
 - $Det \rightarrow a \mid the \mid that\}$
- Use a bottom-up or top-down approach, left-to-right, to show that the sentence “I love that book” belongs to $L(G)$
- Give examples of sentences that do not belong to $L(G)$

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

PROBABILISTIC CFG GRAMMARS FORMALISM

- Probabilistic CFG Grammars
 - Each rule $C \rightarrow a_j$ has a probability associated
 - How to find those probabilities?
 - use a treebank and calculate:
 - A: the number of times $C \rightarrow a_j$ is used
 - B: the number of times the rules of the form $C \rightarrow a_i$ are used
- Then:
- $P(C \rightarrow a_j) [A/B]$

ACTIVE LEARNING MOMENT



EXERCISE

- Consider that in a **treebank**, annotated in terms of the syntactic trees of its sentences, the use of each of the rules of a given grammar is counted:

• $S \rightarrow NP\ VP$	80
• $S \rightarrow Aux\ NP\ VP$	30
• $S \rightarrow VP$	15
• $NP \rightarrow Det\ Nom$	50
• $NP \rightarrow Proper-Noun$	65
• $NP \rightarrow Nom$	15
• $NP \rightarrow Pronoun$	40
• $VP \rightarrow Verb$	40
• $VP \rightarrow Verb\ NP$	40
• $VP \rightarrow Verb\ NP\ NP$	10

What is the probability of the rule $S \rightarrow VP$?

$$\begin{aligned}15/(80+30+15) &= \\15/125 &\end{aligned}$$

PROBABILISTIC CFG GRAMMARS

- Probabilistic CFG Grammars can be used to disambiguate when several parse trees exist
 - The probability of a subtree is the multiplication of the probabilities of its own subtrees; choose the one with the highest probability

ACTIVE LEARNING MOMENT



EXERCISE

- Consider the grammar

$S \rightarrow NP\ VP$	1.0	$NP \rightarrow NP\ PP$	0.4
$PP \rightarrow P\ NP$	1.0	$NP \rightarrow \text{astronomers}$	0.1
$VP \rightarrow V\ NP$	0.7	$NP \rightarrow \text{ears}$	0.18
$VP \rightarrow VP\ PP$	0.3	$NP \rightarrow \text{saw}$	0.04
$P \rightarrow \text{with}$	1.0	$NP \rightarrow \text{stars}$	0.18
$V \rightarrow \text{saw}$	1.0	$NP \rightarrow \text{telescopes}$	0.1

Example from <https://courses.cs.washington.edu/courses/cse590a/09wi/pcfg.pdf>

EXERCISE

- Calculate the two possible syntactic trees for the sentence:

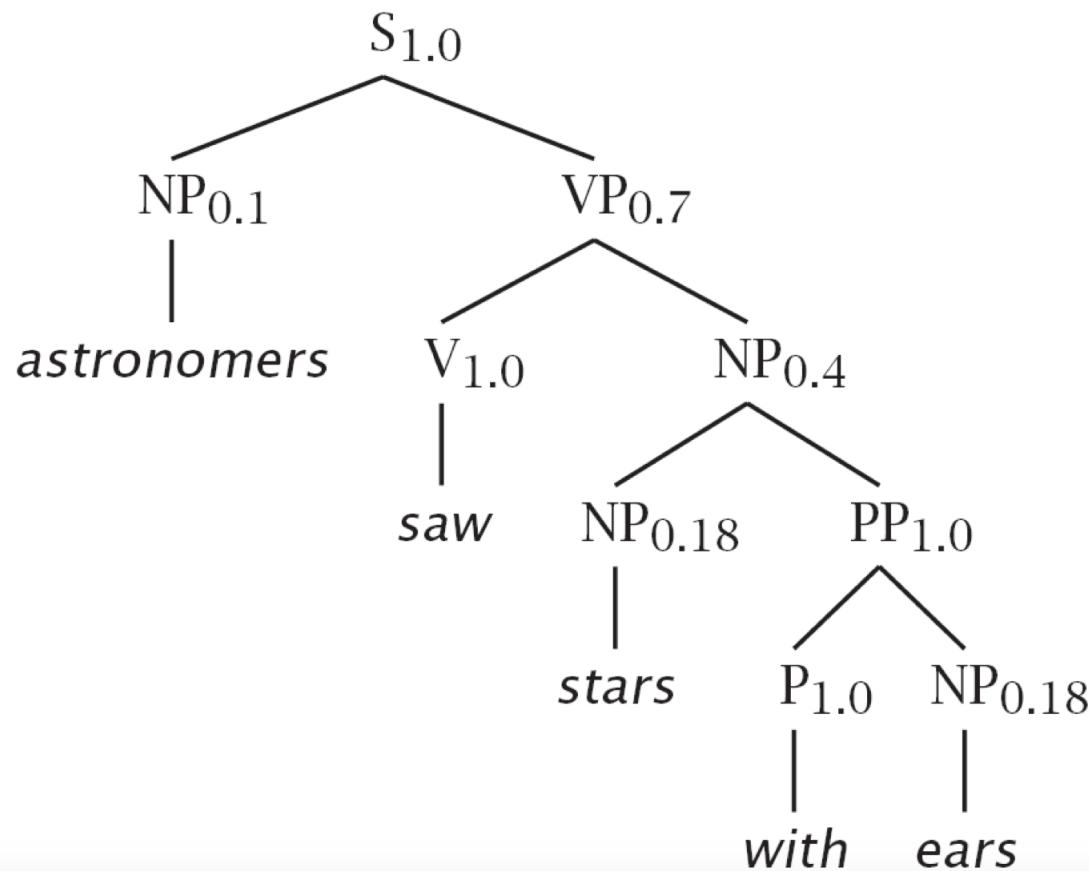
astronomers saw stars with ears

- Then, decide which one is more probable

EXERCISE

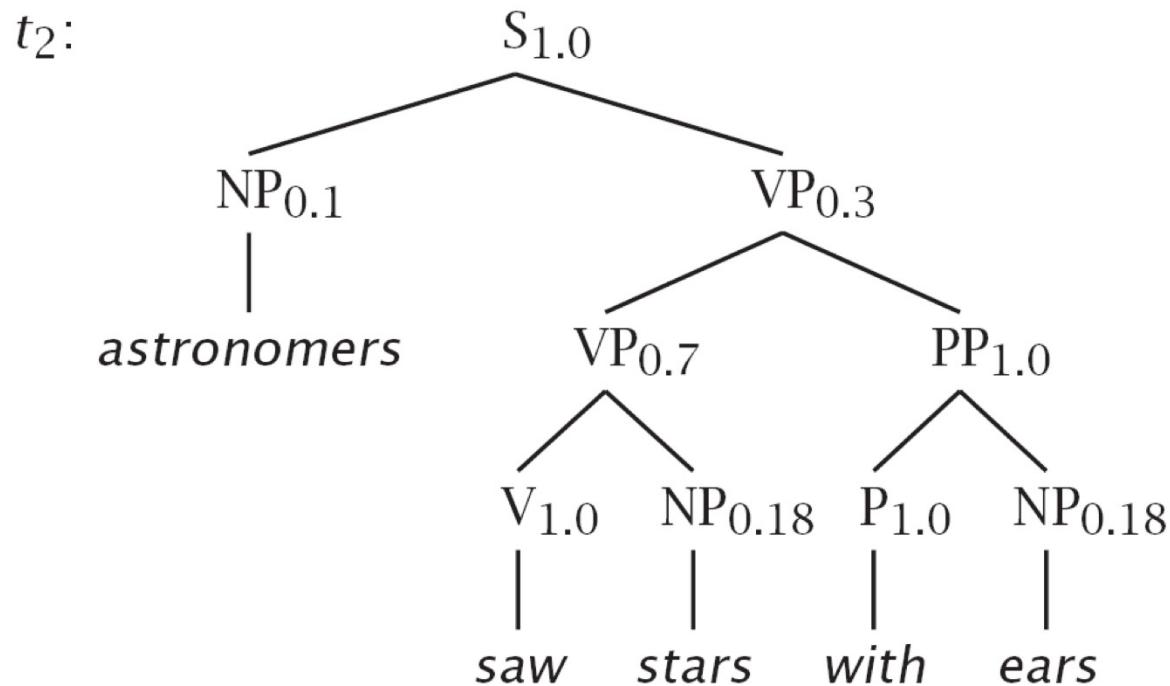
astronomers saw stars with ears

$t_1:$



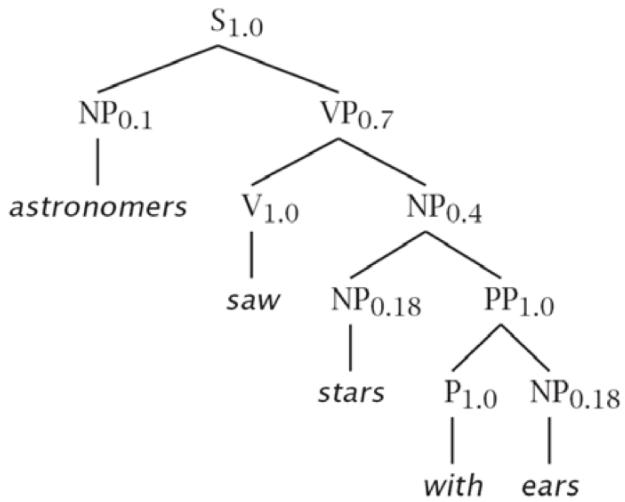
EXAMPLE

astronomers saw stars with ears



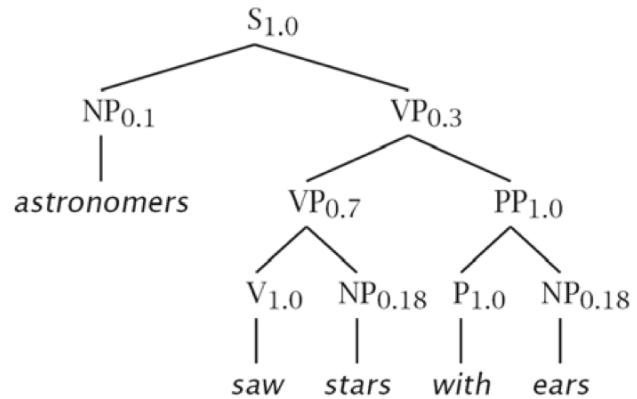
EXAMPLE

$t_1:$



$$\begin{aligned}
 P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
 &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
 &= 0.0009072
 \end{aligned}$$

$t_2:$



$$\begin{aligned}
 P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
 &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
 &= 0.0006804
 \end{aligned}$$



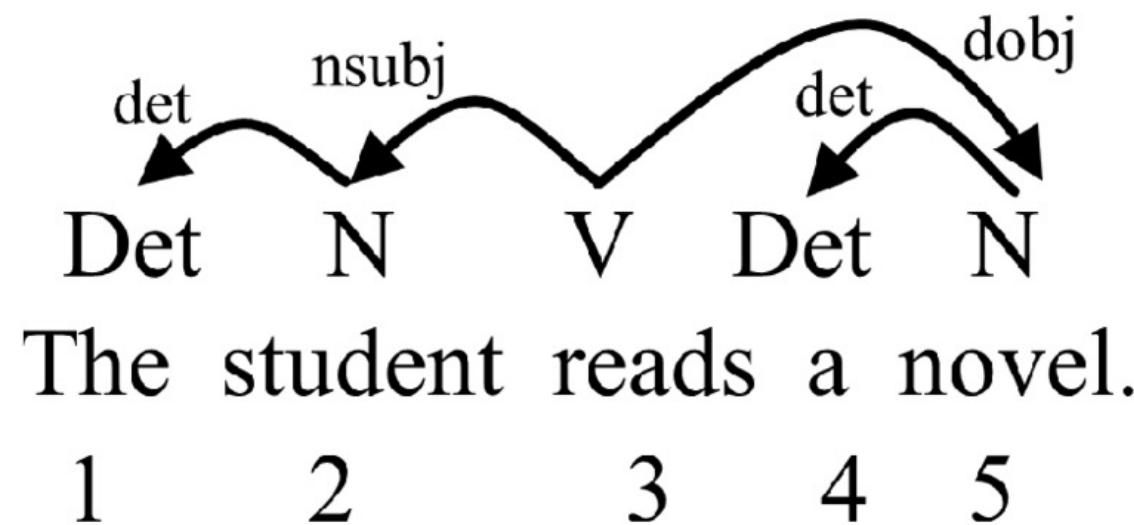
Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

DEPENDENCY GRAMMARS (DG) FORMALISM

- DGs do not use the concept of “constituent”
- A DG has the form $G = (V, A)$, in which:
 - V is a set of vertices (the tokens)
 - A (for arcs) is a set of pairs of vertices
 - Arcs can be labelled
- Each arc represents a (usually grammatical) relation between:
 - The head: role of the central organizing word
 - The dependent: a kind of modifier
- **Derivation with DGs:** sequential application of algorithms that identify and construct the dependency relations among words

EXAMPLE



https://www.researchgate.net/figure/Dependency-structure-of-the-sample-sentence-The-student-reads-a-novel_fig1_332428184

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

EXAMPLE OF A REAL APPLICATION

- EP2LGP5.0: translates from European Portuguese (EP) to the Portuguese Sign Language (LGP)
- Challenge:
 - EP grammar is different from LGP
 - Example:
 - EP: A rainha foi à praia. (The queen went to the beach.)
 - LGP: MULHER REI PRAIA IR (WOMAN KING BEACH GO)



GLOSSES

ACTIVE LEARNING MOMENT



EXERCISE

Portuguese sentences

Preciso ir dormir.

(I need to sleep.)

Queres uma xícara de café?

(Do you want a cup of coffee?)

Você não é uma minoria

(You are not a minority)

Até questionei a minha sanidade

(I even questioned my sanity)

EXERCISE

SIM MEU MULHER SENHOR
(YES MY WOMAN SIR)

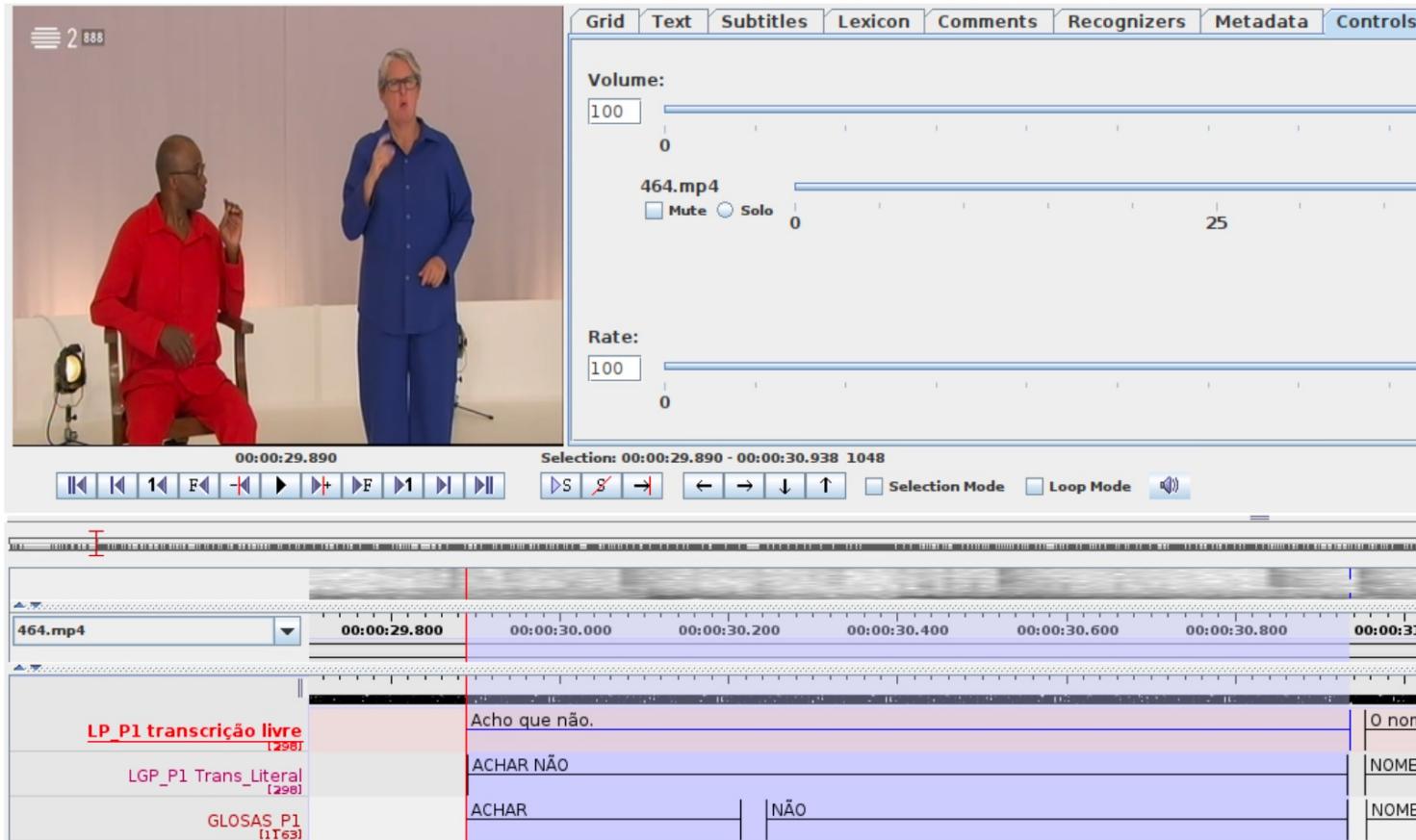
TRISTE MULHER RAPAZ EU ACHAR
(SAD WOMAN BOY I THINK)

192 PARTILHA ELE ATINGIR
(192 SHARE IT REACH)

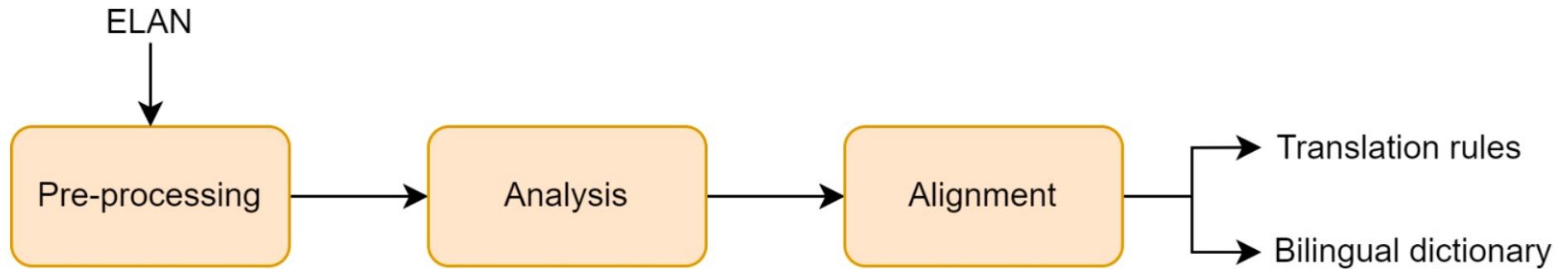
SALARIAIS ALTERNATIVA CORTES
HAVER
(WAGE ALTERNATIVE CUTS THERE
IS)

FROM EP TO LGP (FROM SOUSA 2023)

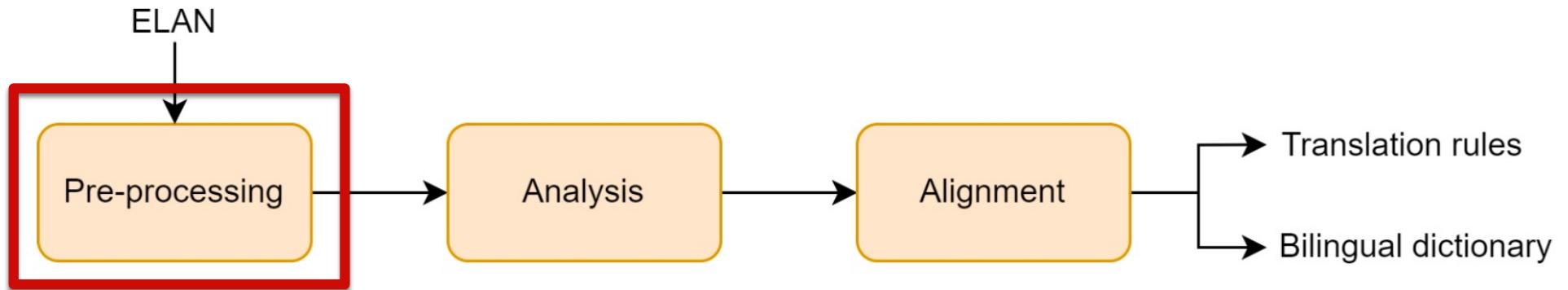
- EP2LGP5.0 takes advantage of an annotated corpus from Católica (with ELAN) – only corpus available between these two languages



FROM EP TO LGP (FROM SOUSA 2023)

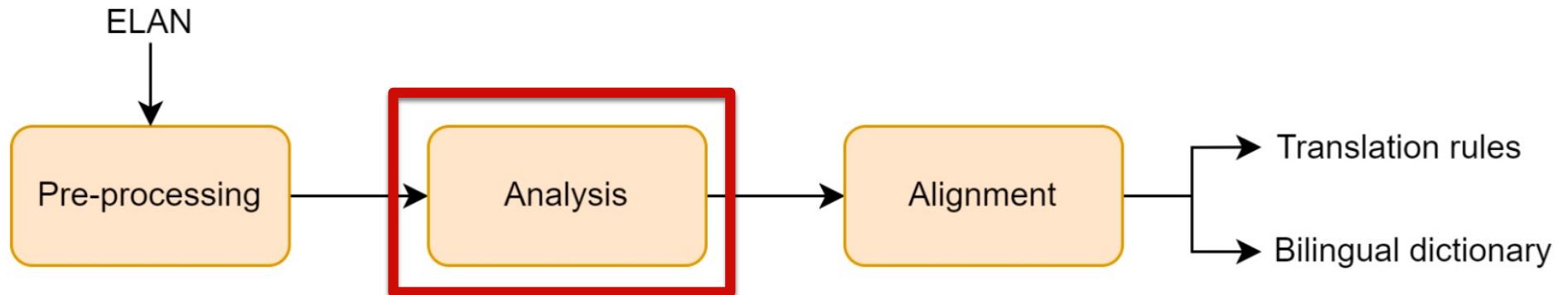


FROM EP TO LGP (FROM SOUSA 2023)

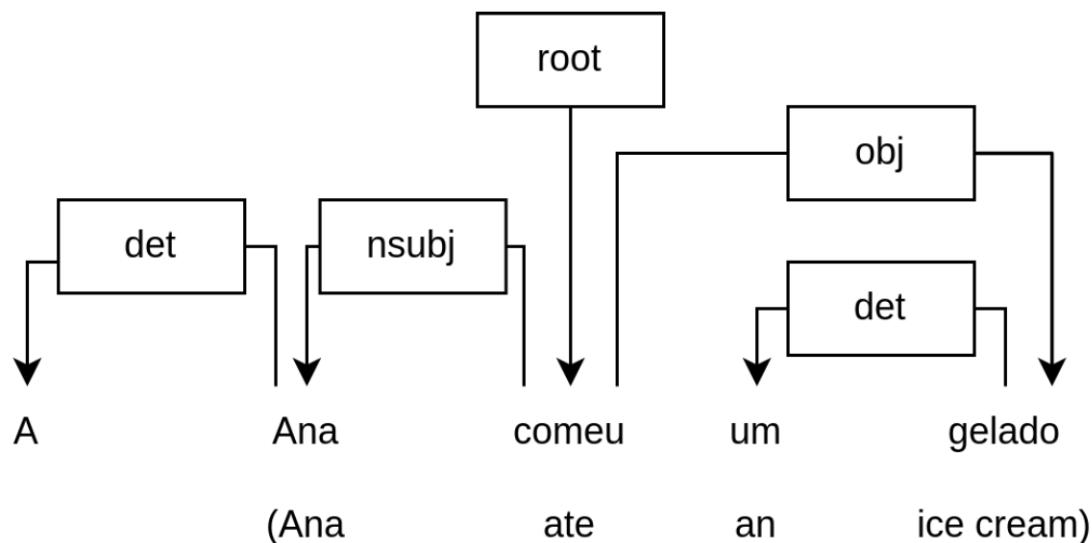


- Extracts the glosses and their grammatical classes, the type of the sentence, ...

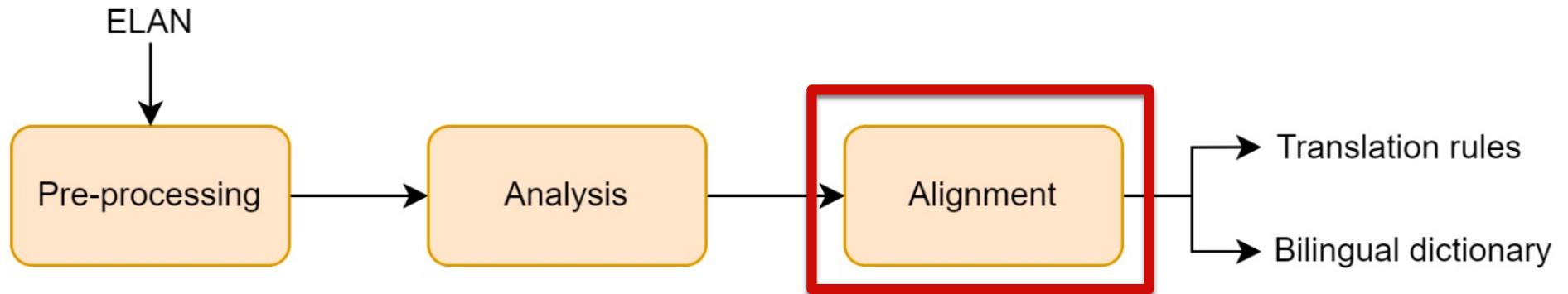
FROM EP TO LGP (FROM SOUSA 2023)



- EP sentences: PoS + Syntactic analysis (dependency relations) + removal of determinants and punctuation



FROM EP TO LGP (FROM SOUSA 2023)



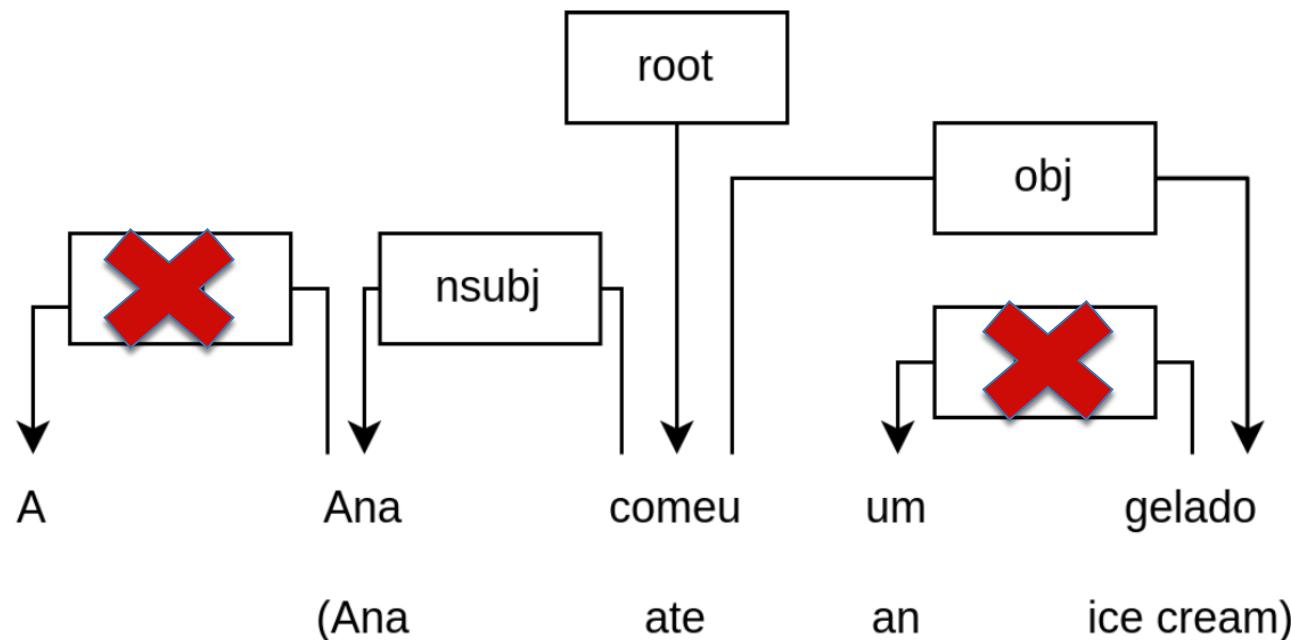
- EP words and LGP glosses are aligned by applying an algorithm based on similarity measures
 - From the alignment, a **bilingual dictionary** and a set of **translation rules** are inferred

FROM EP TO LGP (FROM SOUSA 2023)

- Example of a rule in the [Bilingual Dictionary](#)
 - religião (religion) → IGREJA (CHURCH)
 - houve grande (there was great) → TER – MUITO (HAVE –A–LOT)
- Examples of [Translation Rules](#)
 - Morphosyntactic rules (228 rules)
 - $V_1 N_1 \rightarrow N_1 V_1$
 - General syntactic rules (238 rules)
 - $VP\ NP \rightarrow NP\ VP$

FROM EP TO LGP (FROM SOUSA 2023)

- Running example:
 - A Ana comeu um gelado (Ana ate an ice-cream)



FROM EP TO LGP (FROM SOUSA 2023)

- Running example:
 - A Ana comeu um gelado (Ana ate an ice-cream)

Subject : $N1 \text{ CAN} \rightarrow N1 \text{ CAN}$

Predicate : $V1 \ N2 \text{ CAN} \rightarrow N2 \ V1 \text{ CAN}$

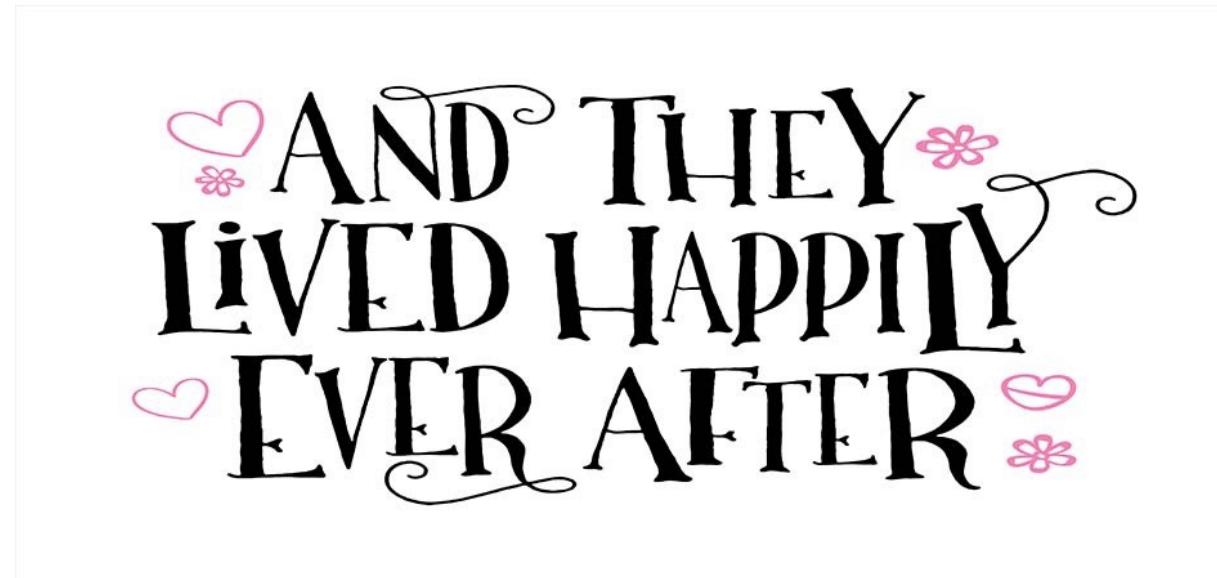
A Ana comeu um gelado. \rightarrow *Ana gelado comeu*

A Ana comeu um gelado. \rightarrow $DT(A - N - A) \text{ GELADO COMER}$

- Notes:
 - “CAN” states that the sentence is declarative
 - $DT(A-N-A)$ means that “Ana” (as a proper name) should be fingerspelled

FROM EP TO LGP (FROM SOUSA 2023)

- To the ones that adore Deep Learning:
 - The created rules were used to create a parallel corpus between EP and LGP
 - Deep Learning models were built



Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

LLMs and Syntactic Parsing

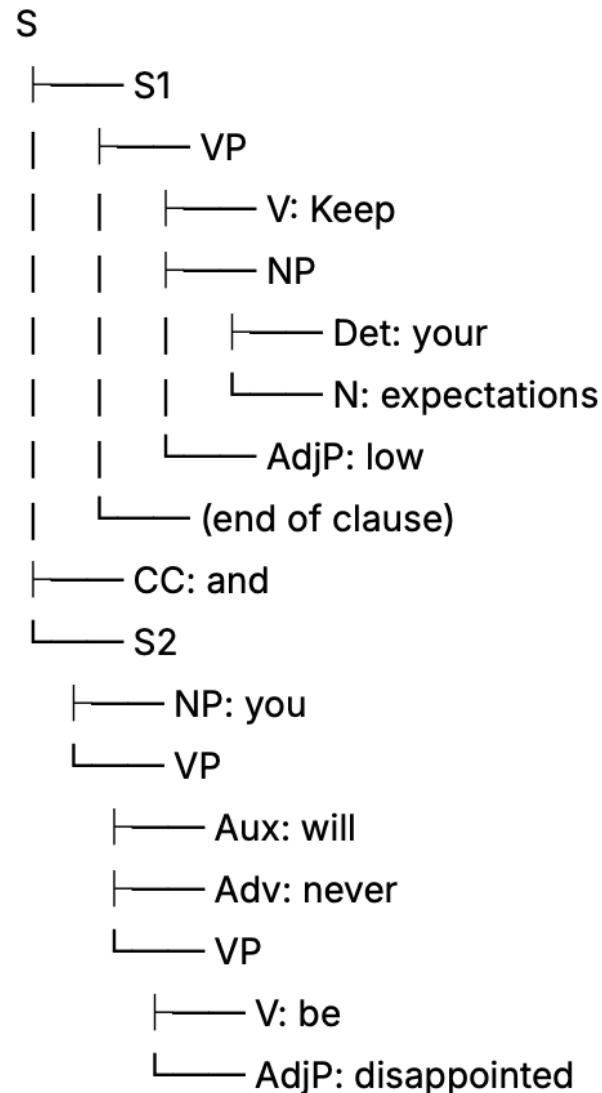
- Task: make a syntactic tree of the sentence:

“keep your expectations low and you'll never be disappointed.”

GEMINI (2.5 FLASH)

```
S (Compound Sentence)
└─ S (Independent Clause 1 / Imperative)
    └─ VP (Verb Phrase)
        └─ V: Keep
        └─ NP (Noun Phrase - Direct Object)
            └─ Det: your
            └─ N: expectations
            └─ AdjP (Adjective Phrase - Object Complement): low
    └─ CONJ: and
    └─ S (Independent Clause 2 / Declarative)
        └─ NP (Noun Phrase - Subject): you
        └─ Aux: 'll (will)
        └─ VP (Verb Phrase)
            └─ Adv: never
            └─ V: be
            └─ AdjP (Adjective Phrase - Predicate Adjective): disappointed
```

GPT-5 Nano



MISTRAL

```
S (Sentence)
  └ ImperativeCl (Imperative Clause)
    └ VP (Verb Phrase)
      └ V (Verb): keep
        └ NP (Noun Phrase)
          └ PossPron (Possessive Pronoun): your
            └ N (Noun): expectations
      └ AP (Adjective Phrase)
        └ Adj (Adjective): low
    └ CoordConj (Coordinate Conjunction): and
    └ S (Sentence)
      └ NP (Noun Phrase)
        └ Pron (Pronoun): you'll
      └ Adv (Adverb): never
      └ VP (Verb Phrase)
        └ Aux (Auxiliary): be
        └ V (Past Participle): disappointed
```

```

S (Compound Sentence)
└── S (Independent Clause 1 / Imperative)
    └── VP (Verb Phrase)
        ├── V: Keep
        └── NP (Noun Phrase - Direct Object)
            ├── Det: your
            └── N: expectations
    └── AdjP (Adjective Phrase - Object Complement): low
    └── CONJ: and
    └── S (Independent Clause 2 / Declarative)
        ├── NP (Noun Phrase - Subject): you
        ├── Aux: 'll (will)
        └── VP (Verb Phrase)
            ├── Adv: never
            ├── V: be
            └── AdjP (Adjective Phrase - Predicate Adjective): disappointed

```

```

S (Sentence)
└── ImperativeCl (Imperative Clause)
    └── VP (Verb Phrase)
        ├── V (Verb): keep
        └── NP (Noun Phrase)
            ├── PossPron (Possessive Pronoun): your
            └── N (Noun): expectations
    └── AP (Adjective Phrase)
        └── Adj (Adjective): low
└── CoordConj (Coordinate Conjunction): and
    └── S (Sentence)
        ├── NP (Noun Phrase)
        │   └── Pron (Pronoun): you'll
        ├── Adv (Adverb): never
        └── VP (Verb Phrase)
            ├── Aux (Auxiliary): be
            └── V (Past Participle): disappointed

```

```

S
└── S1
    └── VP
        ├── V: Keep
        └── NP
            ├── Det: your
            └── N: expectations
    └── AdjP: low
    └── (end of clause)
└── CC: and
└── S2
    ├── NP: you
    └── VP
        ├── Aux: will
        ├── Adv: never
        └── VP
            ├── V: be
            └── AdjP: disappointed

```

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - [An example of a syntactic parser](#)
- Key takeaways
- Suggested readings

SYNTACTIC PARSING WITH THE CKY ALGORITHM

- The Cocke-Kasami-Younger (CKY) algorithm uses dynamic programming.
- Constraint: grammars must be in the Chomsky Normal Form (CNF).
 - Rules must have one of the following forms:
 - NonTerminal \rightarrow NonTerminal_1 NonTerminal_2
 - NonTerminal \rightarrow terminal

SYNTACTIC PARSING WITH THE CKY ALGORITHM

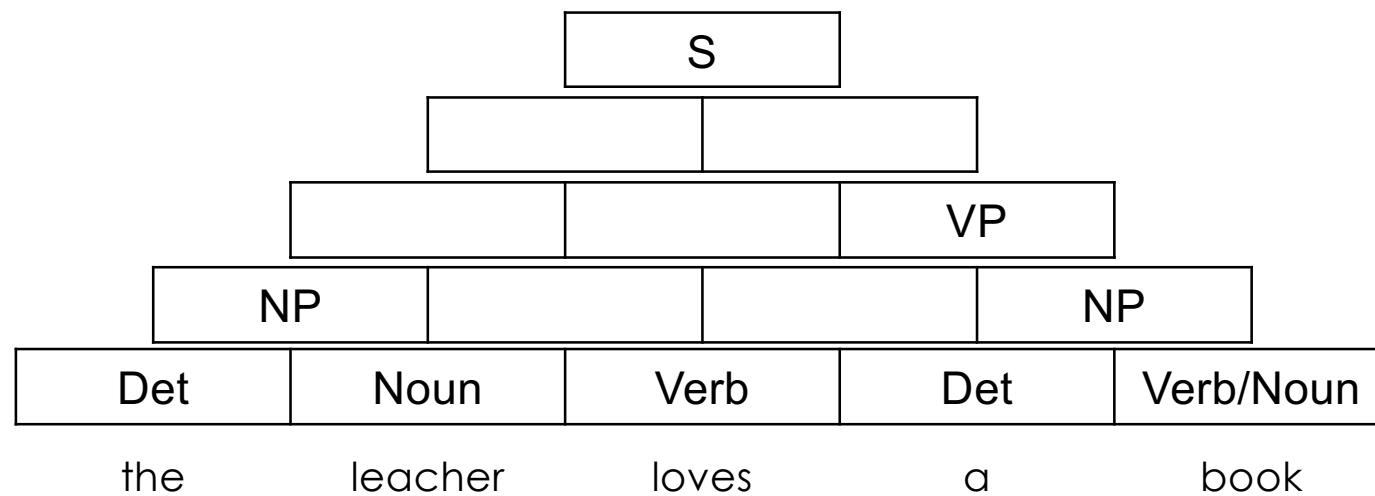
Algorithm 3 CKY

```
j ← 1
while j < n do
    [1, j] = {A : A → wj ∈ R}
    j ++
end while
i ← 2
while i < n do
    j ← 1
    while j < n - i + 1 do
        [i, j] = ∪m=1i-1 {A : A → B C ∈ R, B ∈ [m, j], C ∈ [i - m, j + m]}
        j ++
    end while
    i ++
end while
if S0 ∈ [n, 1] then
    W ∈ L(G)
end if
```

SYNTACTIC PARSING WITH THE CKY ALGORITHM

- Use the CKY algorithm to show that the sentences "the teacher loves a book" $\in L(G)$, and that "the teacher loves a" $\notin L(G)$, being:
- $G = (N, T, S_0, R)$:
 - $N = \{S, NP, VP, Det, Noun, Verb\}$
 - $T = \{\text{book}, \text{João}, \text{love}, \text{a}, \text{the}, \text{that}\}$
 - $S_0 = S$ (S for sentence)
 - $R: \{$
 - $S \rightarrow NP\ VP$
 - $NP \rightarrow Det\ Noun,$
 - $VP \rightarrow Verb\ NP,$
 - $Noun \rightarrow \text{book} \mid \text{table} \mid \text{teacher},$
 - $Verb \rightarrow \text{loves} \mid \text{book}$
 - $Det \rightarrow a \mid \text{the} \mid \text{that}\}$

SYNTACTIC PARSING WITH THE CKY ALGORITHM



KEY TAKEAWAYS

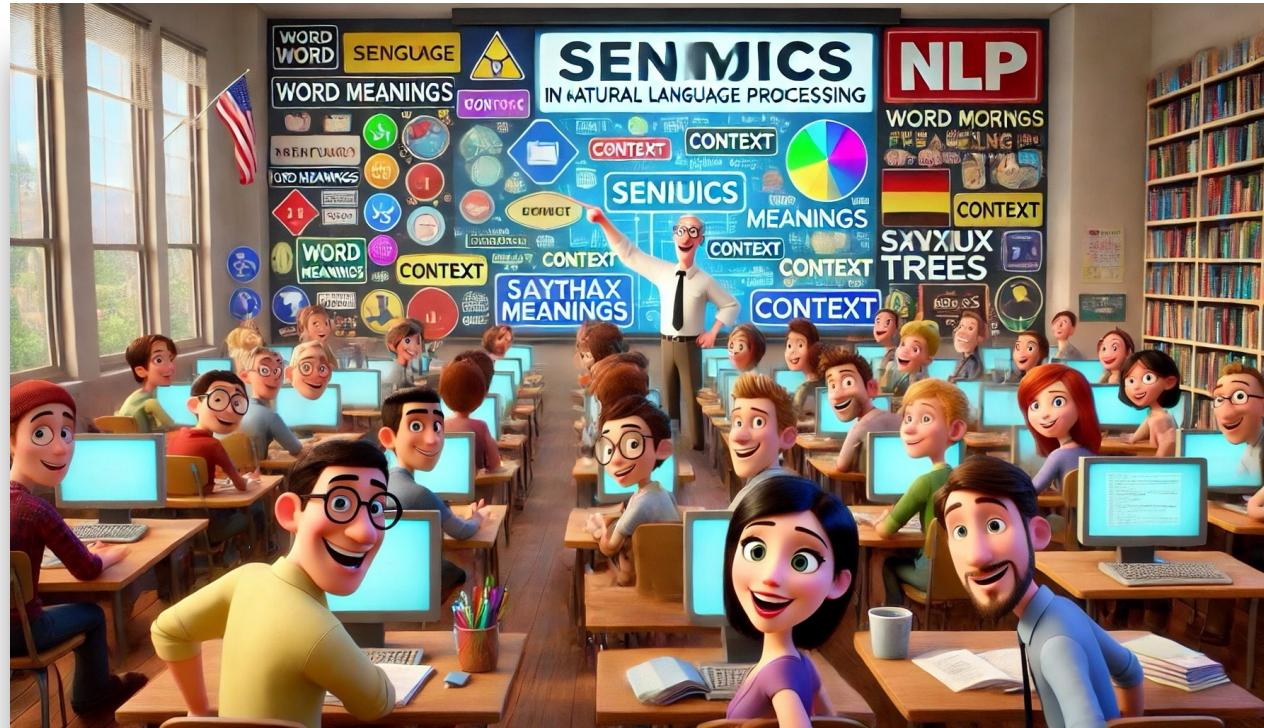
KEY TAKEAWAYS

- Explain the concepts of treebank, constituency grammar, context-free grammar, probabilistic context-free grammar and dependency grammar
- Understand what is the language generated by a CFG
- Apply CKY

SUGGESTED READINGS

READINGS

- Sebenta:
 - Syntax important Natural very Language is
- Jurafsky:
 - Chapter 17 (Context-Free Grammars and...)
 - 17.1-17.3



SEMANTICS

Luisa Coheur

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - Semantic Relations
 - Semantic Resources
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to
 - Define main concepts from Computational Semantics
 - Identify several semantic resources
 - Given a grammar with semantic rules, apply them and obtain the logical form representing the semantics of a given sentence

TOPICS

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - Semantic Relations
 - Semantic Resources
- Key takeaways
- Suggested readings

COMPUTATIONAL SEMANTICS

- The enterprise of designing meaning representations and associated semantic parsers is referred to as computational semantics!!!

COMPUTATIONAL SEMANTICS

- Mapping what the user says in something that captures that meaning and the computer understands
- Let us see, in practice, what this means

COMPUTATIONAL SEMANTICS

- Mapping what the user says in something that captures that meaning and the computer understands
- Scenario

COMPUTATIONAL SEMANTICS

- Mapping what the user says in something that captures that meaning and the computer understands
- Scenario
- Meaning Representation
 - can be First Order Logic, but also a vector...

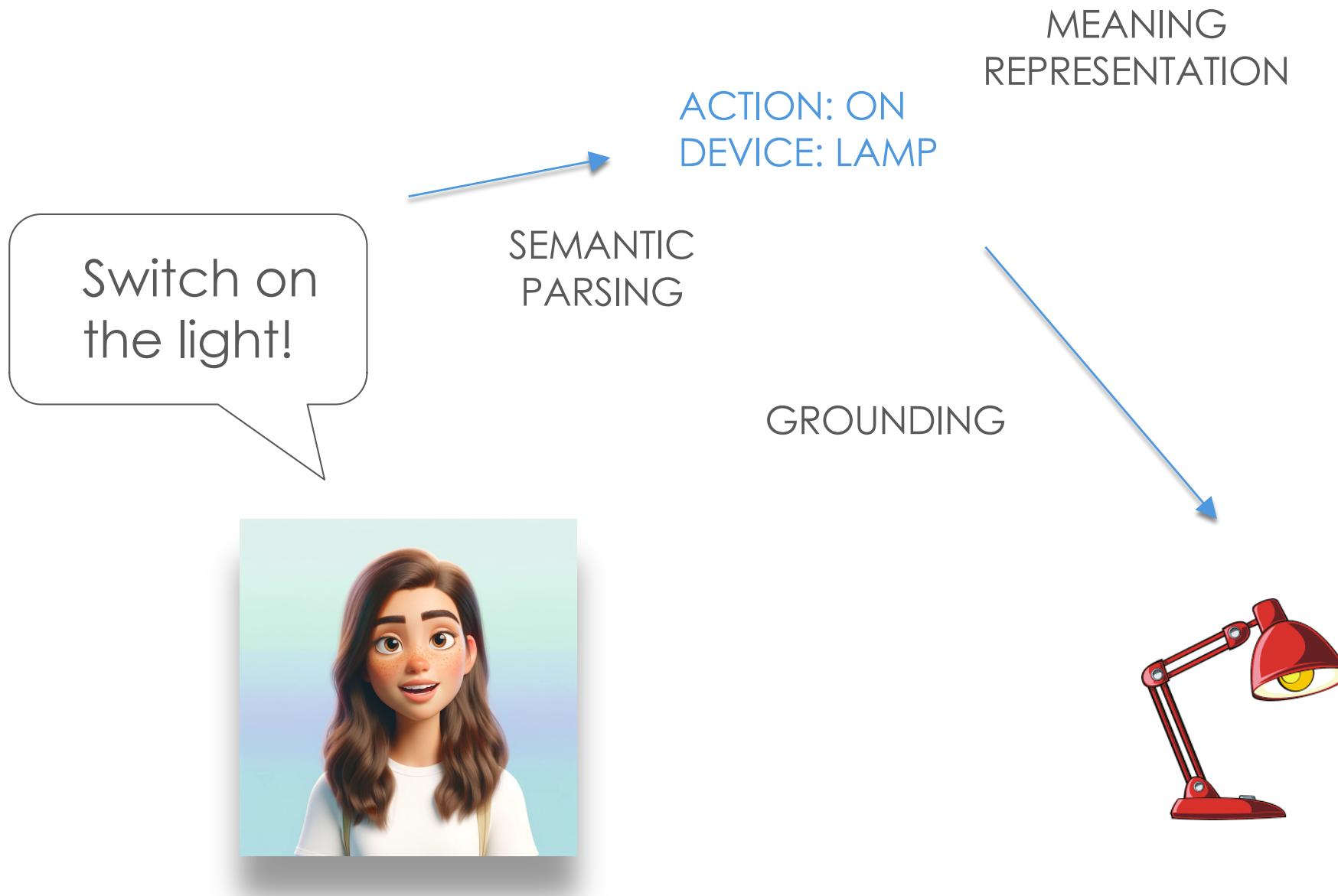
COMPUTATIONAL SEMANTICS

- Mapping what the user says in something that captures that meaning and the computer understands
- Scenario
- Meaning Representation
 - can be First Order Logic, but also a vector
- Actionable Meaning Representation (ex: SQL)

COMPUTATIONAL SEMANTICS

- Mapping what the user says in something that captures that meaning and the computer understands
- Scenario
- Meaning Representation
 - can be First Order Logic, but also a vector
- Actionable Meaning Representation (ex: SQL)
- Semantic parsing or semantic analysis

COMPUTATIONAL SEMANTICS



COMPUTATIONAL SEMANTICS

- Symbolic representation of language: a meaning representation consists of a set of symbols, human-interpretable, corresponding to:
 - Objects (ex: lamp),
 - Properties of objects (ex: red), or
 - relations among objects in some world (ex: love(X, Y))
- So... not vectors

Overview

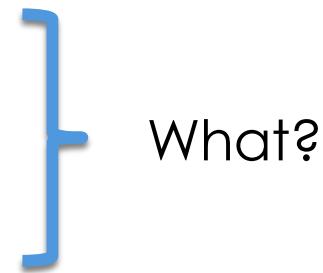
- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - Semantic Relations
 - Semantic Resources
- Key takeaways
- Suggested readings

SYMBOLIC REPRESENTATION

- Some symbolic formalisms
 - First Order Logic (or some subsets/simplified versions of it)
 - Discourse Representation Structures (DRS) (Kamp and Reyle, 1993)
 - ...
 - Abstract Meaning Representation (AMR) (Banarescu et al. 2013)
 - ...

SYMBOLIC REPRESENTATION

- Main used representations
 - Keywords
 - Sentences
 - Regular expressions/templates
 - Logical forms
 - Frames
 - Graphs



What?

SYMBOLIC REPRESENTATION

Keywords

Robot understands:

“left”, “right”, “stop”
(that is, it knows how to
act, when those words
are given as input)

Sentences

Agent understands:

“Who built the
palace” (that
is, it knows how to
answer it)



This can be seen
as understanding
fixed sequences
of keywords.

Regular expressions

Agent understands:

“Who built .* palace” (that is,
it knows how to answer it)



Can be seen as
understanding
keywords within
some pre-defined
order.

SYMBOLIC REPRESENTATION

- First Order Logic:
 - You probably studied this before (v.g., in Logic for Programming)
 - Define constants and relations; use: $\exists \forall \neg \wedge \vee \Rightarrow$
 - Examples:
 - Pedro is a student: $\text{student}(\text{PEDRO})$
 - Note: never use $p1(p2)$, being $p1$ and $p2$ predicates
 - All students are great: $\forall x \text{ student}(x) \Rightarrow \text{great}(x)$
 - Note: \forall “asks” for \Rightarrow
 - Why $\forall x \text{ student}(x) \wedge \text{great}(x)$ Is not an option?
 - There is (at least) one great student: $\exists x \text{ student}(x) \wedge \text{great}(x)$
 - Note: \exists “asks” for \wedge .
 - Why $\exists x \text{ student}(x) \Rightarrow \text{great}(x)$ Is not an option?

ACTIVE LEARNING MOMENT



EXERCISE

- Consider:
 - The constant: John
 - The predicates:
 - $\text{sister}(X, Y)$ that is true if Y is X's sister
 - $\text{like}(X, Y)$ that is true if X likes Y
- Represent in First Order Logic:
 - John likes his sisters
 - John has (at least) one sister
 - All of John's sisters like him

EXERCISE

- Consider:
 - The constant: John
 - The predicates:
 - $\text{sister}(X, Y)$ that is true if Y is X's sister
 - $\text{like}(X, Y)$ that is true if X likes Y
- Represent in First Order Logic:
 - John likes his sisters: $\forall x [\text{sister}(\text{John}, x) \Rightarrow \text{likes}(\text{John}, x)]$
 - John has (at least) one sister: $\exists x \text{sister}(\text{John}, x)$
 - All of John's sisters like him: $\forall x [\text{sister}(\text{John}, x) \Rightarrow \text{likes}(x, \text{John})]$

SYMBOLIC REPRESENTATION

- Frames:
 - Concept coined by Minsky (much more complicated)

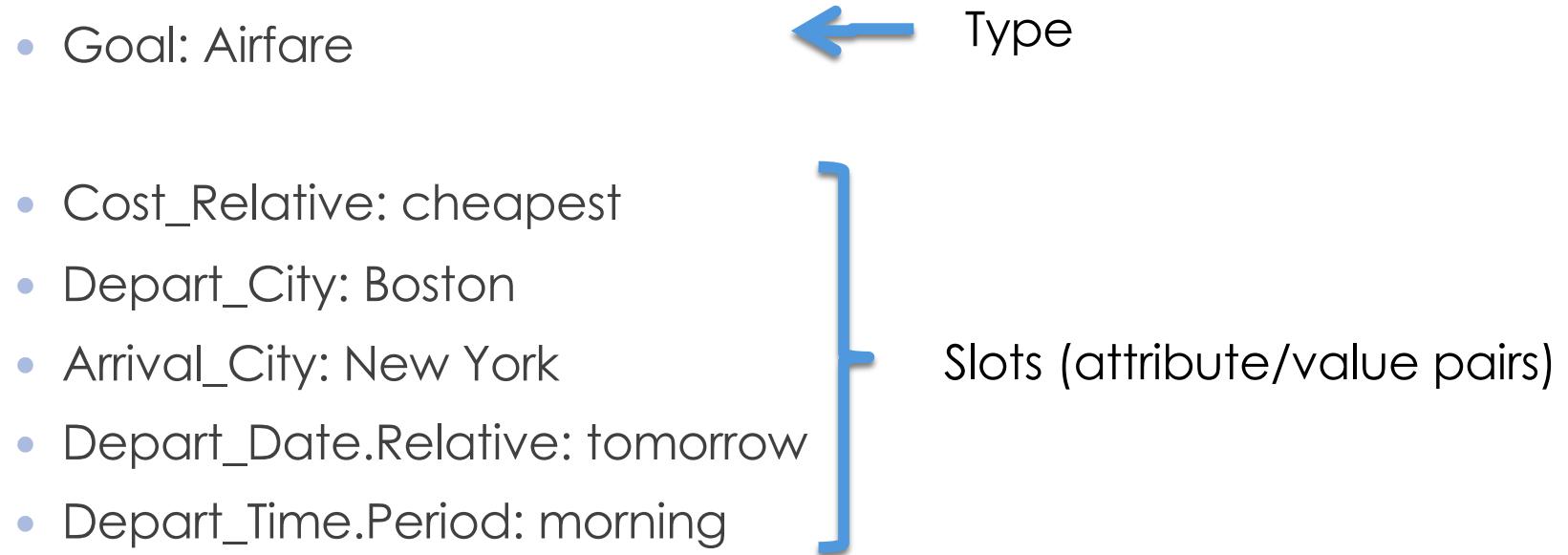


Marvin Lee Minsky

- Simplified definition of frame (Ward and Issar, 1994):
 - Type:
 - Types of actions that can be taken by the system (intent);
 - Slots:
 - Pieces of information relevant to the action
 - Typically pairs attribute/value

SYMBOLIC REPRESENTATION

- Example (Tur et al., 2010):
 - Input:
 - How much is the cheapest flight from Boston to New York tomorrow morning?
 - Meaning Representation as a frame:
 - Goal: Airfare



ACTIVE LEARNING MOMENT



EXERCISE

- Imagine that you are going to create a dialogue system where customers interact with a bot to book cinema tickets. Which types (intents) and slots should be used?
- Possible solution:

Types:

- Movies_Available
- Showtime
- Seat_Availability
- Ticket Booking
- Payment
- Booking Confirmation
- Booking Cancellation
- Booking Modification
- ...

Slots:

- movie_title
- number_of_tickets
- date
- cinema
- seat_preferences
- accessibility_options
- language
- subtitle_options
- ...

REAL EXAMPLE

Customer: I'd like to book tickets for **Strange Days** next **Friday**.

Bot: Sure! You want to book tickets for Strange Days on next Friday. We have showtimes at 3:00 PM, 6:00 PM, and 9:00 PM. Which time works best for you?

Customer: Let's go with the **6 PM show**.

Bot: Got it! You've selected the 6:00 PM show for Strange Days on Friday, October 11th. Which cinema would you prefer? We have Colombo and Amoreiras available.

Customer: **Colombo** sounds good.

Bot: "Perfect! [...] How many tickets would you like?"

- Customer: "Two tickets, please."

movie_title: StrangeDays
number_of_tickets: 2
date/time: Friday 6 PM.
cinema: Colombo

REAL EXAMPLE

- Bot: Great! You're booking two tickets for StrangeDays at Colombo, 6:00 PM, on Friday, October 11th. Would you like to proceed with payment?
- Customer: Actually, I changed my mind. Can I book Memento instead?
- ...

```
movie_title: StrangeDays Memento  
number_of_tickets: 2  
date/time: Friday 6 PM.  
cinema: Colombo
```

REAL EXAMPLE

- Entity Recognition (Named Entity Recognition):
 - Recognize key entities such as:
 - Movie titles (e.g., Memento)
 - Dates (e.g., "next Friday," "October 11")
 - Times (e.g., "6 PM show")
 - Cinema locations (e.g., "Colombo")
 - ...

By the way...



SYMBOLIC MEANING REPRESENTATION (TO CONCLUDE)

- “I have a car” in: a) FOL (top), b) Abstract Meaning Representation (AMR) – graph and textual forms – and, c) framed-based or slot-filler

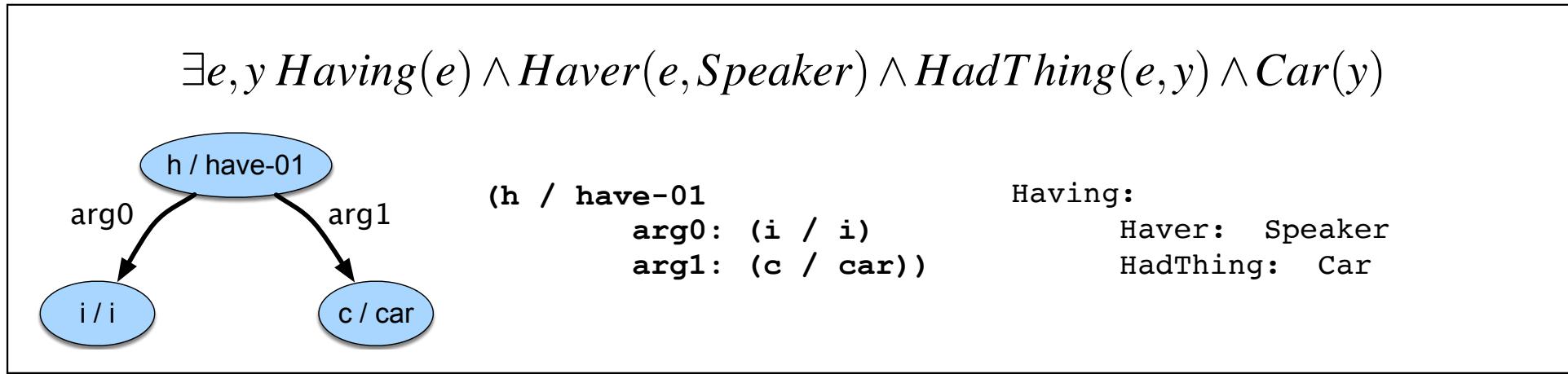


Figure 16.1 A list of symbols, two directed graphs, and a record structure: a sampler of meaning representations for *I have a car*.

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - [Compositional Semantic Parsing](#)
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - Semantic Relations
 - Semantic Resources
- Key takeaways
- Suggested readings

COMPOSITIONAL SEMANTIC PARSING

- Semantic parsing (**compositional process**) generates meaning representations by **recursively combining the meanings of smaller subcomponents** (such as words or phrases) according to syntactic structures
- **Richard Montague** was the first person to present a framework that systematically maps natural language to formal meaning representations (early 1970s)

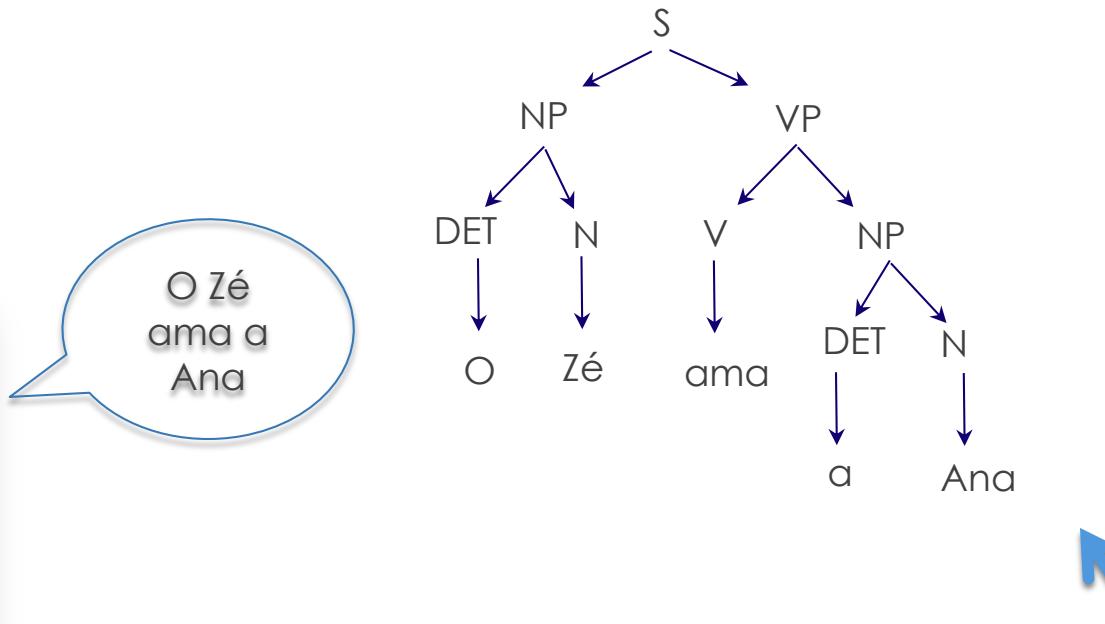


COMPOSITIONAL SEMANTIC PARSING

- Notation (from Sebenta):
 - $x.\text{sem}$: semantics of x
 - $\text{replace}(A.\text{sem} N B.\text{sem})$: replace the semantics of A in the N th argument of the semantics of B .

EXAMPLE

Grammar	Lexicon	POS
$S \rightarrow NP\ VP$	o, a	DET
$NP \rightarrow DET\ N$	Zé, Ana	N
$VP \rightarrow V\ NP$	ama	V

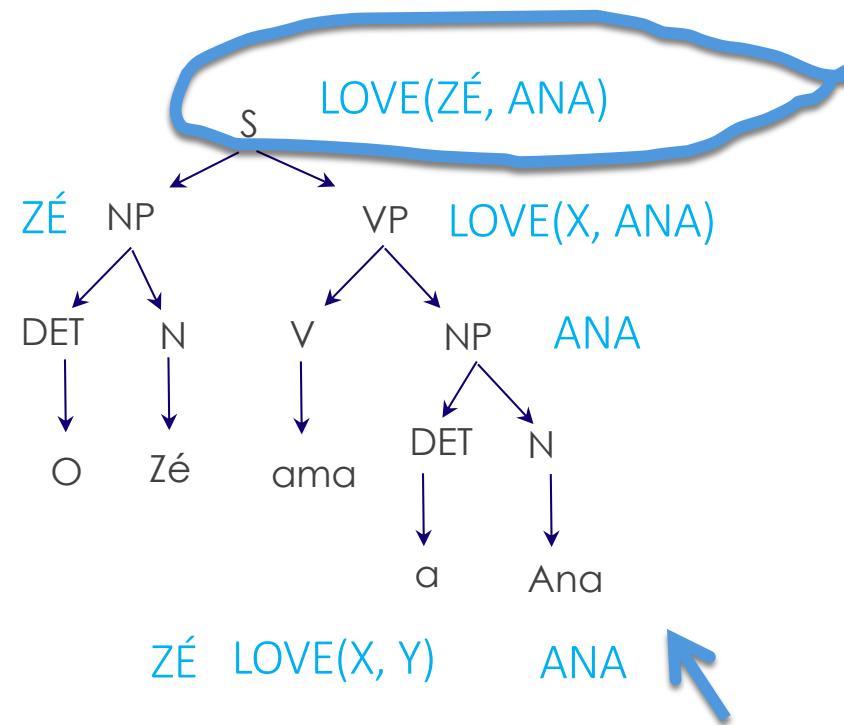
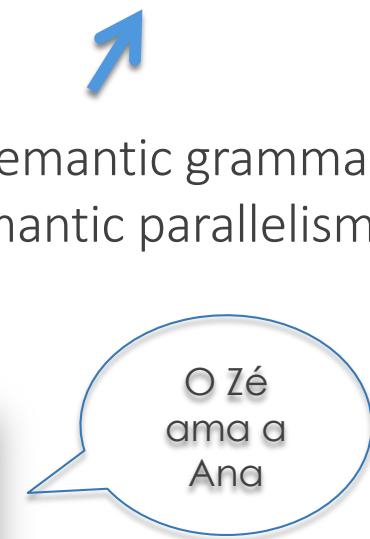


Syntactic tree

EXAMPLE

Grammar/Semantic Rules	Lexicon/SEM	POS
$S \rightarrow NP\ VP$ replace(NP.sem 1 VP.sem)	o, a/---	DET
$NP \rightarrow DET\ N$ N.sem	Zé, Ana/ZÉ, ANA	N
$VP \rightarrow V\ NP$ replace(NP.sem 2 V.sem)	Ama/LOVE(X, Y)	V

Syntactic/semantic grammar
(syntax/semantic parallelism)



Compositional (bottom-up) process

COMPOSITIONAL SEMANTIC PARSING

- Notation (From Eisenstein's NLP book):
 - From Lambda Calculus:
 - $(\lambda y.\lambda x.A(x, y))@a = \lambda x.A(x, a)$ (β -reduction)
 - @ indicates function application
 - Example:
 - $\lambda x.\text{LOVES}(x, \text{MARIA})$
 - is the meaning of "loves Maria"
 - $(\lambda x.\text{LOVES}(x, \text{MARIA}))@\text{ZÉ} = \text{LOVES}(\text{ZÉ}, \text{MARIA})$

ACTIVE LEARNING MOMENT



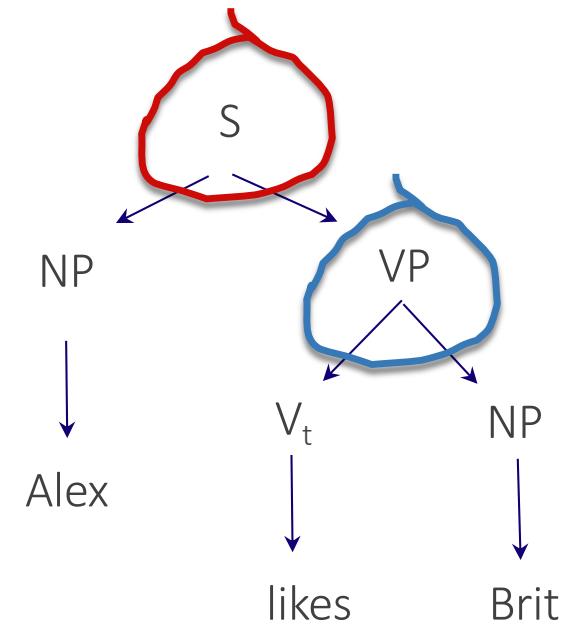
COMPOSITIONAL SEMANTIC PARSING

- Considering the given grammar (from Eisenstein book), find the meaning representation of:
 - *Alex likes Brit*

S	\rightarrow NP VP	VP.sem@NP.sem
VP	\rightarrow V_t NP	$V_t.sem@NP.sem$
VP	\rightarrow V_i	$V_i.sem$
V_t	\rightarrow <i>likes</i>	$\lambda y.\lambda x.\text{LIKES}(x, y)$
V_i	\rightarrow <i>sleeps</i>	$\lambda x.\text{SLEEPS}(x)$
NP	\rightarrow <i>Alex</i>	ALEX
NP	\rightarrow <i>Brit</i>	BRIT

S	$\rightarrow NP \ VP$	$VP.sem @ NP.sem$
VP	$\rightarrow V_t \ NP$	$V_t.sem @ NP.sem$
VP	$\rightarrow V_i$	$V_i.sem$
V_t	$\rightarrow likes$	$\lambda y. \lambda x. LIKES(x, y)$
V_i	$\rightarrow sleeps$	$\lambda x. SLEEPS(x)$
NP	$\rightarrow Alex$	ALEX
NP	$\rightarrow Brit$	BRIT

- Alex likes Brit



$$\begin{aligned}
 V_t.sem @ NP.sem &= \\
 (\lambda y. \lambda x. LIKES(x, y)) @ BRIT &= \\
 \lambda x. LIKES(x, BRIT)
 \end{aligned}$$

$$\begin{aligned}
 VP.sem @ NP.sem &= \\
 (\lambda x. LIKES(x, BRIT)) @ ALEX &= \\
 LIKES(ALEX, BRIT)
 \end{aligned}$$

Too easy. I am sure there is something you are not telling us...





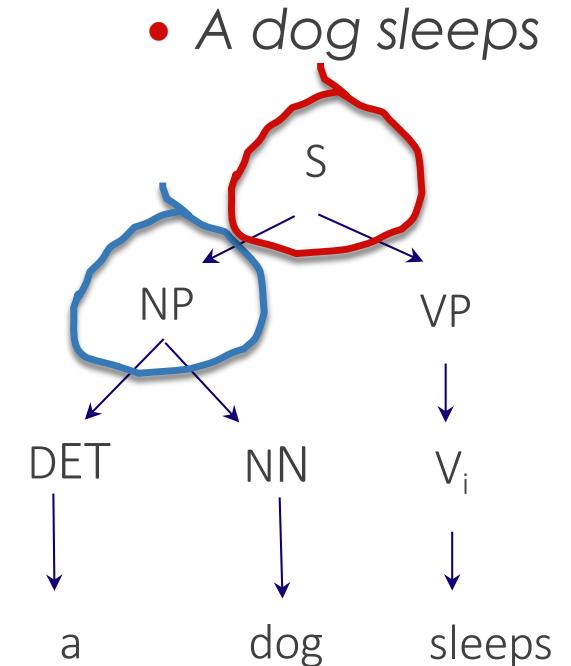
COMPOSITIONAL SEMANTIC PARSING

- Considering this other grammar (also from Eisenstein book), find the meaning representation of:

- A dog *sleeps*
- A dog *likes* Alex

S	\rightarrow NP VP	NP.sem@VP.sem
VP	\rightarrow V_t NP	$V_t.sem@NP.sem$
VP	\rightarrow V_i	$V_i.sem$
NP	\rightarrow DET NN	DET.sem@NN.sem
NP	\rightarrow NNP	$\lambda P.P(NNP.sem)$
DET	\rightarrow <i>a</i>	$\lambda P.\lambda Q.\exists xP(x) \wedge Q(x)$
DET	\rightarrow <i>every</i>	$\lambda P.\lambda Q.\forall x(P(x) \Rightarrow Q(x))$
V_t	\rightarrow <i>likes</i>	$\lambda P.\lambda x.P(\lambda y.LIKES(x, y))$
V_i	\rightarrow <i>sleeps</i>	$\lambda x.SLEEPS(x)$
NN	\rightarrow <i>dog</i>	DOG
NNP	\rightarrow <i>Alex</i>	ALEX
NNP	\rightarrow <i>Brit</i>	BRIT

S	$\rightarrow NP\ VP$	$NP.sem @ VP.sem$
VP	$\rightarrow V_t\ NP$	$V_t.sem @ NP.sem$
VP	$\rightarrow V_i$	$V_i.sem$
NP	$\rightarrow DET\ NN$	$DET.sem @ NN.sem$
NP	$\rightarrow NNP$	$\lambda P.P(NNP.sem)$
DET	$\rightarrow a$	$\lambda P.\lambda Q.\exists xP(x) \wedge Q(x)$
DET	$\rightarrow every$	$\lambda P.\lambda Q.\forall x(P(x) \Rightarrow Q(x))$
V_t	$\rightarrow likes$	$\lambda P.\lambda x.P(\lambda y.LIKES(x, y))$
V_i	$\rightarrow sleeps$	$\lambda x.SLEEPS(x)$
NN	$\rightarrow dog$	DOG
NNP	$\rightarrow Alex$	ALEX
NNP	$\rightarrow Brit$	BRIT



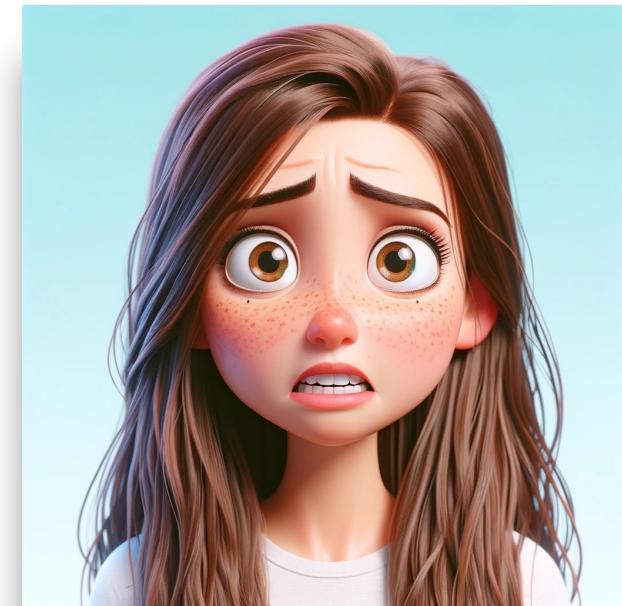
$$DET.sem @ NN.sem = (\lambda P.\lambda Q.\exists xP(x) \wedge Q(x)) @ DOG = \\ \lambda Q.\exists x DOG(x) \wedge Q(x)$$

$$NP.sem @ VP.sem = (\lambda Q.\exists x DOG(x) \wedge Q(x)) @ (\lambda z.SLEEPS(z)) = \\ \exists x DOG(x) \wedge (\lambda z.SLEEPS(z) @ x) = \\ \exists x DOG(x) \wedge SLEEPS(x)$$

Change variables to avoid ambiguity

$(\lambda A.A(x)) @ B$ and B contains a λ , do $B @ x$

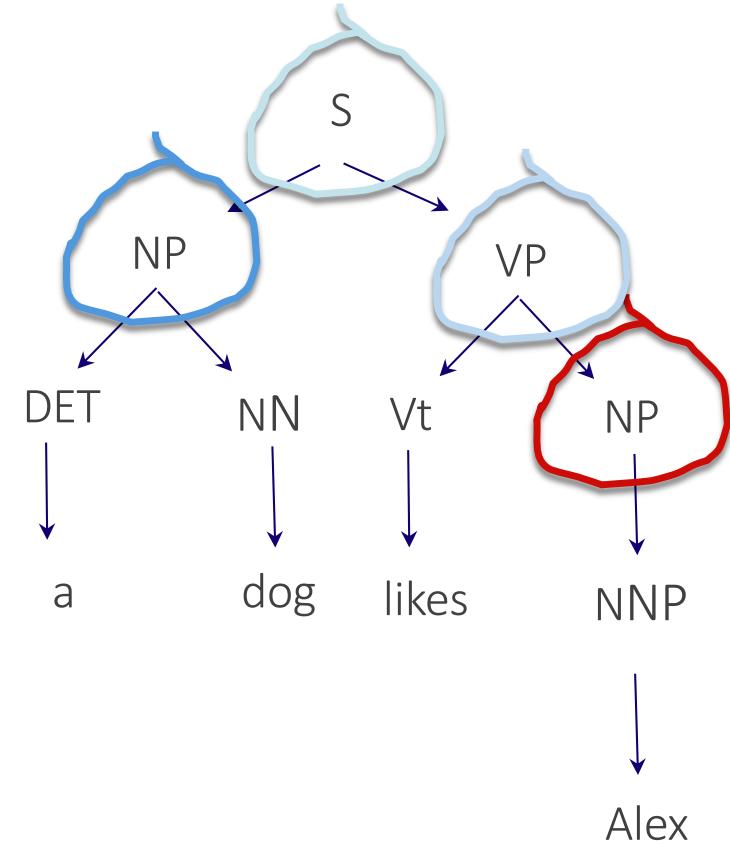
Hmm... Could you
please show another
example?



S	$\rightarrow NP \ VP$	$NP.sem @ VP.sem$
VP	$\rightarrow V_t \ NP$	$V_t.sem @ NP.sem$
VP	$\rightarrow V_i$	$V_i.sem$
NP	$\rightarrow DET \ NN$	$DET.sem @ NN.sem$
NP	$\rightarrow NNP$	$\lambda P.P(NNP.sem)$

DET	$\rightarrow a$	$\lambda P.\lambda Q.\exists xP(x) \wedge Q(x)$
DET	$\rightarrow every$	$\lambda P.\lambda Q.\forall x(P(x) \Rightarrow Q(x))$
V_t	$\rightarrow likes$	$\lambda P.\lambda x.P(\lambda y.LIKES(x,y))$
V_i	$\rightarrow sleeps$	$\lambda x.SLEEPS(x)$
NN	$\rightarrow dog$	DOG
NNP	$\rightarrow Alex$	ALEX
NNP	$\rightarrow Brit$	BRIT

- A dog likes Alex



(from previous exercise) $DET.sem @ NN.sem = \lambda Q.\exists x \text{DOG}(x) \wedge Q(x)$

$\lambda P.P(NNP.sem) = \lambda P.P(ALEX) \leftarrow \text{Stop! No @}$

$V_t.sem @ NP.sem = (\lambda P.\lambda x.P(\lambda y.LIKES(x,y))) @ \lambda Q.Q(ALEX) = \lambda x.(\underline{\lambda Q.Q(ALEX)} @ \lambda y.LIKES(x,y)) =$

$\lambda x.(\underline{\lambda y.LIKES(x,y)} @ ALEX) = \lambda x.LIKES(x, ALEX)$

$NP.sem @ VP.sem = (\lambda Q.\exists x \text{DOG}(x) \wedge Q(x)) @ \lambda y.LIKES(y, ALEX) =$

$\exists x \text{DOG}(x) \wedge (\underline{\lambda y.LIKES(y, ALEX)} @ x) = \exists x \text{DOG}(x) \wedge \text{LIKES}(x, ALEX)$

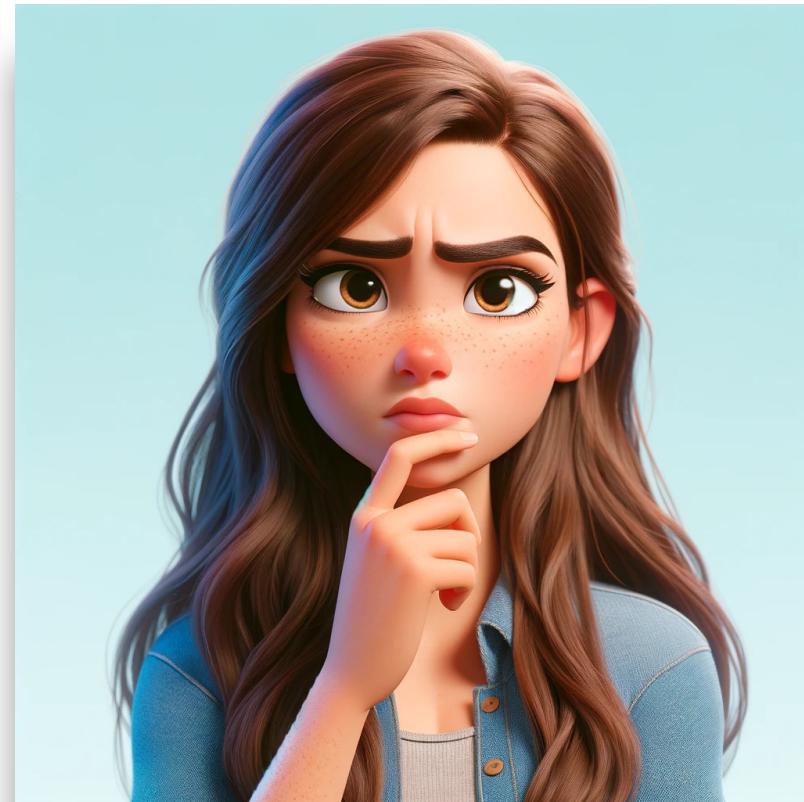
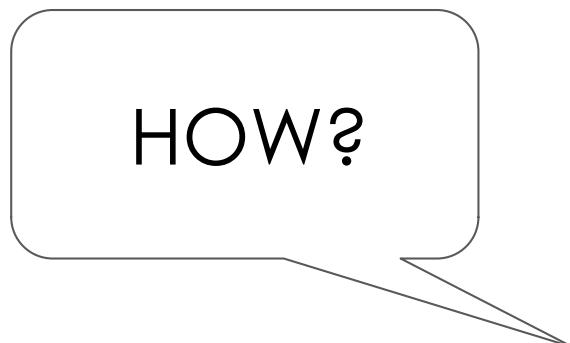
Change variables!

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - [Semantic Parsing as a sequence prediction problem](#)
 - Semantic Stuff
 - Semantic Relations
 - Semantic Resources
- Key takeaways
- Suggested readings

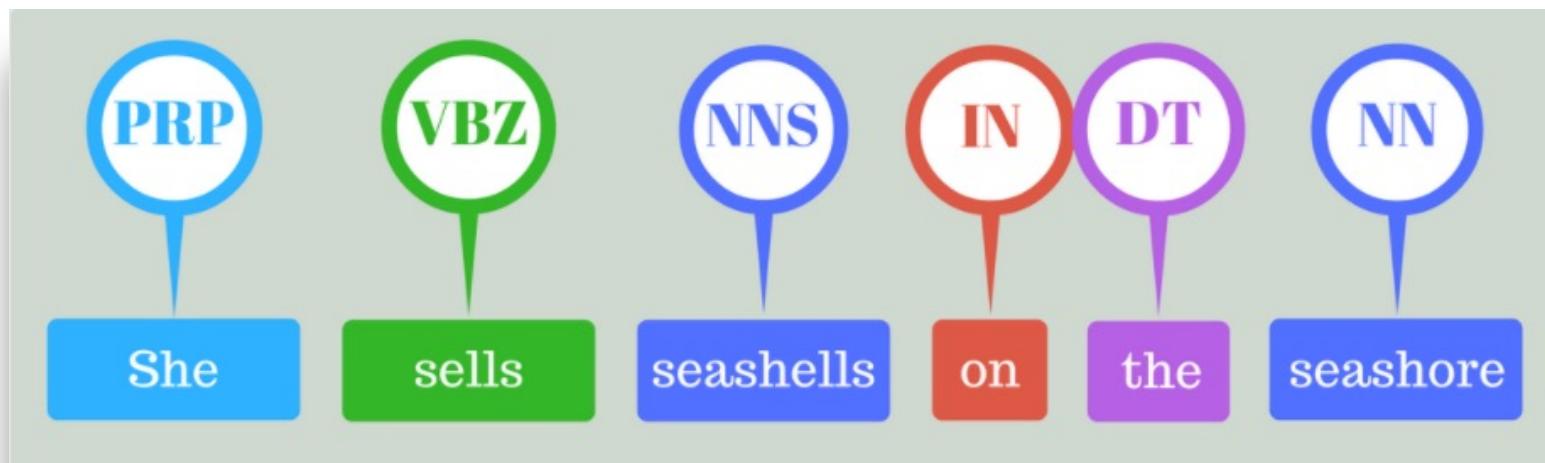
SEMANTIC PARSING AS SEQUENCE PREDICTION

- Semantic parsing can also be model as a sequence prediction task



SEQUENTIAL TASKS IN NLP

- Well...
 - POS-TAGGING was a sequence prediction task
 - Remember?



<https://nlpforhackers.io/wp-content/uploads/2016/08/Intro-POS-Tagging.png>

SEQUENTIAL TASKS IN NLP

- A new notation can work a miracle
 - BIO Tagging:
 - If we want to recognize a set of chunks (example: locations), we can use:
 - B for the word in the Beginning of the chunk (sometimes ignored)
 - I for words Inside the chunk
 - O for words Outside the chunk
 - Notice that there are many different flavours for this

SEQUENTIAL TASKS IN NLP

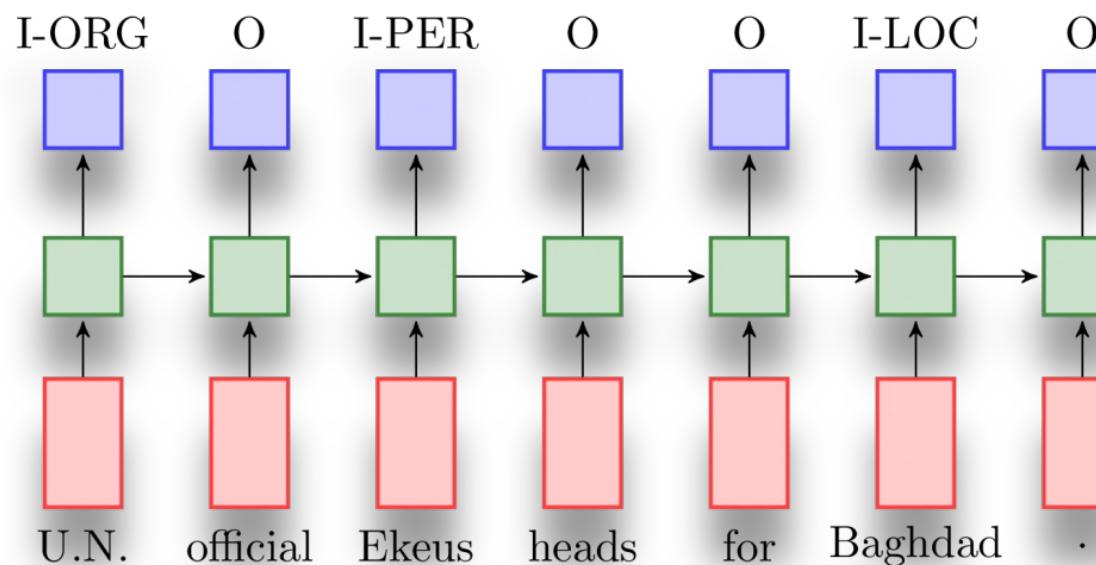
- Even **Syntactic Analysis** (remember?) can be done as a sequential task with the BIO notation

W e	s a w	t h e	y e l l o w	dog
PRP	VBD	DT	JJ	NN
B-NP	O	B-NP	I-NP	I-NP

<https://www.nltk.org/book/ch07.html>

SEQUENTIAL TASKS IN NLP

- The same for Named Entity Recognition



<https://www.depends-on-the-definition.com/guide-sequence-tagging-neural-networks-python/>

SEMANTIC PARSING AS SEQUENCE PREDICTION

- Now let us see how Semantic Parsing can also be model as a **sequence prediction task**

W	find	recent	comedies	by	james	cameron
S	↓	↓	↓	↓	↓	↓
O	O	B-date	B-genre	O	B-dir	I-dir
D	movies					
I	find_movie					

Figure 1: *An example utterance with annotations of semantic slots in IOB format (S), domain (D), and intent (I), B-dir and I-dir denote the director name.*

Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM
Dilek Hakkani-Tu ̈r, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen†, Jianfeng Gao, Li
Deng, and Ye-Yi Wang

SEMANTIC PARSING AS SEQUENCE PREDICTION

- Example:
 - Given
 - *How much is the cheapest flight from Boston to New York?*
 - How to create the frame (meaning representation):
 - Goal: Airfare <----- Intent (classification task, as in your project)
 - Cost Relative: cheapest
 - Depart_City: Boston
 - Arrival_City: New York

How much is the cheapest flight from Boston to New York?
O O O O B-Cost_Relative O O B-Depart_City O B-Arrival_City I-Arrival_City

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - [Semantic Relations](#)
 - Semantic Resources
- Key takeaways
- Suggested readings

SEMANTIC RELATIONS

- **Synonyms:** two words are synonyms if they have the same meaning (sense), regardless of context.
 - Example (EN):
 - Large and big (are these words really synonyms?)
 - Ok: How big/large is that plane?
 - Not ok: Joana is my big/large sister.
 - Example (PT):
 - Balofa vs. anafada
 - Bafio vs. mofo
 - ...

SEMANTIC RELATIONS

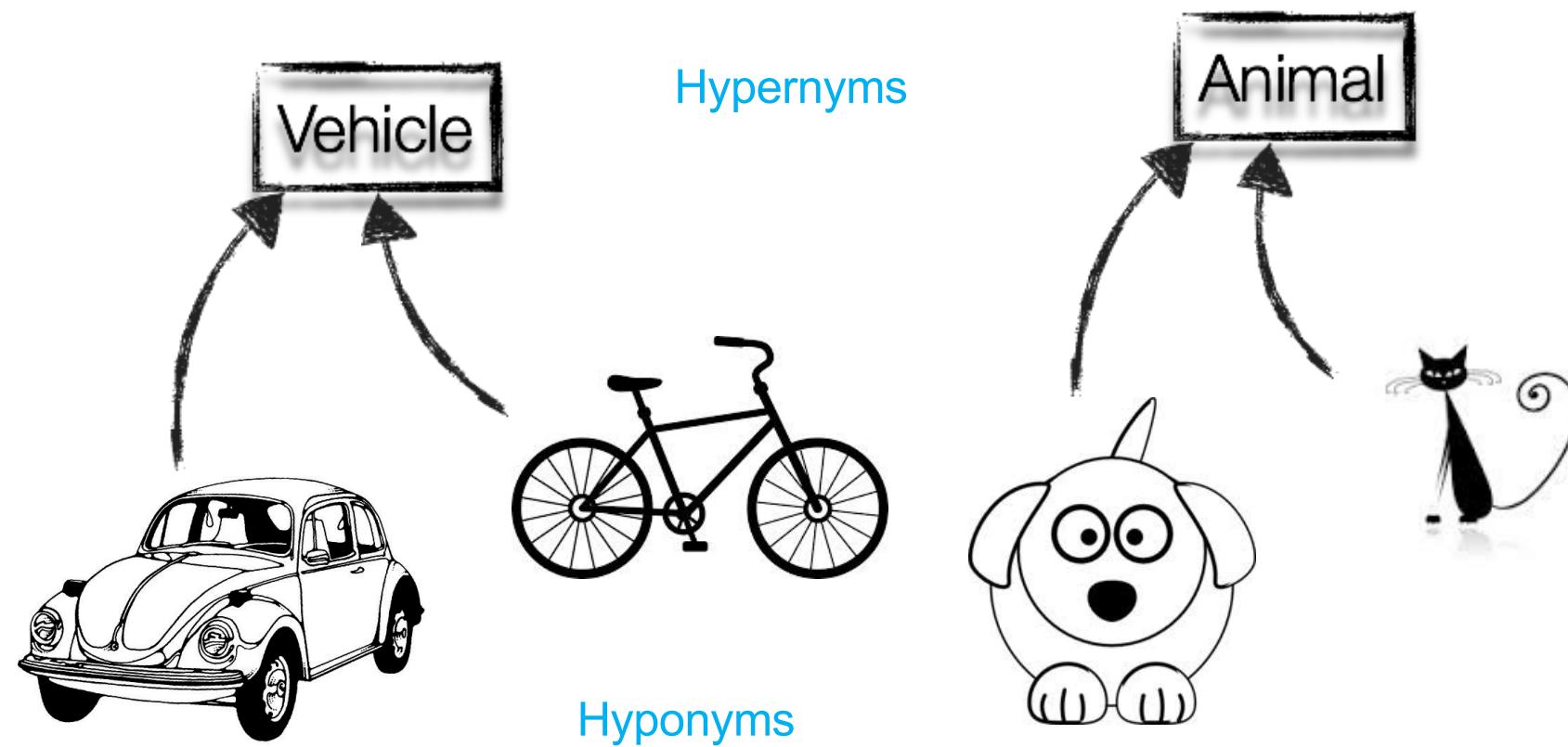
- **Antonyms:** two words are antonyms if they are opposites regarding one aspect of their meaning (being the rest fairly similar).
 - Example:
 - short vs. tall
 - rise vs. fall
 - cold vs. hot
 - ...

SEMANTIC RELATIONS

- **Hyponym:** a word is a hyponym (or subordinate)/hypernym (or superordinate) of another word if its meaning is more specific/general, respectively, than the meaning of the other word.
 - Example:
 - dog vs. animal (dog is a hyponym of animal)

SEMANTIC RELATIONS

- More examples:

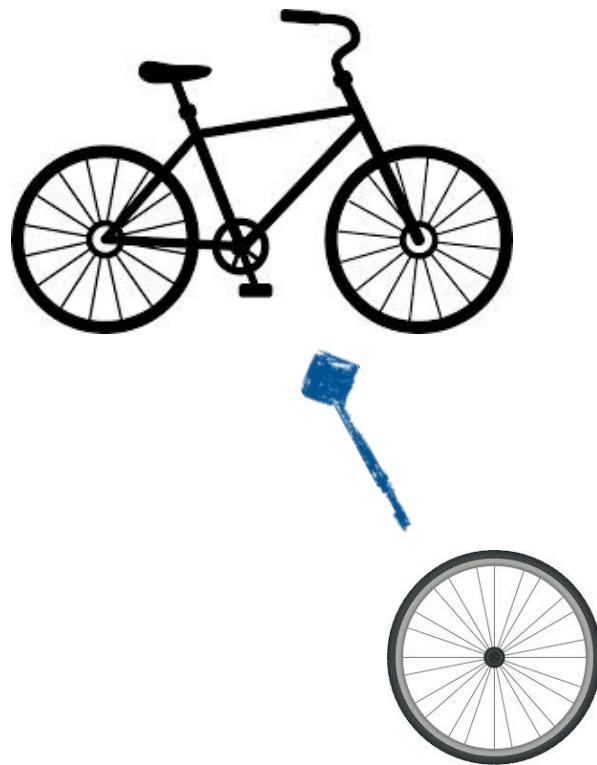


SEMANTIC RELATIONS

- **Meronym:** a word is a meronym (part-whole relation) of another word if its meaning corresponds to a part of the meaning of the other word.
 - Example:
 - wheel vs. car (wheel is a meronym of car; car is a holonym of wheel)

SEMANTIC RELATIONS

- Part of (meronymy of)



SEMANTIC RELATIONS

- **Homonyms:** two words are homonyms if they share the same pronunciation (homophones) AND the same spelling (homographs) but have unrelated meanings.
 - Examples:
 - homonyms – bank (to seat) vs. bank (financial institution); spell (a magical formula and to name or write the order of the letters in a word);
 - just homophones – concelho vs. conselho, riu vs. rio, write and right;
 - just homographs – almoço (nc) vs. almoço (v)

SEMANTIC RELATIONS

- **Polysemic:** a words is polysemic if it has various RELATED meanings.
 - Example:
 - banco (banco de dados vs. banco-cofre ou instituição financeira vs. edifício-banco)(bank – data; bank – finances)
 - dar (dar um livro, dar uma festa)(give – book; give – party)
 - ...

ACTIVE LEARNING MOMENT



EXERCISE

- What is the semantic relation between the following words?
 - couch/sofa
 - awake/asleep
 - stop/go
 - strong/weak
 - mammal/pig
 - get as obtain/ get as buy

EXERCISE

- What is the semantic relation between the following words?
 - couch/sofa: synonymy
 - awake/asleep: antonymy
 - stop/go: antonymy
 - strong/weak: antonymy
 - mammal/pig: hyponymy (pig is a hyponym of mammal)
 - get as obtain/ get as buy: polysemy

Overview

- Learning objectives
- Topics
 - Computational Semantics
 - Language Representation: symbolic approach
 - Compositional Semantics
 - Compositional Semantic Parsing
 - Semantic Parsing as a sequence prediction problem
 - Semantic Stuff
 - Semantic Relations
 - [Semantic Resources](#)
- Key takeaways
- Suggested readings

WORDNET (Fellbaum, 1998)

- Large lexical database for English
 - Set of lemmas annotated with a set of senses called synsets:
 - Instead of representing concepts in logical terms (or as vectors), concepts are represented by lists of words that can be used to express that concept
 - Each synset:
 - contains a brief definition (“gloss”) and, in most cases, one or more short sentences illustrating the use of the synset members,
 - is linked to other synsets by a semantic relation
- Check: <https://wordnet.princeton.edu>
 - Available (and also included in nltk)

WORDNET

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: <lexical filename > (gloss) "an example sentence"

Noun

- <noun.act>S: (n) **murder**, [slaying](#), [execution](#) (unlawful premeditated killing of a human being by a human being)

Verb

- <verb.social>S: (v) **murder**, [slay](#), [hit](#), [dispatch](#), [bump off](#), [off](#), [polish off](#), [remove](#) (kill intentionally and with premeditation) "*The mafia boss ordered his enemies murdered*"
- <verb.change>S: (v) [mangle](#), [mutilate](#), **murder** (alter so as to make unrecognizable) "*The tourists murdered the French language*"

Noun

- S: (n) animal, animate being, beast, brute, creature, fauna (a living organism characterized by voluntary movement)
 - direct hyponym / full hyponym
 - part meronym
 - S: (n) head, caput (the upper part of the human body or the front part of the body in animals; contains the face and brains) "he stuck his head out the window"
 - S: (n) face (the part of an animal corresponding to the human face)
 - member holonym
 - domain term category
 - substance meronym
 - direct hypernym / inherited hypernym / sister term
 - derivationally related form

Adjective

- S: (adj) animal, carnal, fleshly, sensual (marked by the appetites and passions of the body) "animal instincts"; "carnal knowledge"; "fleshly desire"; "a sensual delight in eating"; "music is the only sensual pleasure without vice"
 - similar to
 - S: (adj) physical (involving the body as distinguished from the mind or spirit) "physical exercise"; "physical suffering"; "was sloppy about everything but her physical appearance"
 - derivationally related form
 - antonym
 - W: (adj) mental [Indirect via physical] (involving the mind or an intellectual process) "mental images of happy times"; "mental calculations"; "in a terrible mental state"; "mental suffering"; "free from mental defects"

Proceedings of the Workshop on Multimodal Wordnets (MMWN-2020), pages 14–19
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

English WordNet 2020: Improving and Extending a WordNet for English using an Open-Source Methodology

John P. McCrae¹, Alexandre Rademaker², Ewa Rudnicka³, Francis Bond⁴

¹ Insight Centre for Data Analytics, NUI Galway, john@mccr.ae

² IBM Research and FGV/EMAp, alexrad@br.ibm.com

³ Wroclaw University of Science and Technology, ewa.rudnicka@pwr.edu.pl

⁴ Nanyang Technological University, bond@ieee.org

PROPBANK

- Corpus of text where predicate-argument relations were added to syntactic trees.
- Check:
<https://verbs.colorado.edu/~mpalmer/projects/ace.html>

PROPBANK

- [John]ARG0 broke [the window]ARG1
- [The window]ARG1 broke

ARG0	agent	ARG3	starting point, benefactive, attribute
ARG1	patient	ARG4	ending point
ARG2	instrument, benefactive, attribute	ARGM	modifier

Table 1.1: List of arguments in PropBank

FRAMENET

- Lexical database of English
- Idea: the meanings of words are defined by a semantic frame (the type of the event, relation, or entity and the participants in it).
- Check: <https://framenet.icsi.berkeley.edu/>
- Available (and also in nltk)

FRAMENET

- Example: Cooking involves:
 - a person doing the cooking (Cook),
 - the food that is to be cooked (Food),
 - something to hold the food while cooking (Container)
 - a source of heat (Heating_instrument).

Cooking_creation

[Lexical Unit Index](#)

Definition:

This frame describes food and meal preparation. A Cook creates a Produced_food from (raw) Ingredients. The Heating_Instrument and/or the Container may also be specified.

Caitlin BAKED some cookies from the pre-packaged dough.

FEs:

Core:

Cook [Cook]

Semantic Type: Sentient

The Cook prepares the Produced_food.

Drew COOKED dinner for his friends.

Drew BAKED an apple pie.

Produced_food [Food]

The Produced_food is the result of a Cook's efforts.

Drew PREPARED dinner for his friends.

Drew BAKED an apple pie for dessert.

Non-Core:

Container [Container]

Semantic Type: Container

This FE identifies the Container that holds the food being produced.

BAKE the quiche in a pie tin.

Things that apply the heat directly are Heating_Instruments, e.g. crock-pot, electric griddle.

FRAMENET

- With FrameNet we can conclude that:
 - "John sold a car to Mary" essentially describes the same basic situation (semantic frame) as "Mary bought a car from John"

OTHER RESOURCES

- VerbNet
- BabelNet
- ...

KEY TAKEAWAYS

KEY TAKEAWAYS

- There are two main ways to represent language: via a symbolic framework and via vectors
- Computational Semantics is a key area in NLP and compositional semantics is part of it
- There are several semantic relations between words and between constituents of a sentence
- Along the years there were several efforts towards building semantic resources

SUGGESTED READINGS

SUGGESTED READINGS

- Readings:
 - “Sebenta” 7.3 and 7.4
 - Jurafsky, on-line version, 3rd ed. draft, October 16, 2019:
chapter 19 (19.1-19.5), Word Senses and WordNet
 - Jurafsky, on-line version, 3rd ed. draft, October 16, 2019:
chapter 20 (20.1-20.5), Semantic Role Labelling



TRENDS: PRE-TRAINED MODELS, MULTI-TASK LEARNING, COMPRESSION TECHNIQUES AND APPLICATIONS

Luísa Coheur

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to define pre-trained models and explain different ways to leverage them, from simple Prompting to Transfer Learning.
 - In particular, students should be able to explain and apply Feature-based Transfer Learning and Fine-tuning.
- Additionally, students should be able to explain the concept of Multi-task Learning and how it can be conducted.
- Furthermore, students should be able to describe well-known compression techniques and NLP tasks.

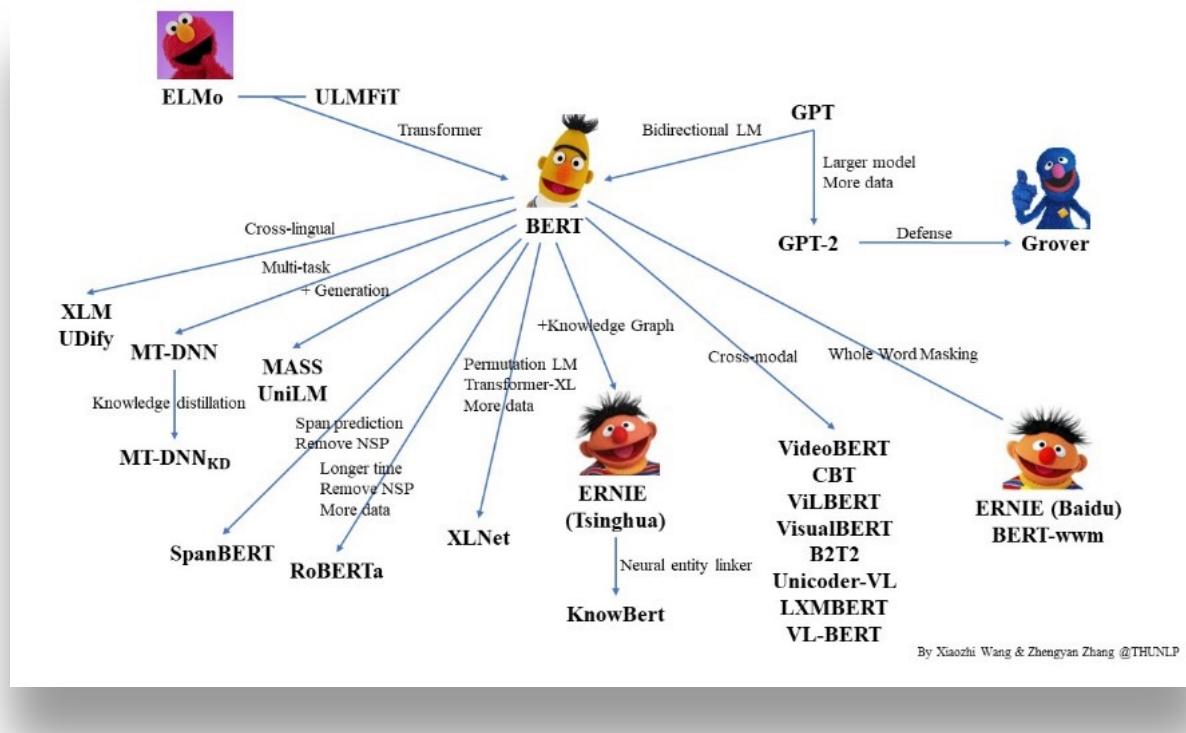
TOPICS

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

PRE-TRAINED MODELS

- Pre-trained models are machine learning models that have been previously trained on large datasets and saved for future use
 - Examples:
 - BERT and GPT (Generative Pre-trained Transformer)



PRE-TRAINED MODELS

- Pre-trained models are then used in different specific tasks
 - Idea:
 - You learn your native language. It take years, but finally, you level is excellent (pre-trained model). Future use: you find a job as a salesman in the pharmaceutical industry. You need to acquire new, specific vocabulary ← REMEMBER?

From your colleagues (2024):

<https://www.youtube.com/watch?v=spwSbuSG6c0>

BY THE WAY... ARE LLMs PRE-TRAINED MODELS?

- YES – an LLM is a type of pre-trained model
 - trained on massive amounts of text data with the objective of predicting the next token
- And NO – LLMs are more than just pre-trained models
 - They are also fine-tuned with techniques like Instruction Tuning or Reinforcement Learning from Human Feedback (RLHF), and they are ready to perform many tasks
 - They are usually used via prompting rather than retraining

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
 - key takeaways
 - Suggested readings

HOW TO USE PRE-TRAINED MODELS (DIRECT USE)

- The direct use is usually via **inference** – the internal process the model uses to generate an output from a given input
 - For instance, using BERT to extract representations
- Notice that:
 - Prompts – the external act of providing a Natural language input (the “prompt”) to the model – is usually not applied to simple pre-trained models (just to LLMs)

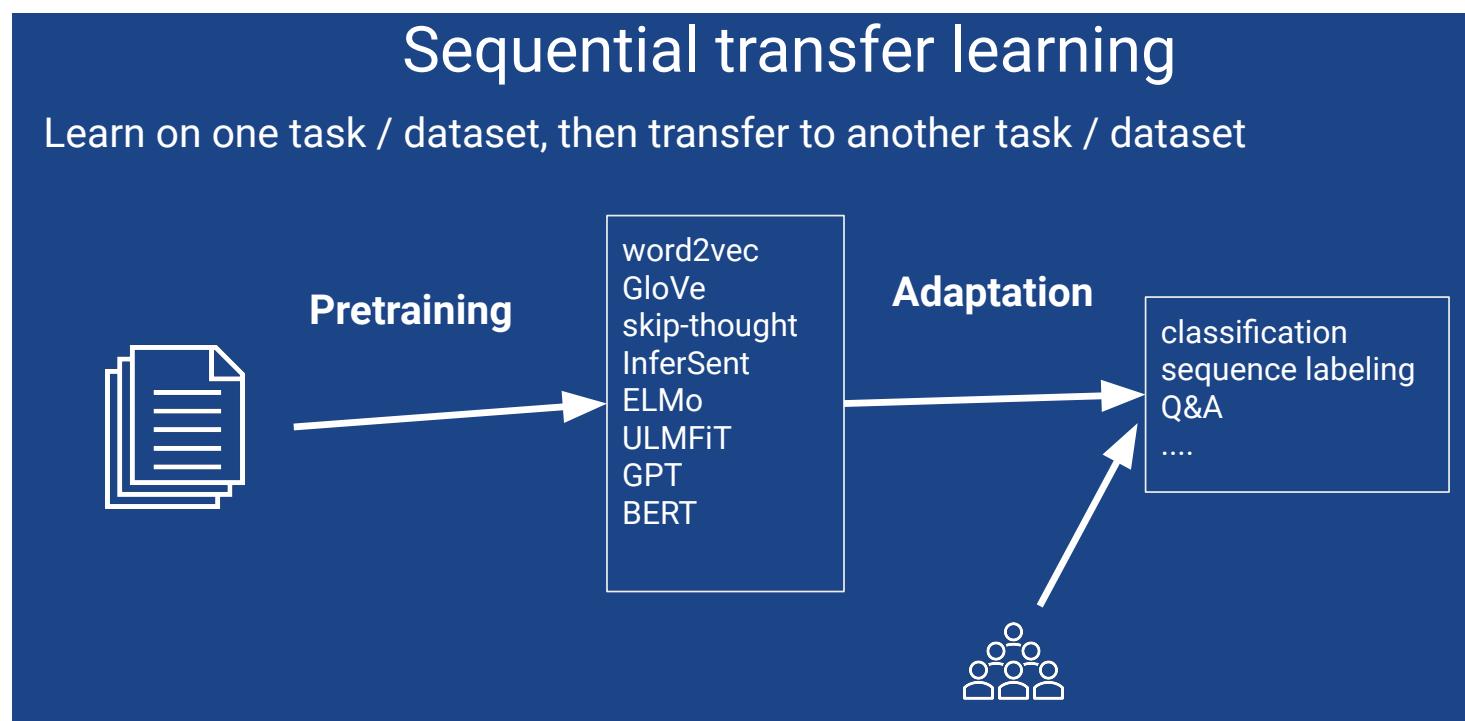
We will talk about prompt engineering in the next class

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

TRANSFER LEARNING

- Transfer learning is a machine learning technique in which a pre-trained model, developed for a particular task, is reused as the starting point for a model on a second task



TRANSFER LEARNING

- Some types of Transfer Learning
 - Feature-Based Transfer Learning: the features learned from the source domain/task are used as a starting point for the target domain/task
 - Fine-tuning: the pre-trained model's weights are adjusted (the model is re-trained on a variety of concrete examples) to better fit the target domain/task data

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
 - key takeaways
 - Suggested readings

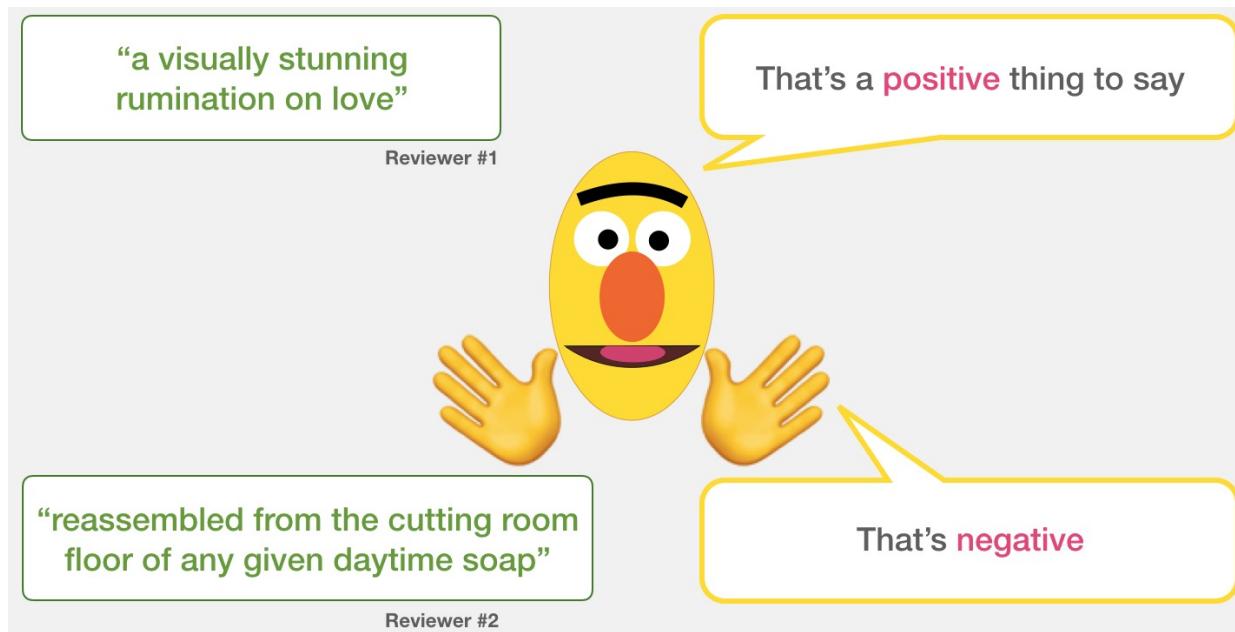
FEATURE-BASED TRANSFER LEARNING

- The next slides are from from “A Visual Guide to USING BERT for the First Time” (by Jay Alammar)



FEATURE-BASED TRANSFER LEARNING

Sentences (data from the “Stanford Sentiment Treebank”)	label
a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s horror films	1
apparently reassembled from the cutting room floor of any given daytime soap	0
they presume their audience won't sit still for a sociology lesson	0
this is a visually stunning rumination on love , memory , history and the war between art and commerce	1
jonathan parker 's bartleby should have been the be all end all of the modern office anomie films	1



Sentiment Analysis is an NLP TASK! (we already saw this)

FEATURE-BASED TRANSFER LEARNING

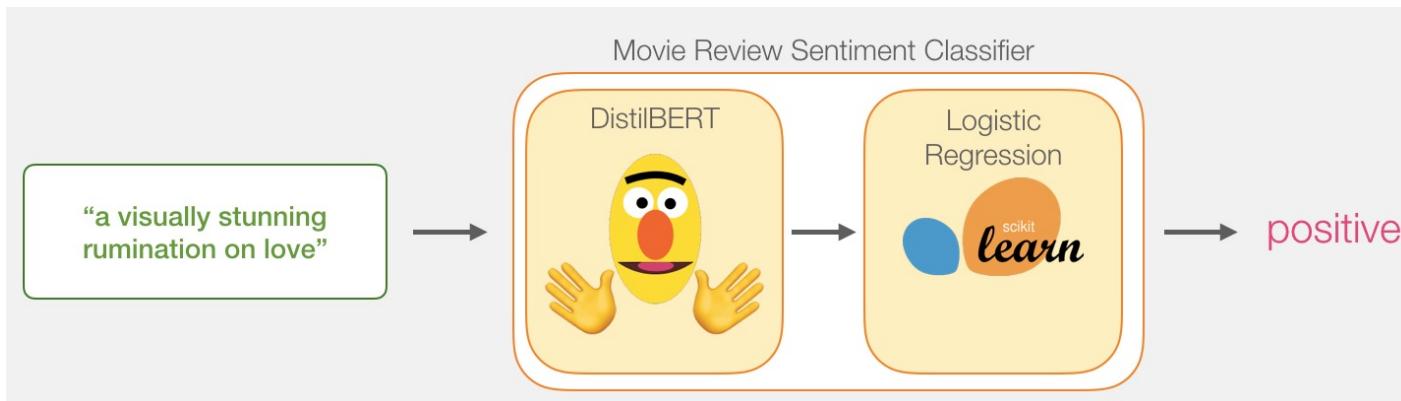
CLASSIC APPROACH



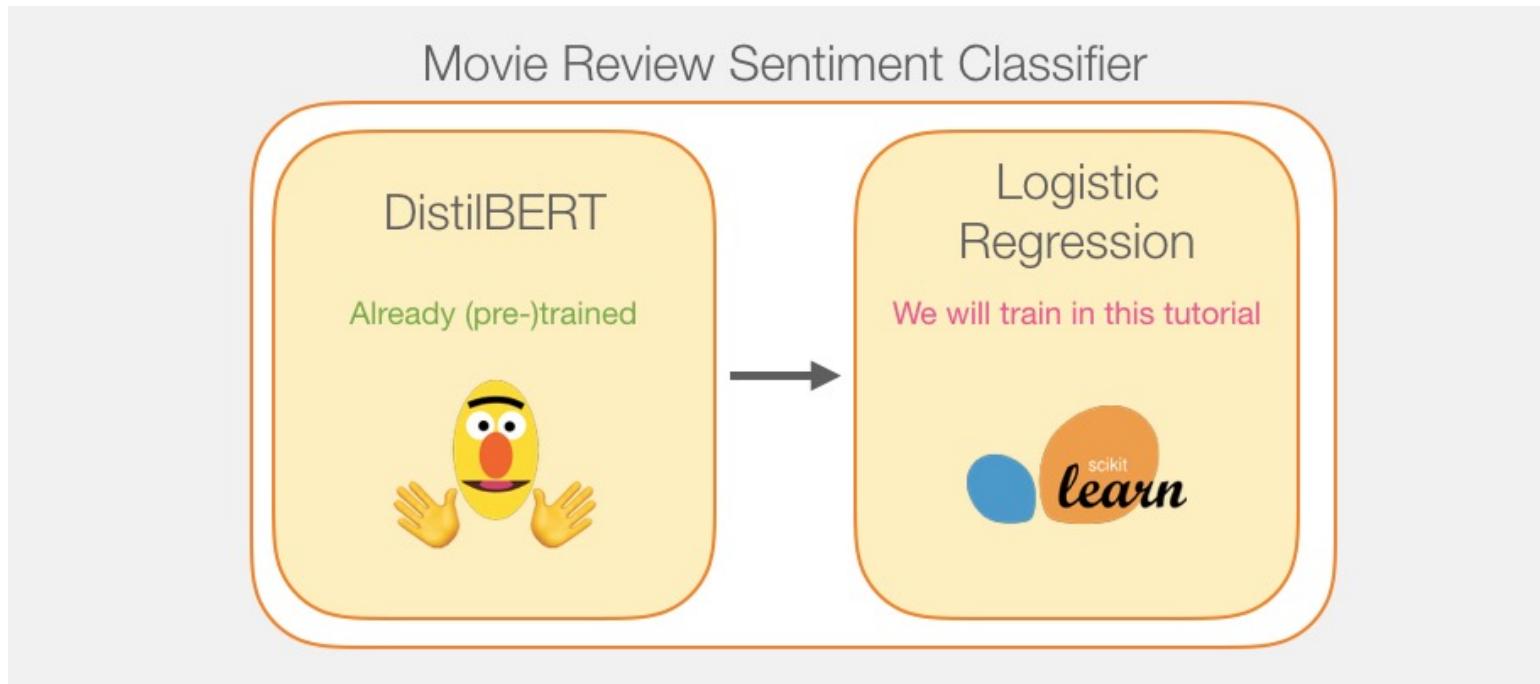
Example:
LOGISTIC REGRESSION
(or other classifier)

...

NOW

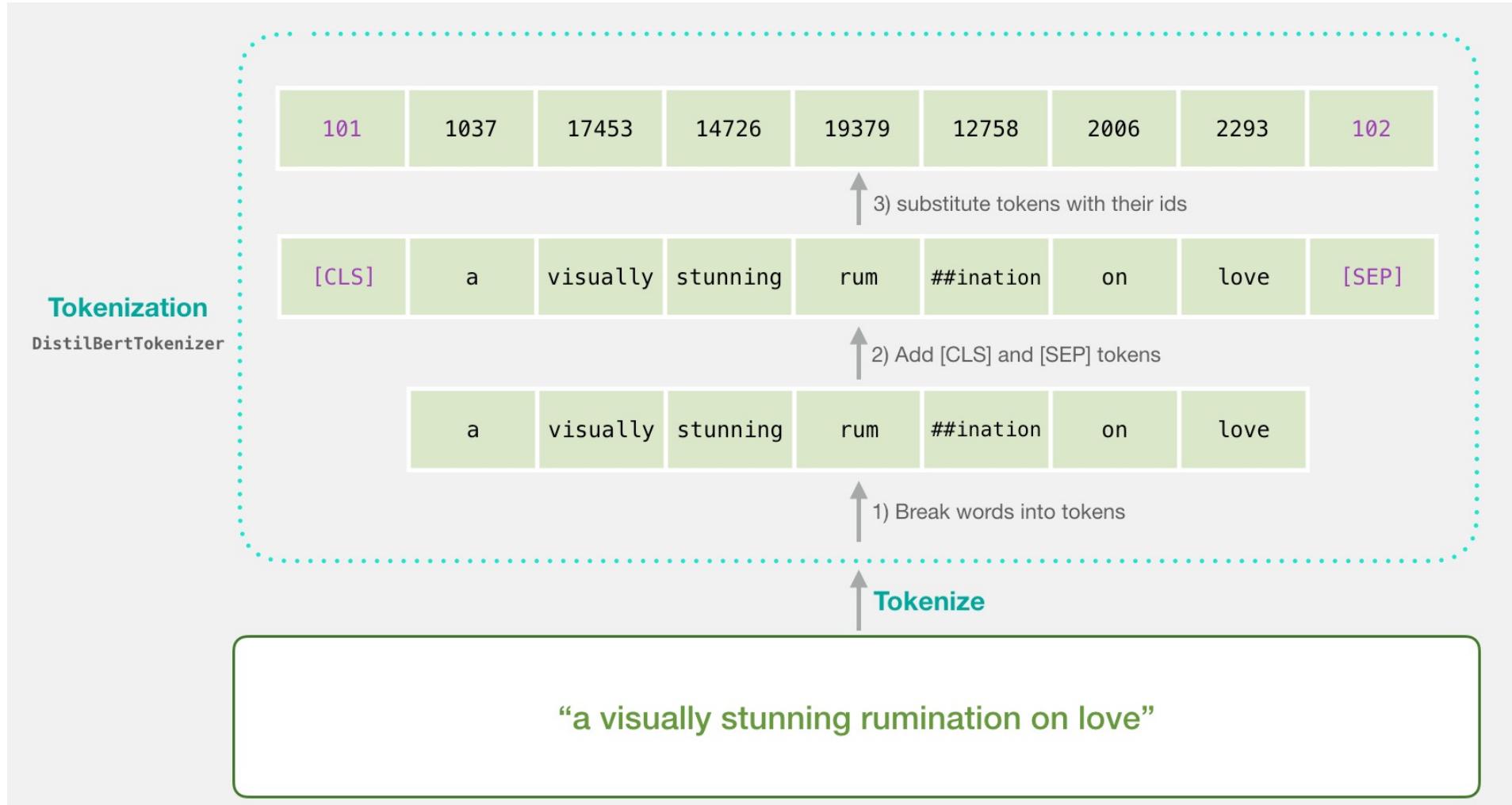


FEATURE-BASED TRANSFER LEARNING

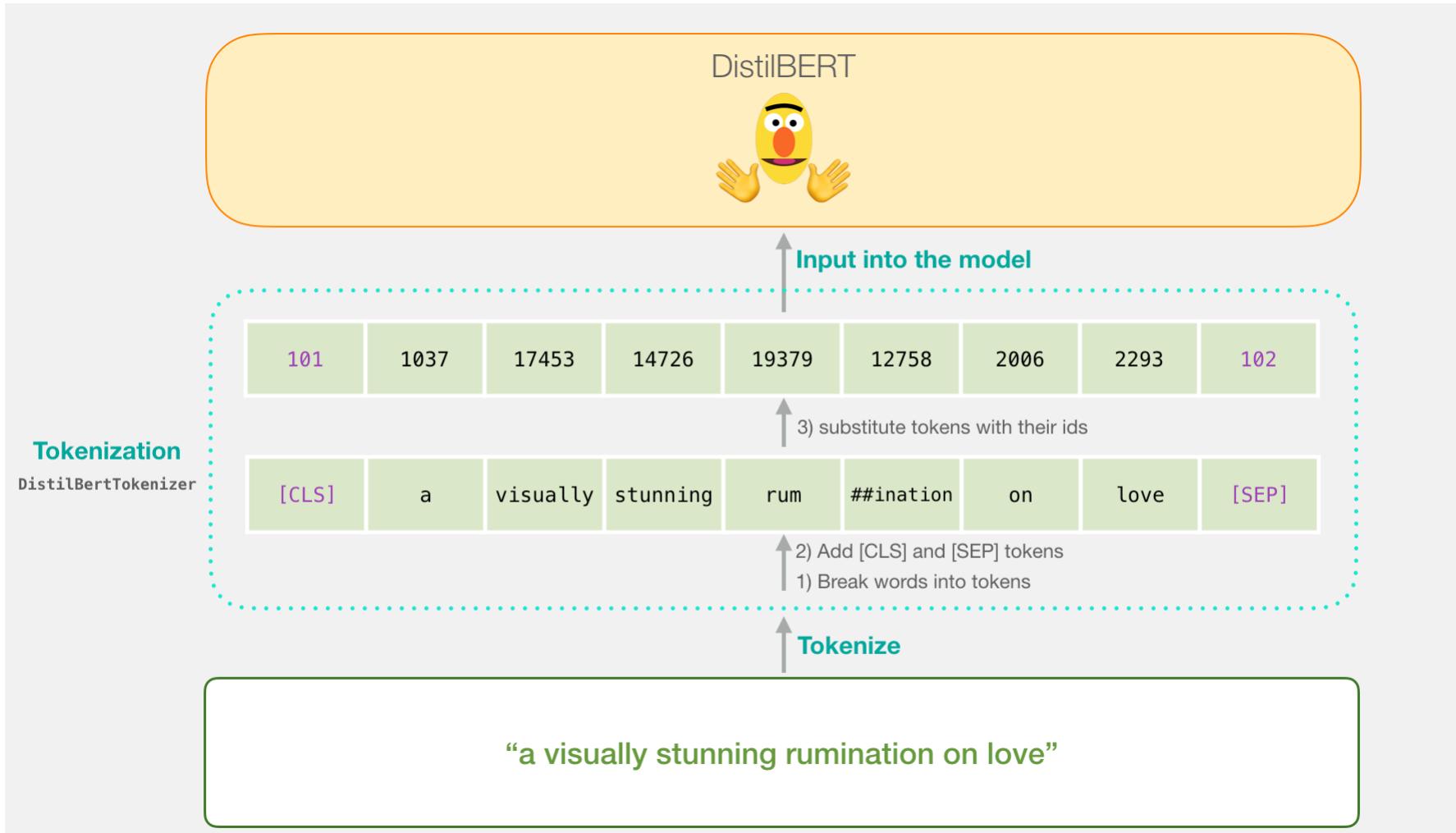


Note that DistilBERT is trained on the English language, but it was not trained to perform sentence classification.

FEATURE-BASED TRANSFER LEARNING

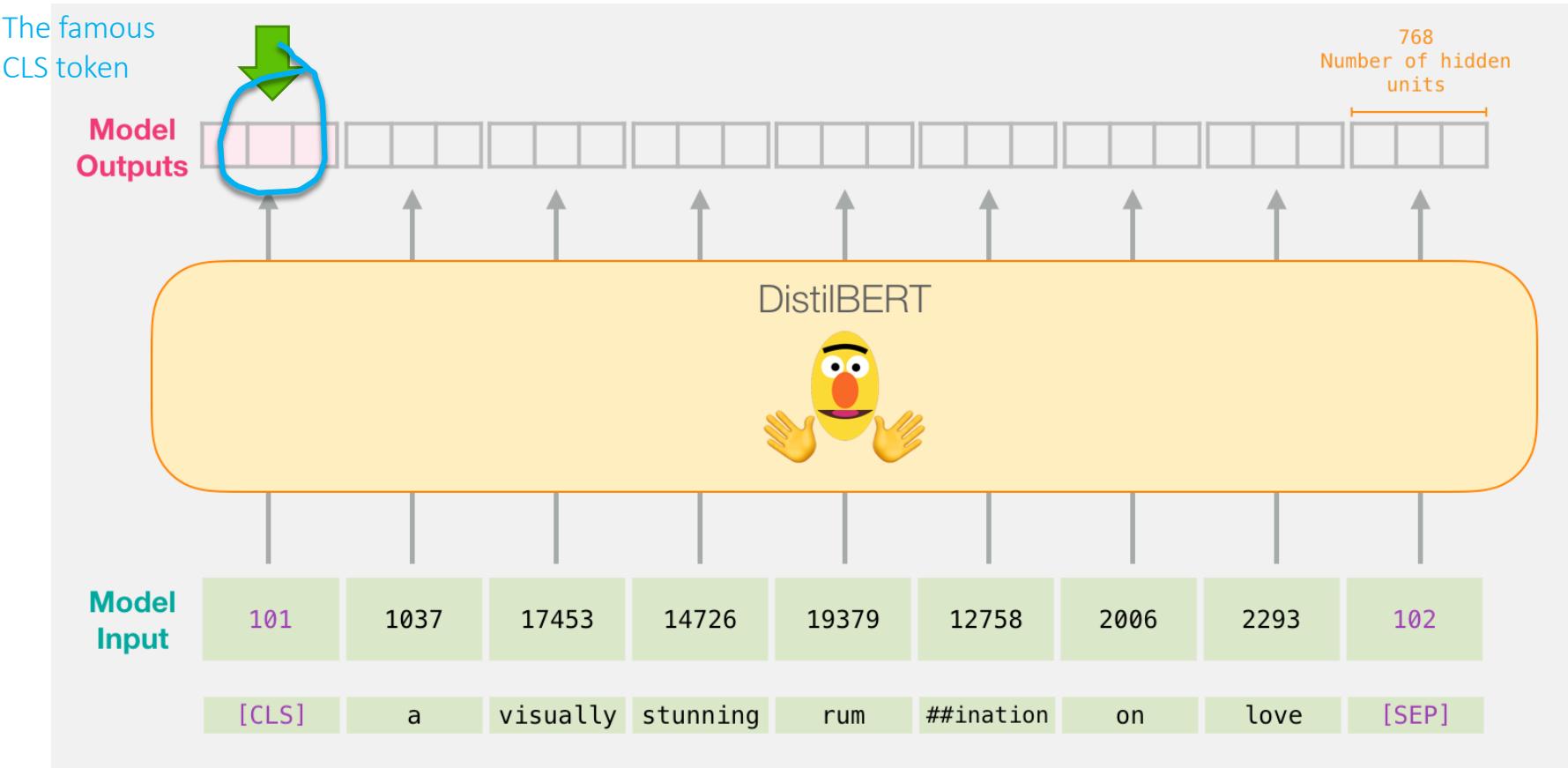


FEATURE-BASED TRANSFER LEARNING

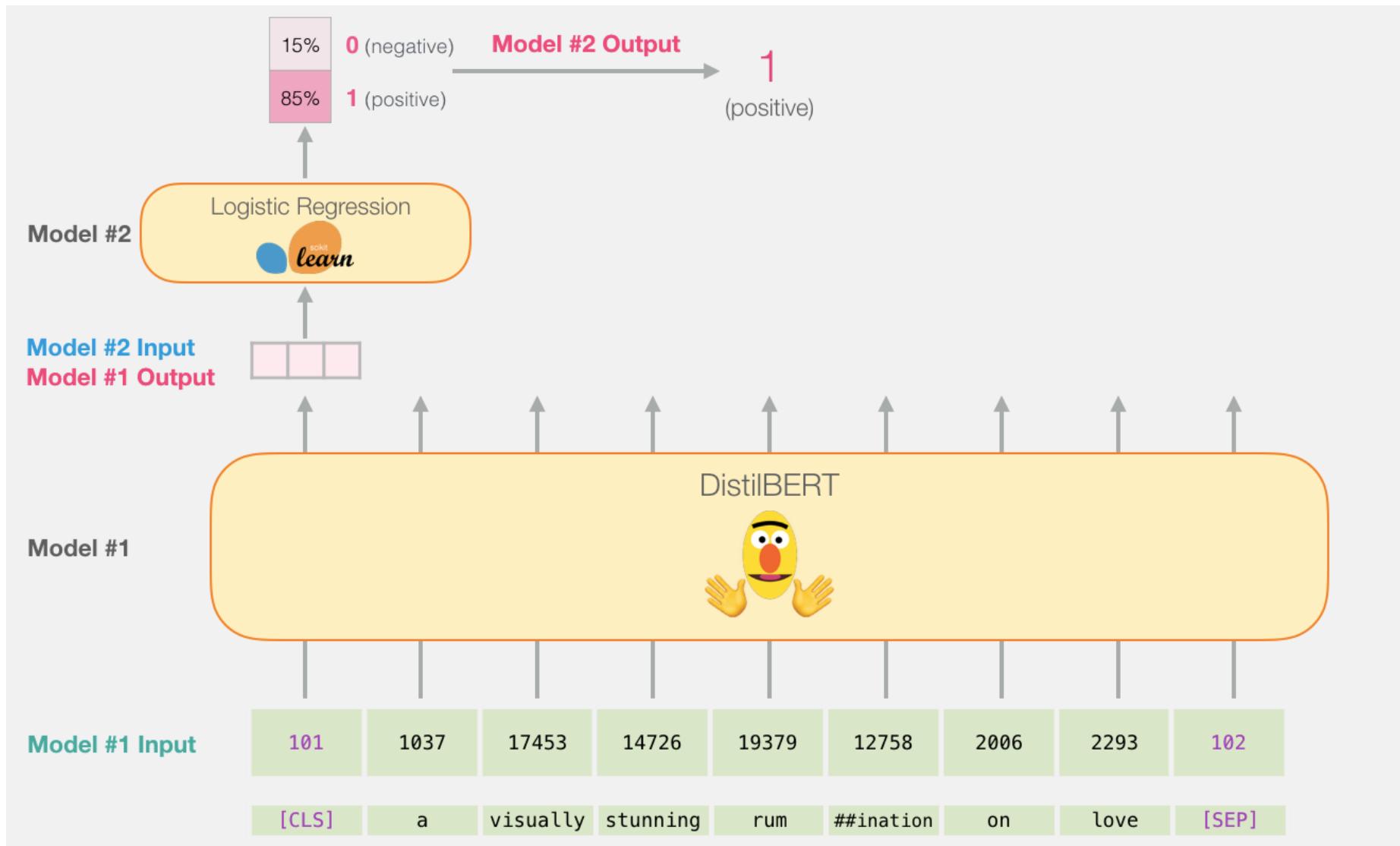


Note: Each token has an ID. Each ID is associated with a raw in an embedding matrix in BERT

FEATURE-BASED TRANSFER LEARNING



FEATURE-BASED TRANSFER LEARNING



OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - [Fine-tuning](#)
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

FINE-TUNING

- Fine-tuning is specific form of [transfer learning](#) in which the parameters of a model are adjusted to fit with specific observations
 - Involves taking a pre-trained model and continuing the training process on a new, typically smaller, dataset (usually, new layer(s) is(are) added to fit the targets specified in the new domain/task).

FINE-TUNING TECHNIQUES

- We can:
 - Train/freeze the entire pre-trained model and/or just train some layers while freezing others
 - Freeze in the beginning some layers and then train the whole architecture
 - ...

FINE-TUNING TECHNIQUES

- In theory, you will not need much data, and it will not be (super) expensive...
- But:
 - fine-tuning these models for a particular task can be (very) costly, both in terms of time and computational resources



- Parameter-efficient fine-tuning methods are needed!

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

PARAMETER-EFFICIENT FINE-TUNING METHODS

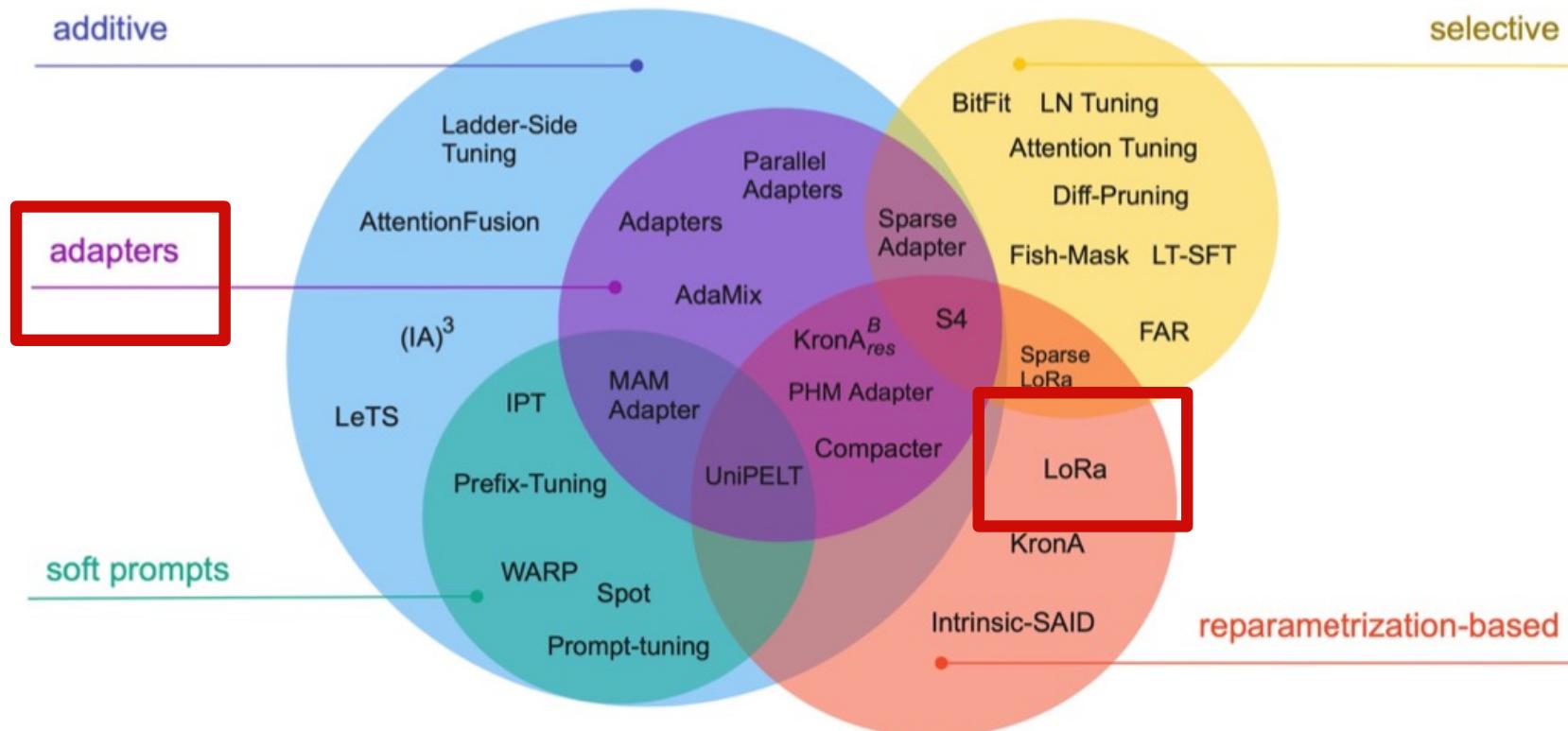
- Parameter-efficient fine-tuning (**PEFT**) focuses on fine-tuning pre-trained models efficiently by **training only a small number of extra parameters**

Small number
of *****EXTRA*****
parameters?
Hum...



PARAMETER-EFFICIENT FINE-TUNING METHODS

- There is a world of parameter-efficient fine-tuning techniques!



ADAPTERS

- Adapters are small, trainable layers inserted into a pre-trained model's architecture
 - These layers are designed to capture task-specific information while keeping the pre-trained model's weights frozen
 - thus, no need for extensive retraining of the entire model

ADAPTERS

- How Adapters work:
 - Start with a pre-trained model that has learned general features from a large dataset
 - Insert adapter modules (weights randomly initialized) into specific layers of the pre-trained model
 - These adapters have a small number of parameters compared to the full model
 - Only the adapter layers are trained on the new task's dataset, while the rest of the model remains unchanged
 - This reduces the computational cost
 - During inference, the pre-trained model, along with the adapter layers, is used to make predictions on new data
 - They are “loaded” to augment the frozen base model

ADAPTERS

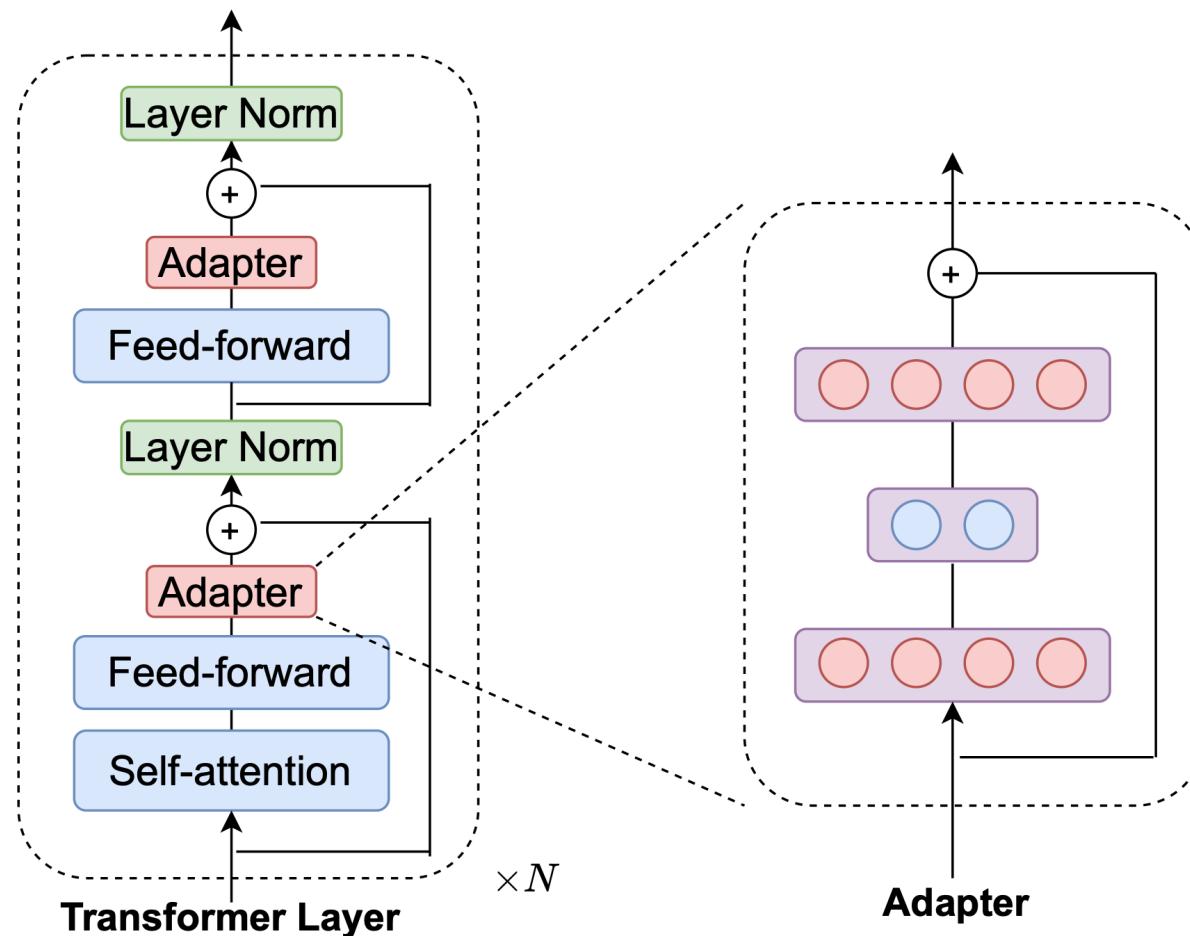


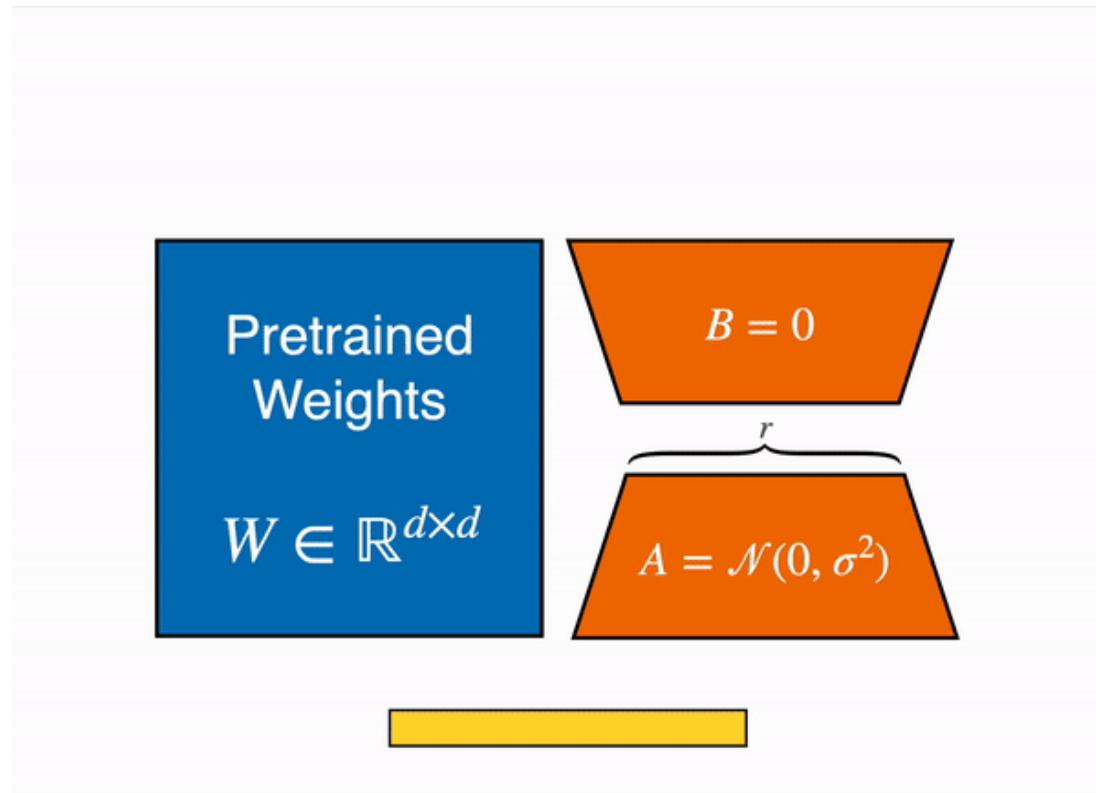
Image from <https://aclanthology.org/2021.acl-long.172/>

ADAPTERS

- In some scenarios, only 3% of task-specific parameters are needed to almost match the results of the 100% task-specific parameters used by the fully fine-tuned model
 - Sometimes adapter-based tuning outperforms fine-tuning on low-resource and cross-lingual tasks (he et al. 2021)

LoRA

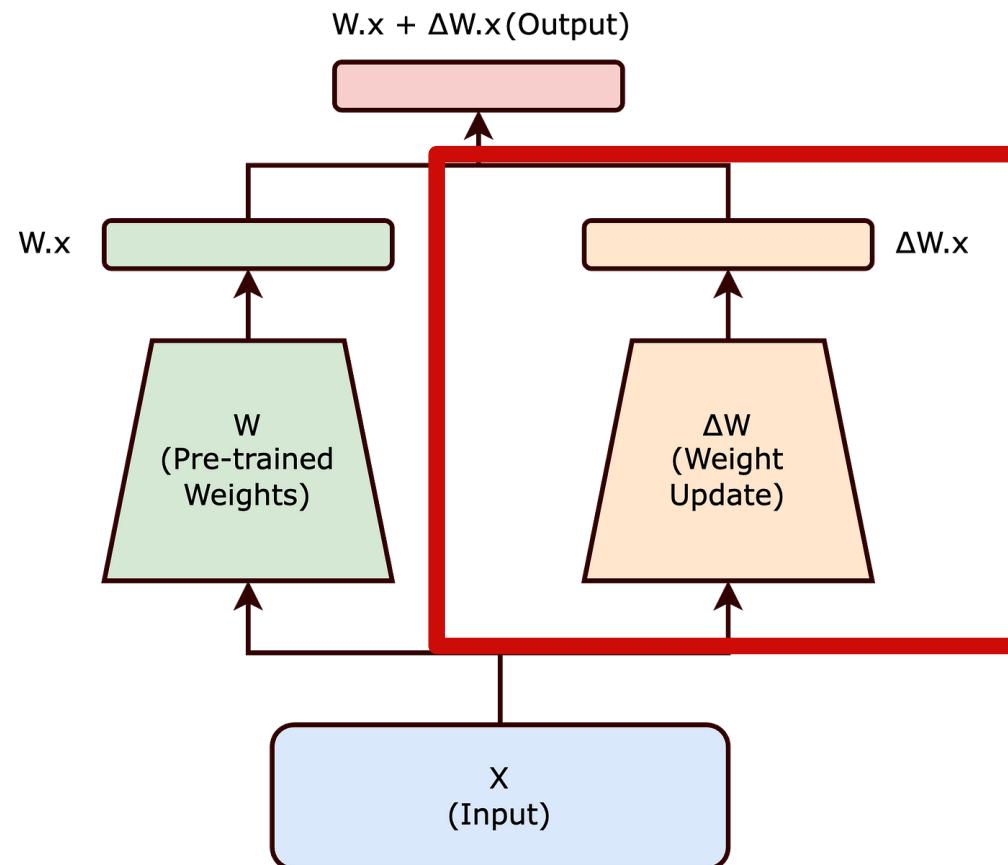
- Low-Rank Adaptation (LoRA) decomposes the model weight matrices using low-rank decomposition



From <https://medium.com/@manindersingh120996/understanding-low-rank-adaptation-lora-for-efficient-fine-tuning-of-large-language-models-082d223bb6db>

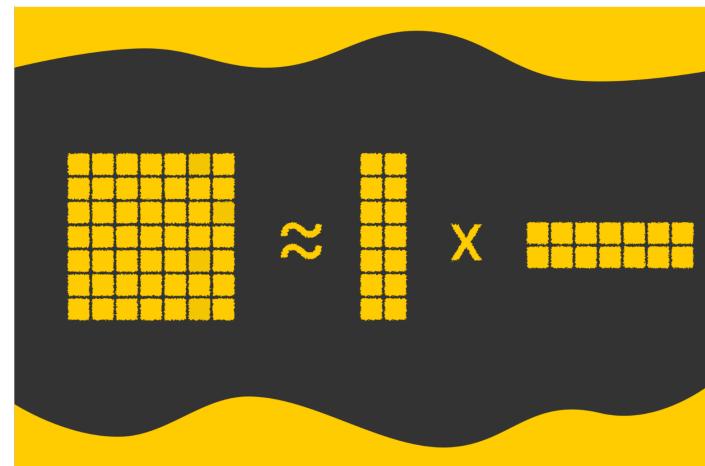
LoRA

- $h = Wx + \Delta Wx$
 - W are the model weights
 - ΔW its **accumulated gradient update during adaptation**



LoRA

- $h = Wx + \Delta Wx = Wx + BAx$
 - W are the model weights
 - ΔW its accumulated gradient update during adaptation
 - BAx are the LoRA weight changes



- Question: which should be A and B dimensions?

LoRA (SOME MATH)

- The concept of low rank
 - The rank of a matrix is given by the number of its linearly independent columns (or rows).
 - Note: It can be proven that the number of independent columns (known as column rank) is always equal to the number of independent rows (called row rank).
 - Example:

$$\text{rank} \left(\begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 0 \\ 3 & 0 & 4 \end{bmatrix} \right) = 3$$

LoRA (SOME MATH)

- But:

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix} = 1$$

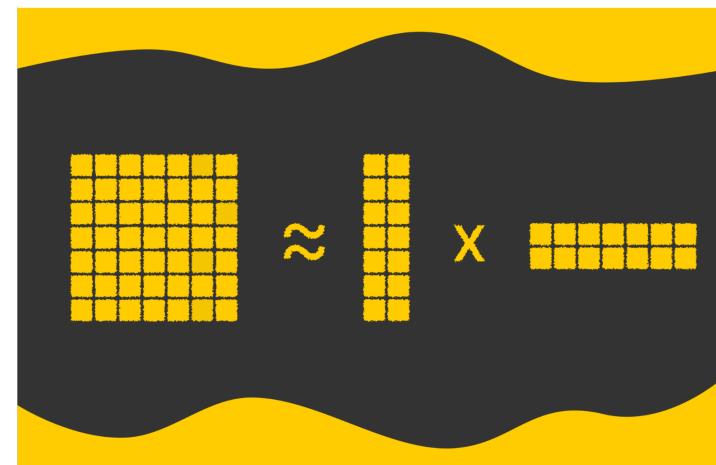
- the second row is just the first row multiplied by 2.
 - This means **the rows are not linearly independent**
 - The rank of this matrix is 1
 - So, this is a **low-rank matrix**

LoRA (SOME MATH)

- Rank decomposition of the matrix ΔW is the factorization of the form

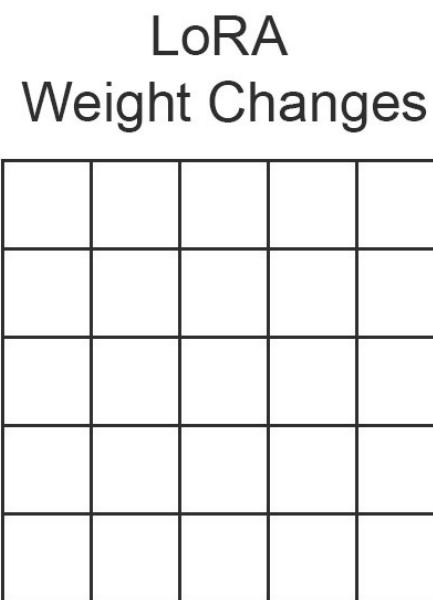
$$\Delta W = A B$$

where $\text{rank}(\Delta W) = r$

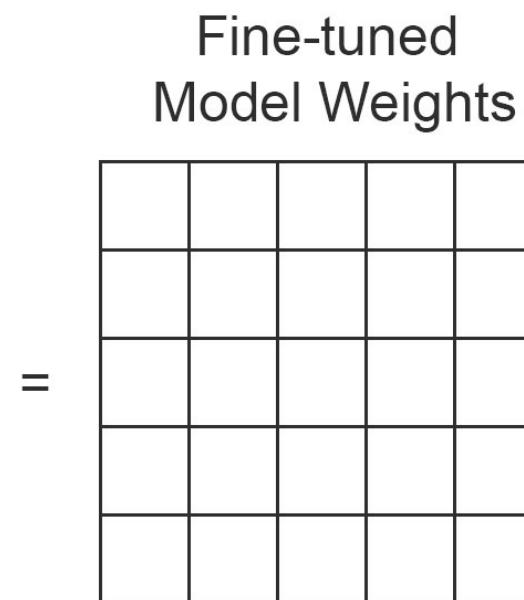


Note: It can be proven that every (finite) matrix has a rank decomposition. Techniques like SVD (Singular Value Decomposition – remember?) can be used to construct such a decomposition

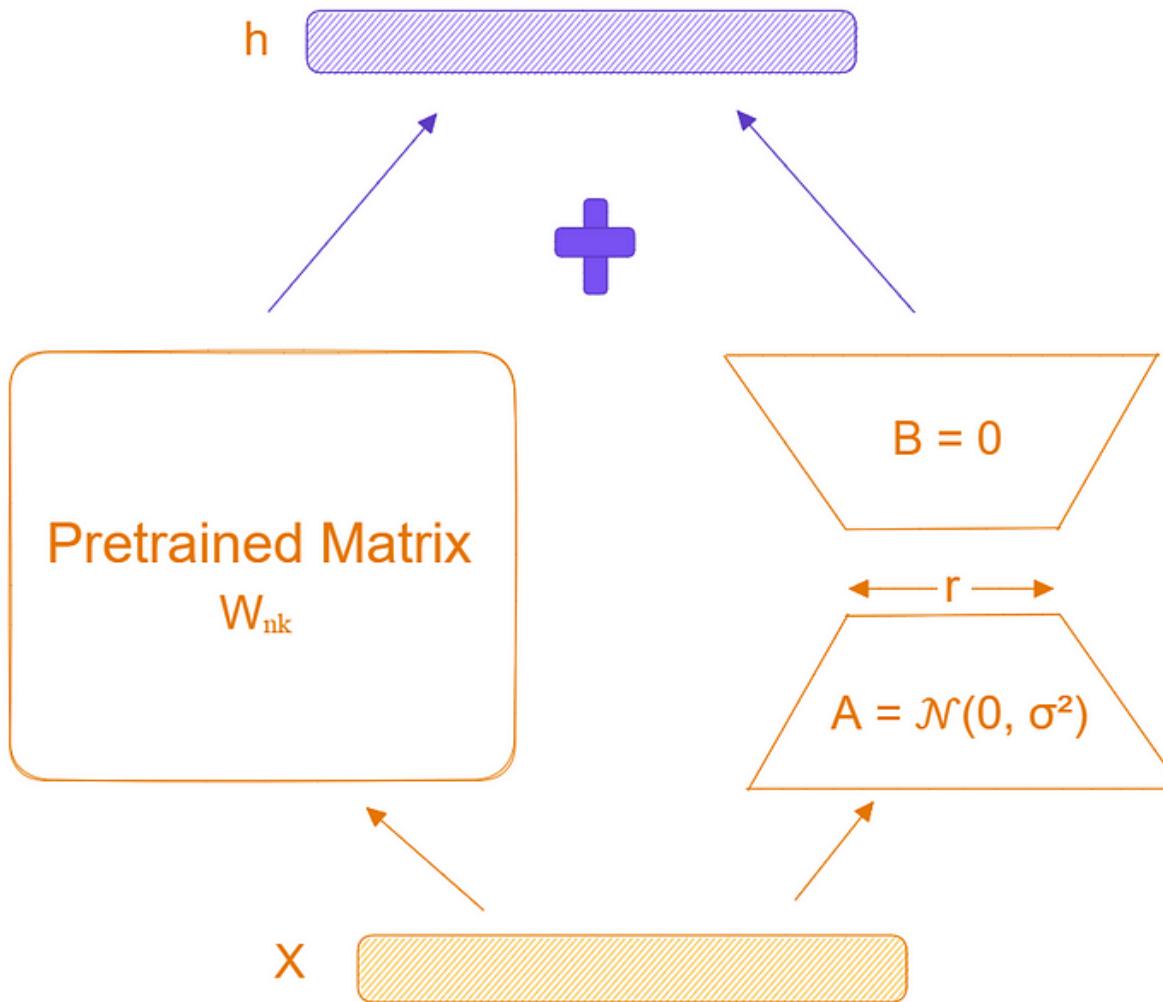
LoRA (MAIN IDEA)



+ Model Weights



LoRA (MAIN IDEA)



By the way:

- $A = N(0, \sigma^2)$ refers to a random matrix A that is initialized from a normal distribution (or Gaussian distribution) with a mean of 0 and a variance of σ^2 .
- $B = 0$ means that the matrix B is initialized to all zeros.

LoRA

- Why does this make sense?
 - Large models are trained to capture the general representation of their domain. These models capture a variety of features which allow them to be used for diverse tasks with reasonable zero-shot accuracy
 - However, when adapting such a model to a specific task or dataset, only a few features need to be emphasized or re-learnt. This means that the update matrix (ΔW) can be a low-rank matrix.

From <https://www.ml6.eu/blogpost/low-rank-adaptation-a-technical-deep-dive>

LoRA

- For a model like GPT-3, trainable parameters are reduced by 10000 times.

From your colleagues (LoRA and Fine-Tuning):

https://www.youtube.com/watch?v=rcMYq_c5bMg

<https://youtu.be/fRUPzzlEHU4>

<https://www.youtube.com/watch?v=liNX8JsBzxA>

Generated by DALL-E



OVERVIEW

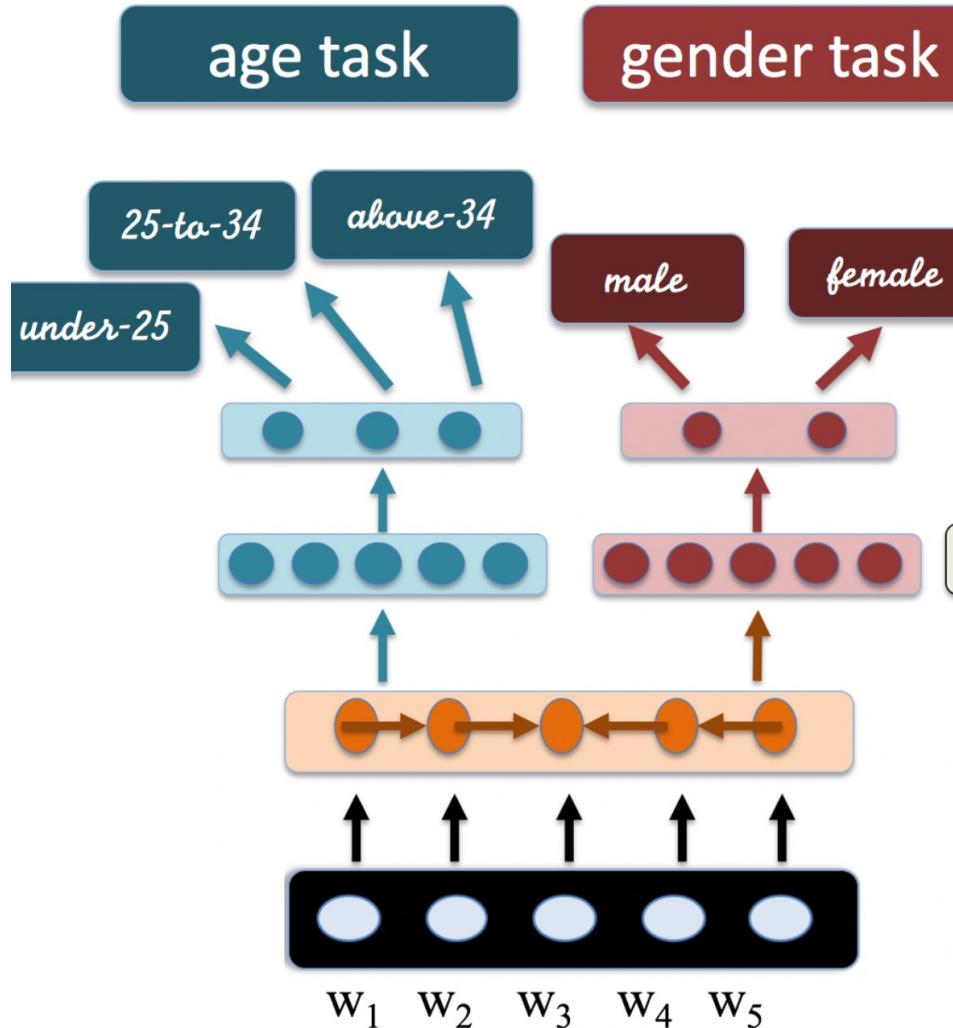
- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

MULTI-TASK LEARNING

- Multi-task learning is a subfield of machine learning in which multiple learning tasks are solved at the same time, leveraging the shared knowledge across these tasks to improve overall performance

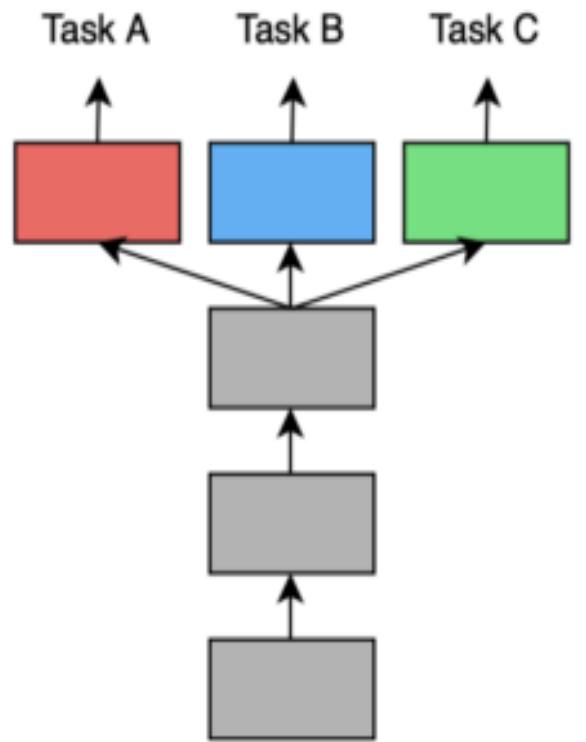
MULTI-TASK LEARNING

- Multi-task learning models:
 - learn a shared representation of the input data that captures the common features and patterns across the tasks
 - learn task-specific information. Each task has its own set of parameters that capture the unique characteristics of that task

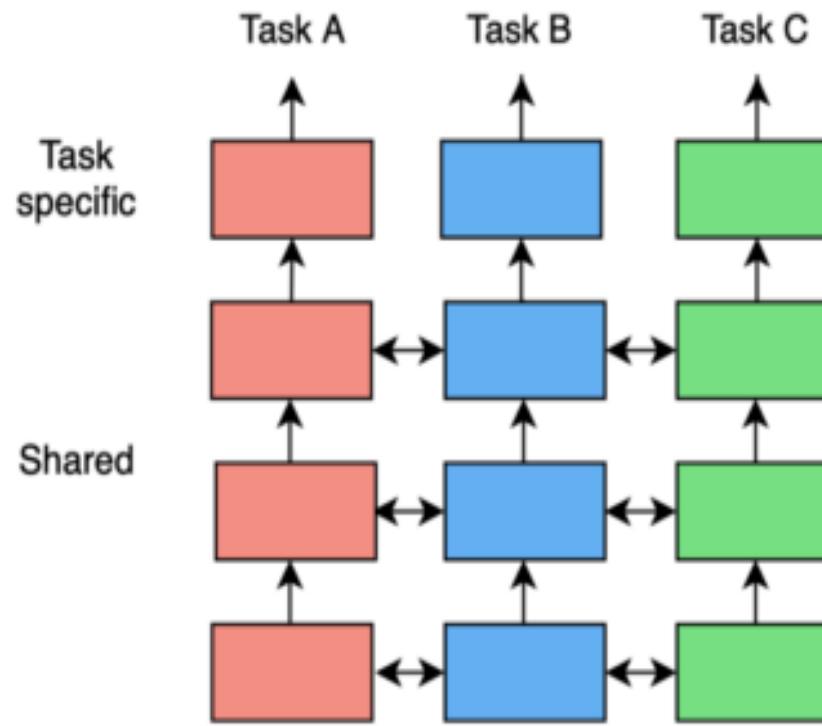


https://www.researchgate.net/publication/337019719_Sentence-Level_BERT_and_Multi-Task_Learning_of_Age_and_Gender_in_Social_Media

MULTI-TASK LEARNING



(a) Hard parameter sharing



(b) Soft parameter sharing

MULTI-TASK LEARNING

- As usual, besides soft/hard parameter sharing, there are many other approaches

EXAMPLE: DECANLP TASKS

- The Natural Language Decathlon (decaNLP) is a 10-task challenge:
 - Question Answering,
 - Machine Translation,
 - Summarization,
 - Natural Language Inference,
 - Sentiment Analysis,
 - Semantic Role Labeling,
 - Relation Extraction,
 - Goal-Oriented Dialogue,
 - Semantic Parsing,
 - Common sense Reasoning

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



THINK-PAIR-SHARE

- Any idea of how to train a model in all these tasks (with so many different inputs/outputs)?

Examples

<u>Question</u>	<u>Context</u>	<u>Answer</u>
What is a major importance of Southern California in relation to California and the US?	...Southern California is a major economic center for the state of California and the US....	major economic center
What is the translation from English to German?	Most of the planet is ocean water.	Der Großteil der Erde ist Meerwasser
What is the summary?	Harry Potter star Daniel Radcliffe gains access to a reported £320 million fortune...	Harry Potter star Daniel Radcliffe gets £320M fortune...
Hypothesis: Product and geography are what make cream skimming work. Entailment , neutral, or contradiction?	Premise: Conceptually cream skimming has two basic dimensions – product and geography.	Entailment
Is this sentence positive or negative?	A stirring, funny and finally transporting re-imagining of Beauty and the Beast and 1930s horror film.	positive



EXAMPLE: DECANLP TASKS

- Key Idea: one model, one format, ten tasks
 - DecaNLP reformulates all tasks as question answering:
 - Input = question + question
 - Output = answer text
 - The model just learns to “answer questions”, whether that question is about translation, sentiment, or inference.

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

COMPRESSION TECHNIQUES

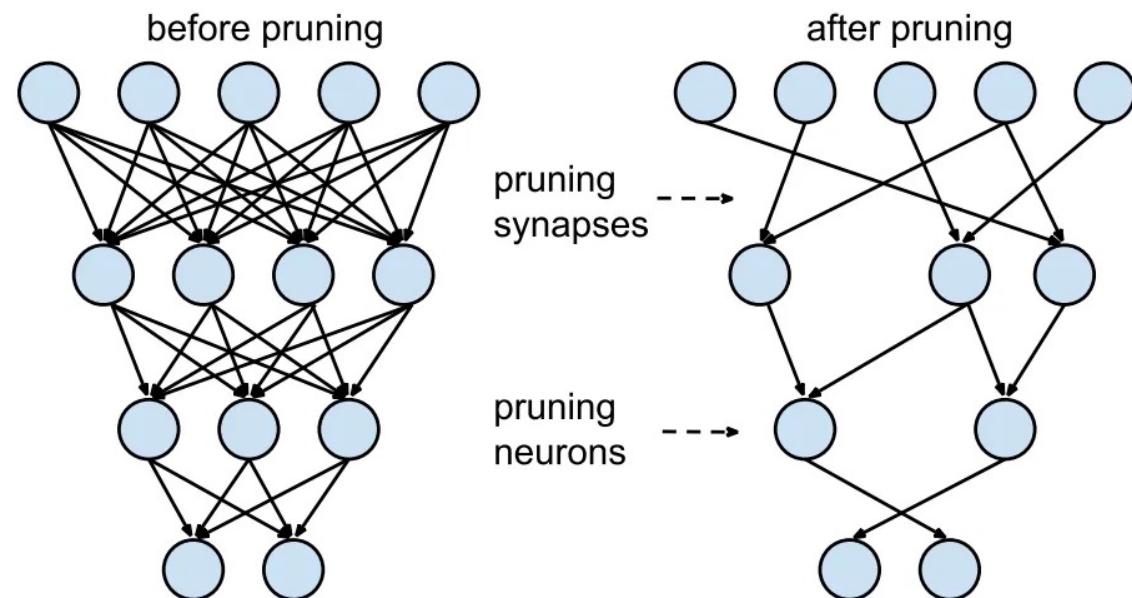
- Compression techniques are a set of methods used to reduce the size or complexity of neural network models without significantly sacrificing performance
 - Examples:
 - Pruning
 - Quantization
 - Teacher-student model (knowledge distillation)

PRUNING

- Pruning involves removing unnecessary weights or neurons from the neural network.

- Examples:

- Prune weights: prune connections that are below some predefined thresholds
- Prune neurons
- Prune layers
- ...



QUANTIZATION

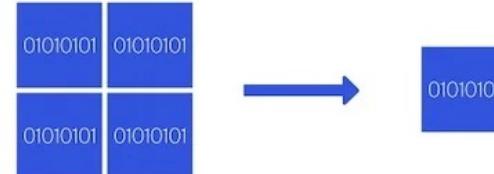
- Quantization reduces the precision of the weights and activations in the neural network, typically from 32-bit floating-point numbers to lower bit-width integers. This reduces the memory and computational requirements of the model

Quantization

Floating point Integer

3452.3194 → 3452

32 bit 8 bit



TEACHER-STUDENT MODEL (KNOWLEDGE DISTILLATION)

- Knowledge distillation involves training a smaller “student” model to mimic the predictions of a larger “teacher” model
 - In teacher-student training, the dataset provides hard targets (a single target label) and the teacher provides soft targets (a distribution over all labels – logits are used)

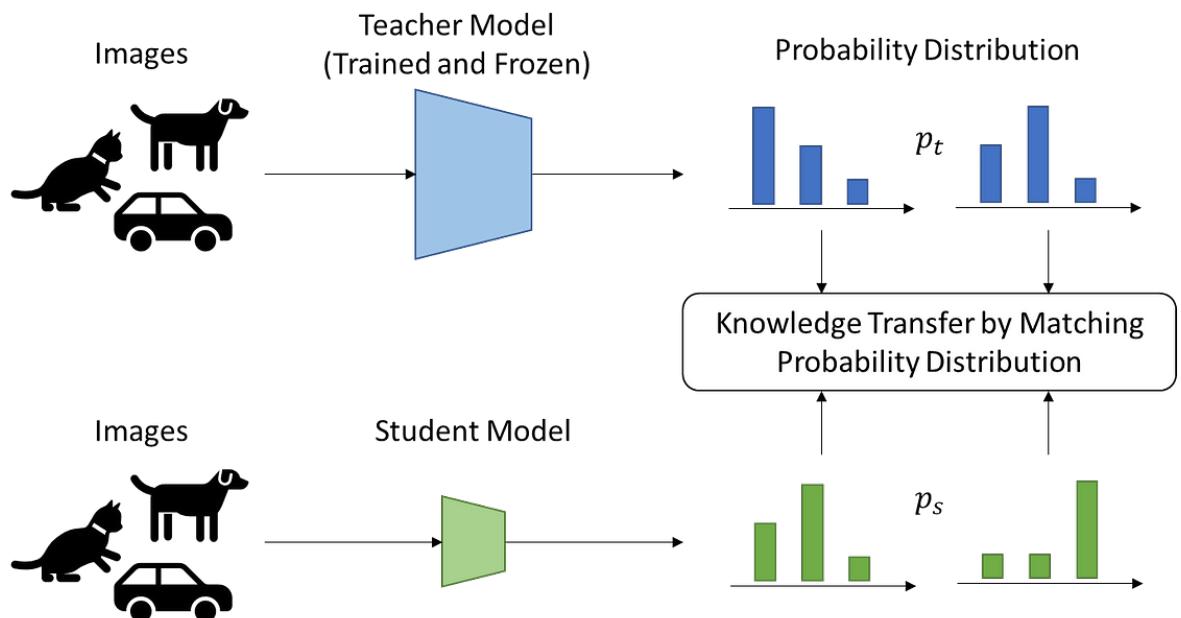
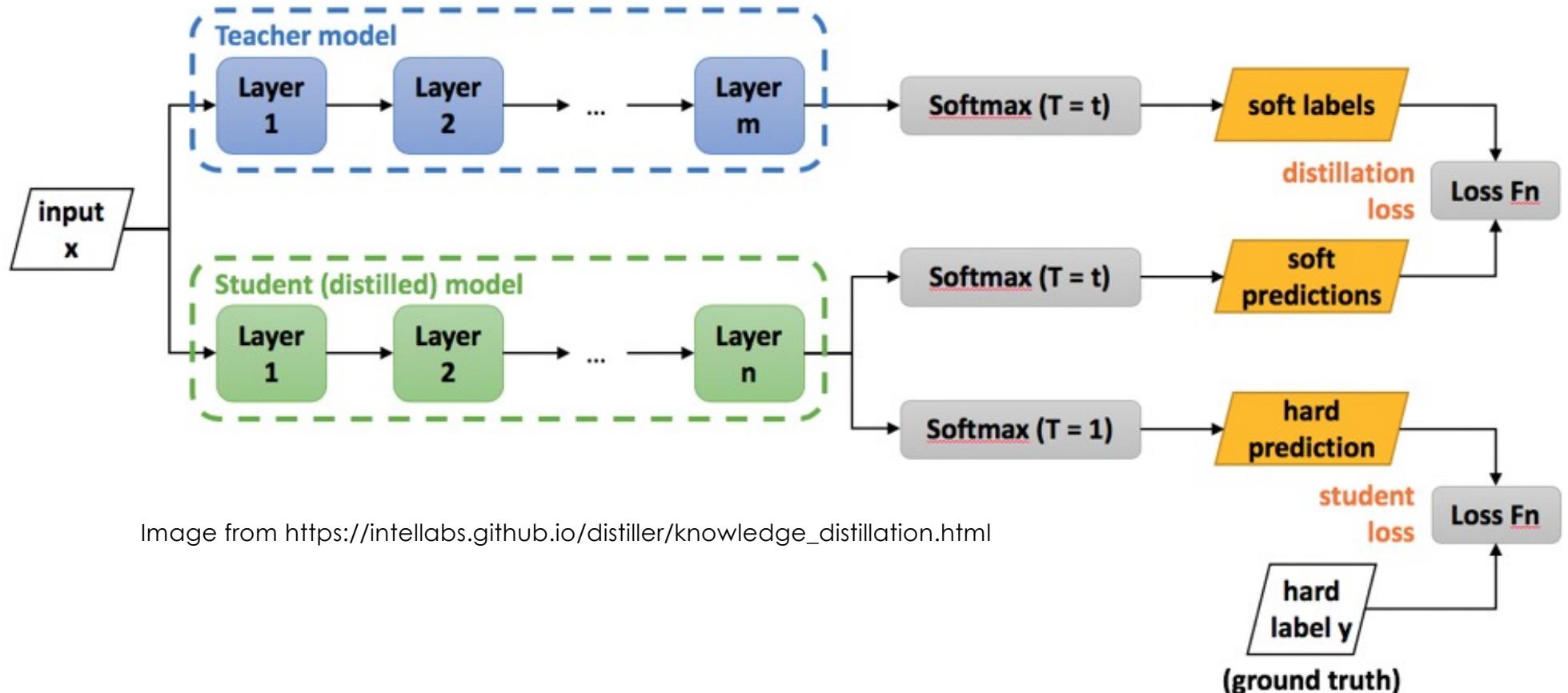


Image from <https://medium.com>

TEACHER-STUDENT MODEL (KNOWLEDGE DISTILLATION)



T for temperature: we will talk about this in the next class

OVERVIEW

- Learning objectives
- Topics
 - Pre-trained models
 - How to use pre-trained models
 - Direct Use
 - Transfer Learning
 - Feature-based Transfer Learning
 - Fine-tuning
 - Parameter-efficient fine-tuning methods
 - Multi-task learning
 - Compressing Techniques
 - Applications
- key takeaways
- Suggested readings

APPLICATIONS: QUESTION/ANSWERING

- Question Answering (QA): receive a question and a context that contains information necessary to output the desired answer ← this is a new definition (in the early days, no context was provided)
 - Widely used dataset: SQuAD

Passage Segment

...The European Parliament and the Council of the European Union have powers of amendment and veto during the legislative process...

Question

Which governing bodies have veto power?

APPLICATIONS: QUESTION/ANSWERING

- Interesting: this is an example of Span-based QA
 - we "only" need to find the beginning and the end of the answer

Passage Segment

...The European Parliament and the Council of the European Union have powers of amendment and veto during the legislative process...

Question

Which governing bodies have veto power?

APPLICATIONS: QUESTION/ANSWERING

- But:
 - QA could be very complicated!
 - From the shared task PÁGICO (LINGUAMÁTICA):
 - (PT) Quais os jogadores de futebol de língua portuguesa que passaram por mais de três países estrangeiros na sua vida profissional?
 - (EN) Which Portuguese-speaking football players have played in more than three foreign countries during their professional careers?

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

- How do you think QA was implemented 20 years ago?

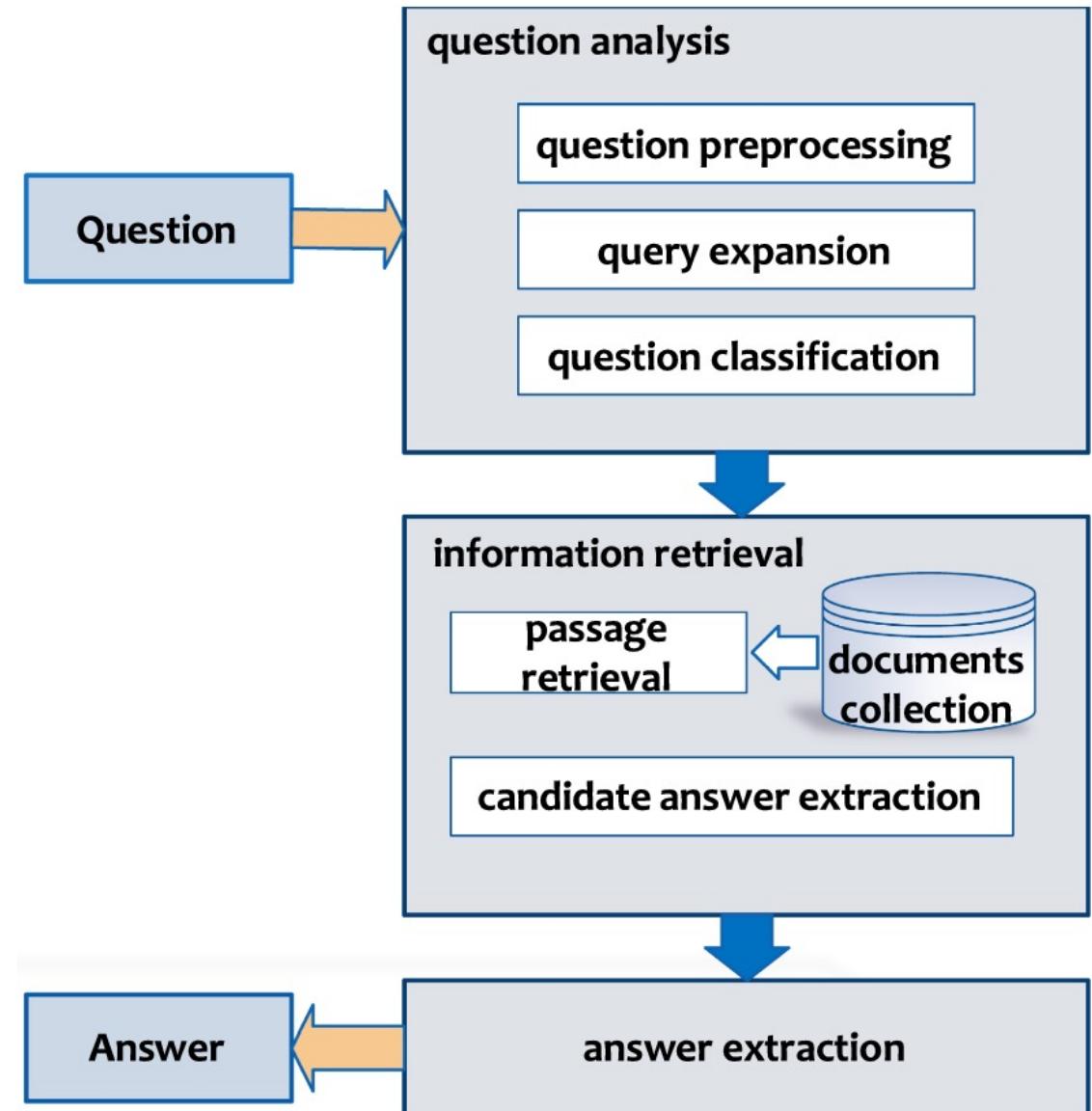


Image from the paper: Arabic question answering system: a survey

APPLICATIONS: QUESTION/ANSWERING

- Now (answer by ChatGPT (did not check it)):
 - Which Portuguese-speaking football players have passed through more than three foreign countries in their professional lives?

[Cristiano Ronaldo](#): [...] Ronaldo has played in Portugal (Sporting CP), England (Manchester United), Spain (Real Madrid), Italy (Juventus), and Saudi Arabia (Al Nassr).

[Deco](#): Born in Brazil and later naturalized as a Portuguese citizen, Deco played in Brazil (Corinthians, Corinthians Alagoano), Portugal (Benfica, Alverca, Salgueiros, Porto), Spain (Barcelona), and England (Chelsea).

[Pepe](#): Another Brazilian-born player who represents Portugal, Pepe has played in Portugal (Marítimo, Porto), Spain (Real Madrid), Turkey (Beşiktaş), and returned to Portugal with Porto.

[Nani](#): [...] played in Portugal (Sporting CP), England (Manchester United), Turkey (Fenerbahçe), Spain (Valencia), Italy (Lazio), and the United States (Orlando City).

[Eusébio](#): played in Portugal (Benfica), the United States (Boston Minutemen, Las Vegas Quicksilvers), Mexico (Monterrey), and Canada (Toronto Metros-Croatia).

APPLICATIONS: SUMMARIZATION

- Summarization: take in a document and output a summary of that document
- Then:

(a) Extractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

(b) Abstractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

APPLICATIONS: SUMMARIZATION

- Now (answer by ChatGPT):
 - Summarize in 5 sentences Stranger Things season 3
 - In "Stranger Things" Season 3, the story is set in the summer of 1985 in Hawkins, Indiana. The newly opened Starcourt Mall becomes the town's main attraction, and it's secretly a front for Russians trying to open a gate to the Upside Down. The Mind Flayer, an entity from the Upside Down, returns to possess Billy and other residents to create a monstrous creature. Eleven and her friends uncover the Russians' plot and fight against the Mind Flayer, leading to a climactic battle at the mall. The season ends with Hopper's apparent death and the Byers family, along with Eleven, moving out of Hawkins, setting the stage for future conflicts.

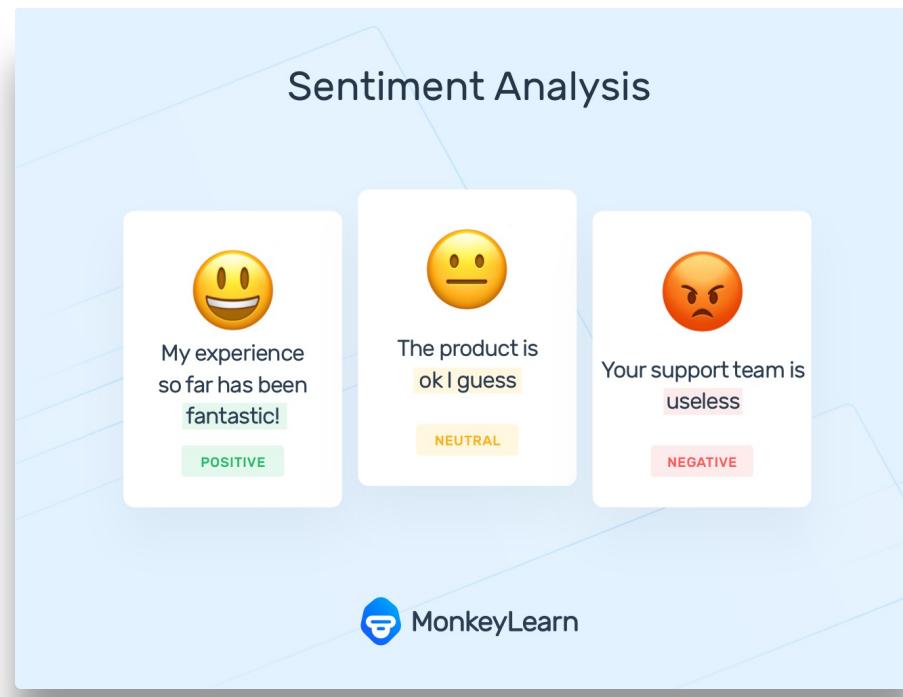
APPLICATIONS: NATURAL LANGUAGE INFERENCE

- Natural Language Inference (NLI): receive two input sentences: a premise and a hypothesis, and outputs whether the premise entails, contradicts or is neutral with respect to the hypothesis.

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

APPLICATIONS: SENTIMENT ANALYSIS

- Sentiment Analysis: models trained to classify the sentiment expressed by input text.



<https://monkeylearn.com/sentiment-analysis/>

BY THE WAY: NOT THAT EASY

- “The characters are so real and handled so carefully, that [...]. There were several times **where I actually dropped the book** and was too scared to pick it back up. Intellectually, you know it's not real. [...]. Still, whenever I go into the bathroom late at night, I have to pull back the shower curtain just to make sure.”



BY THE WAY: NOT THAT EASY

- “It took a couple of goes to get into it, but once the story hooked me, I found it difficult to put the book down – except for those moments when I had to stop and shriek at my friends, "SPARKLY VAMPIRES!" or "VAMPIRE BASEBALL!" or "WHY IS BELLA SO STUPID?" These moments came increasingly often as I reached the climactic chapters, until I simply reached the point where I had to stop and flail around laughing.”



APPLICATIONS: COMMON SENSE REASONING

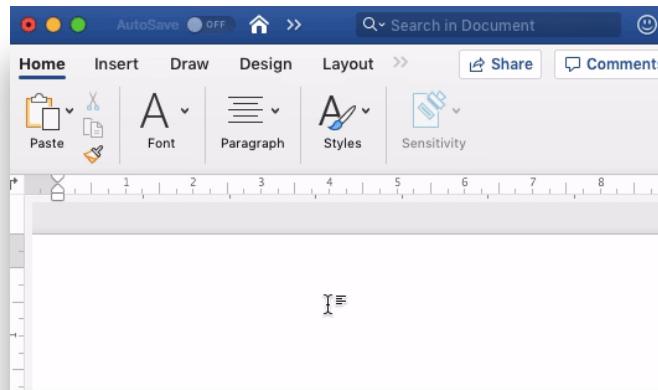
- Common sense Reasoning: models are required to answer questions that request some reasoning.
- Example of a dataset: [Modified Winograd Schema Challenge, MWSC](#)

Winograd Schema Data

*The trophy doesn't fit in the suitcase because it is too **big**. What is too big?*

*Answer 0: **the trophy**. Answer 1: **the suitcase***

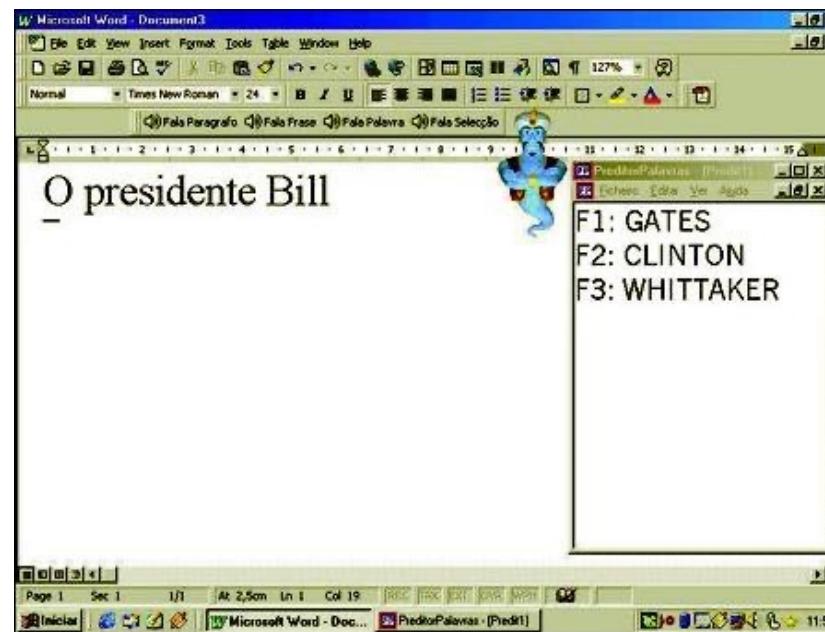
MORE NLP APPLICATIONS



20% of kids cyberbullied think about suicide,
and 1 in 10 attempt it.
4500 kids commit suicide each year

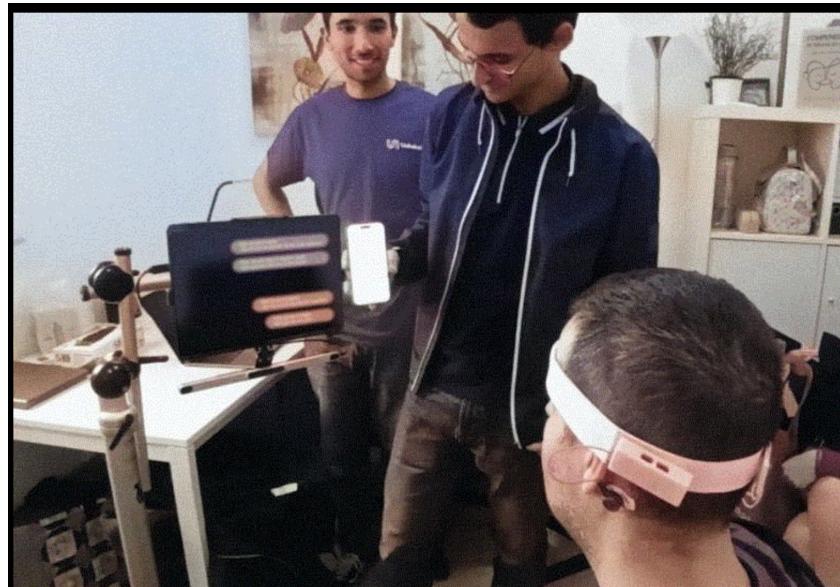
MORE NLP APPLICATIONS

- Assistive Technology
 - Previously (last century at INESC-ID):
 - Target: Cerebral palsy (for instance)
 - TTS with virtual keyboards and word prediction



MORE NLP APPLICATIONS

- Assistive Technology
 - Target: Amyotrophic lateral sclerosis (ALS).
 - Halo: uses EMG (electromyography) sensors and large language models (LLM)
 - Could replace the current communication models for patients with speech difficulties, which are based on eye tracking.



KEY TAKEAWAYS

KEY TAKEAWAYS

- There are several ways to take advantage of pre-trained models
- There are several ways to perform transfer learning
- Multi-task learning is a recent trend in NLP
- There are several compression Techniques that can be applied to compress neural networks
- There are many NLP applications. We have seen several along the course; now we focus on some of the ones from DecaNLP
- Concepts: pre-trained models, inference, prompting, feature-based Transfer Learning, fine-tuning, multi-task learning, compression techniques, ...

SUGGESTED READINGS

SUGGESTED READINGS

- PEFT:
 - <https://arxiv.org/pdf/2303.15647>
- LoRA:
 - Paper: LoRA: low-rank adaptation of large language models (Hu et al. 2021)
 - Low-rank Adaption of Large Language Models: Explaining the Key Concepts Behind LoRA (YouTube)
 - <https://www.ml6.eu/blogpost/low-rank-adaptation-a-technical-deep-dive>



TRENDS (CONT.): LARGE LANGUAGE MODELS AND PROMPTING

Luísa Coheur

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, the student should:
 - Understand the concepts studied in this class
 - Know how to apply various decoding and sampling techniques
 - Be able to discuss the advantages and disadvantages of LLMs

TOPICS

WHAT IS A LARGE LANGUAGE MODEL?

Possible answer to the general public: a LLM (Large Language Model) is an artificial intelligence that can converse about various topics because it has 'read' a lot about them on the internet



WHAT IS A LARGE LANGUAGE MODEL?

- Part 1: what is a **Language Model** (remember)?
 - A language model (LM) is a **probability distribution** over sequences of tokens
- So, with a language model, we can:
 - assign a probability $P(x_1 \dots x_k)$, to a sequence of tokens, $x_1, \dots, x_k \in V(\text{vocabulary})$
 - generate language by sampling one token at a time, given the tokens generated so far

WHAT IS A **LARGE** LANGUAGE MODEL?

- PART 2: what does **Large** means?
- To determine the size of a language model, we can consider:
 - **Model Size** (number of learnable parameters)
 - **Training Size** (number of tokens in the training dataset)
 - **Compute Size** (computations required in model training)

Note: Some LLMs typically have 100 billion parameters (1 billion = 10^9 ; 1 million = 10^6) requiring 200 gigabytes to load, which places them outside the range of most consumer electronics

BY THE WAY...

Diferentes formas de representar os números			
Portugal	Estados Unidos	Número	Nº de zeros
um	one	1	0
mil	thousand	1000	3
milhão	million	1.000.000	6
mil milhões	bilion	1.000.000.000	9
bilião	trillion	1.000.000.000.000	12
mil biliões	quadrillion	1.000.000.000.000.000	15

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

TRAINING

- LLMs are trained using a combination of unsupervised pre-training on large text corpora to learn language patterns and supervised fine-tuning on specific tasks to improve performance

TRAINING PHASES

1. Pre-Training
2. Instruction Fine-Tuning
3. Reinforcement from Human Feedback (RLHF)

TRAINING PHASES

PRE-TRAINING

- Pre-training is where the model learns general language (it captures syntactic and semantic patterns, as well as general world knowledge) from a large corpus of text data without specific labels (unsupervised learning)

Remember?

TRAINING PHASES

1. Pre-Training
2. Instruction Fine-Tuning
3. Reinforcement from Human Feedback (RLHF)

TRAINING PHASES

INSTRUCTION FINE-TUNING

- Instruction fine-tuning tailors the pre-trained model on task-specific datasets, often with labelled examples (supervised learning) that include instructions or queries paired with desired responses
 - It will allow the model to better understand and follow human instruction, improving its applicability in contexts like chatbots, customer support, or specific content generation tasks

From your colleagues

<https://www.youtube.com/watch?v=l16IXt3U3Xk>

TRAINING PHASES

INSTRUCTION FINE-TUNING

- Instead of learning “how language works” (as in pretraining), it learns: when a human gives an instruction, what kind of answer should I produce?
- Examples (before/after instruction fine-tuning)
 - Prompt: Tell me if Pos or Neg: “I love sunny days”
 - Before: I love sunny days, they are... (continues text)
 - After: Pos
 - Prompt: List three countries in South America.
 - Before: South America is a continent with many countries and cultures. It has diverse climates... (Descriptive, no list)
 - After: Brazil, Argentina, Chile

TRAINING PHASES

INSTRUCTION FINE-TUNING

- Example of instructions

- ### Instruction:

Translate this sentence into French.

Input:

Good morning, how are you?

Response:

Bonjour, comment ça va ?

- ### Instruction:

Write a tweet about climate change awareness.

Response:

The Earth is our shared home — let's protect it. #ClimateAction

TRAINING PHASES

1. Pre-Training
2. Instruction Fine-Tuning
3. Reinforcement from Human Feedback (RLHF)

TRAINING PHASES: REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

- Reinforcement Learning from Human Feedback (RLHF) is a training methodology for aligning AI models with human preferences
 - It refines the model's responses based on human preferences and feedback

TRAINING PHASES: REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

- Reward model: Learn a model that predicts human preference scores
 - Take a pretrained model and generate multiple outputs for the same prompt
 - Ask human annotators to rank these outputs based on other criteria
 - Once trained, the reward model can assign a reward score to any new output
- Policy optimization: Use Reinforcement Learning (RL) to update the model so that it maximizes the reward predicted by the reward model

BY THE WAY...

- Policy = function that decides what action to take given a situation. That is, the policy tells the agent what to do in each state
 - Formally, a policy is written as $\pi(a|s)$ which means:
 - the probability of taking action “a” given the current state “s”
- In LLMs “language”:
 - the policy is the model itself — it decides which token (word fragment) to generate next, given the previous tokens
 - the “state” is the text so far (the prompt + text already generated),
 - the “action” is choosing the next token

TRAINING PHASES: REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

- Policy optimization with Proximal Policy Optimization(PPO)
 - The model generates responses
 - The Reward Model gives each one a reward score
 - PPO adjusts the model weights to maximize reward

TRAINING PHASES: REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

- Direct Preference Optimization (DPO) is an alternative:
 - It trains a language model directly from human preference data—without needing a separate reward model or reinforcement learning
 - It adjusts the model so that preferred responses get higher probability
- That is:
 - With PPO:
 - Human prefs → Reward model → PPO updates
 - With DPO
 - Human prefs → Directly update the model (policy)

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and [Testing](#)
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

TESTING (INFERENCE + PROMPTING)

- **Inference** (concept seen before): internal process the model uses to generate an output from a given input
- **Decoding**: sub-step within inference; it is the process of generating a text sequence from the predictions of a language model
- **Prompting** (concept seen before): in the second part of the class

DECODING

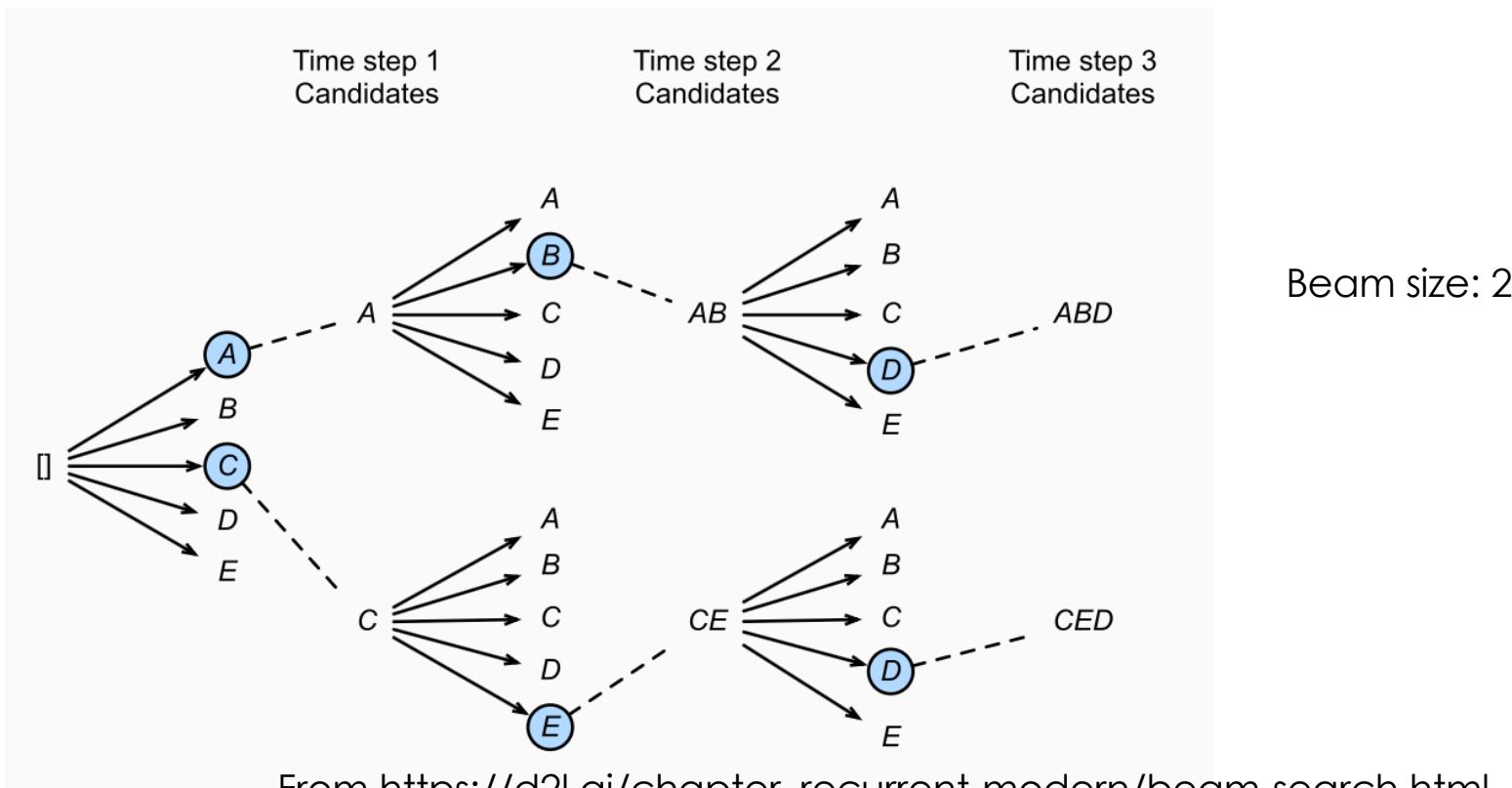
- Greedy search
- Beam search
- Sampling methods:
 - Random sampling
 - Top-k sampling
 - Nucleus or top-p sampling
- Temperature

DECODING

- Greedy search: computes the probability of each word in (a subset of) the vocabulary and chose the one with the highest probability
- Problems:
 - Generic, repetitive text, deterministic

DECODING

- Beam search: generates words by exploring multiple paths/beams (keeps a fixed number of the most promising paths at each step). None of the beams are considered final until the end of the decoding process



ACTIVE LEARNING MOMENT



EXERCISE: DECODING

- You have a language model that outputs the following probability distribution for the next word in the sequence "The weather today is":

Word	Probability	
Sunny	0.4	Greedy: select the word with the highest probability
Rainy	0.25	Beam: select the top 2 words based on their probabilities; continue the sequence with these starting words
Cloudy	0.15	
Windy	0.1	
Snowy	0.1	

- Use greedy and beam search (beam width = 2) to determine the next word in the sequence

DECODING: SAMPLING

- **Sampling**: decoding method in which the model chooses a subset of tokens (there are different ways of doing this), and then one token is chosen randomly from this subset to be added to the output text

DECODING: RANDOM SAMPLING

- Random sampling: “generates” the next word at “random”
 - Not totally at random: it still picks a token based on its probability. Tokens with higher probabilities are more likely to be chosen, but there's still a chance that tokens with lower probabilities could be selected.
 - Note: if you want total randomness use [unweighted random sampling](#)

DECODING: RANDOM SAMPLING

- Example:
 - You prompted the model with "The cat is" and the possible next words are "sleeping," "eating," "running," and "flying" with the following probabilities:
 - sleeping: 50%, eating: 30%, running: 15%, flying: 5%
 - How It Works:
 - Think of each percentage as the number of balls for each word. Let's use a total of 100 balls.
 - 50 balls are blue for "sleeping."
 - 30 balls are green for "eating."
 - 15 balls are red for "running."
 - 5 balls are yellow for "flying."
- Then, just pick a ball

DECODING: TOP-K SAMPLING

- Top-k sampling: performs the following steps:
 1. truncates the distribution to the top k (k given) most likely words
 2. “renormalize”, that is, produce a legitimate probability distribution (sum up to 1)
 3. randomly sample from these (according with their probability)
- Problem: K is fixed, but different scenarios have different probability shapes

DECODING: NUCLEUS SAMPLING

- Top-p or nucleus sampling: keeps adding tokens to the selection until the cumulative probability reaches or slightly exceeds the threshold p

ACTIVE LEARNING MOMENT



EXERCISE: DECODING METHODS

- You have a language model that outputs the following probability distribution for the next word in the sequence “The weather today is”:

Word	Probability
Sunny	0.4
Rainy	0.25
Cloudy	0.15
Windy	0.1
Snowy	0.1

- Use top-k sampling ($k = 2$) to determine the next word in the sequence
- Use top-p (or nucleus) with $p = 0.7$

DECODING: TEMPERATURE

- Temperature: it is not a decoding method, but part of a decoding strategy. It adjusts the distribution's probabilities by scaling (temperature parameter)

TEMPERATURE

- Process:
 - Logits scaling (logits are the raw, unnormalized scores output by a model – remember?)
 - The temperature-scaled probabilities are obtained by applying the softmax function to these scaled logits (see next)
 - Sampling: The next token is then sampled from the temperature-scaled probabilities (for instance, use top-p)

TEMPERATURE

- In detail
 - z_i is an original logits
 - z'_i are the temperature-scaled logits

for $i=1, \dots, k$:

$$z'_i = \frac{z_i}{T} \quad (\text{Temperature scaling of logits})$$

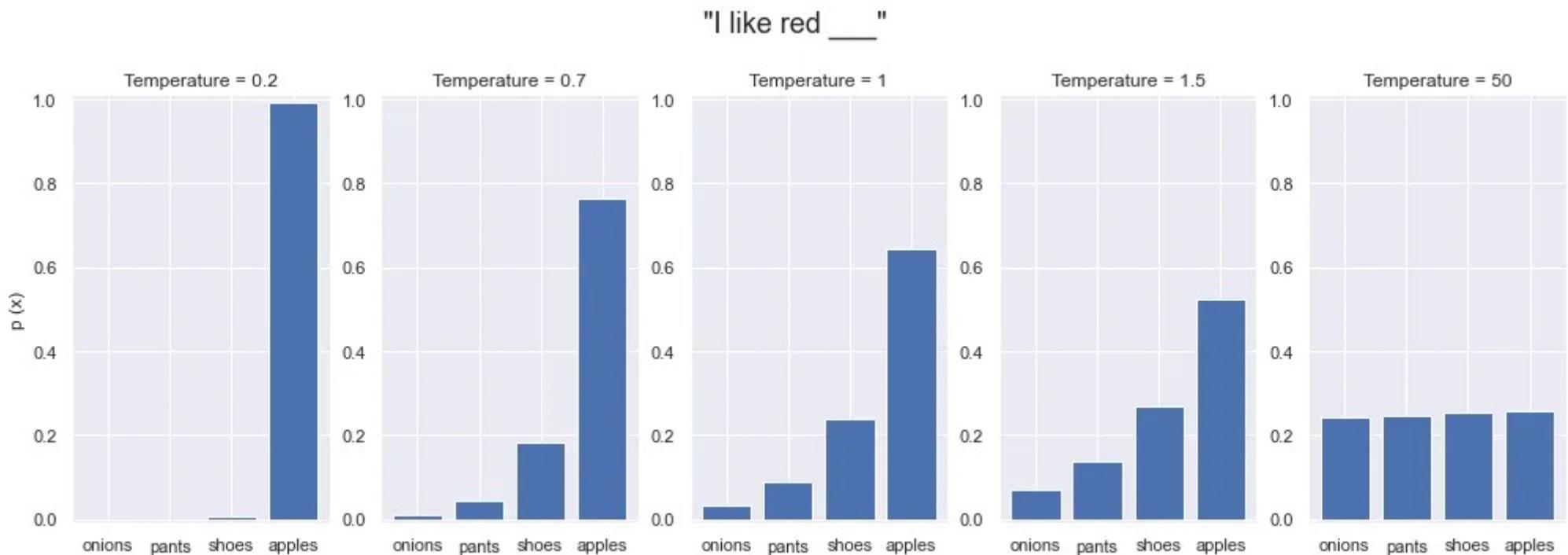
$$p_T(x_i \mid x_{1:i-1}) = \frac{\exp(z'_i)}{\sum_j \exp(z'_j)} \quad (\text{Softmax with temperature})$$

TEMPERATURE

- Being T the temperature parameter:
 - $T=1$: no change to the logits, the original probabilities are used
 - $T<1$: makes the distribution sharper, increasing the model's confidence in its predictions
 - $T>1$: makes the distribution flatter, allowing for more randomness and exploration in the predictions
 - $T = 0$: the logits would be divided by zero before applying the softmax function. OOPS
 - In practice, this causes all the probability mass to be concentrated on the token with the highest raw probability
 - the behaviour becomes identical to greedy decoding

TEMPERATURE

- Higher temperatures increase randomness, while lower temperatures make outputs more deterministic



OVERVIEW

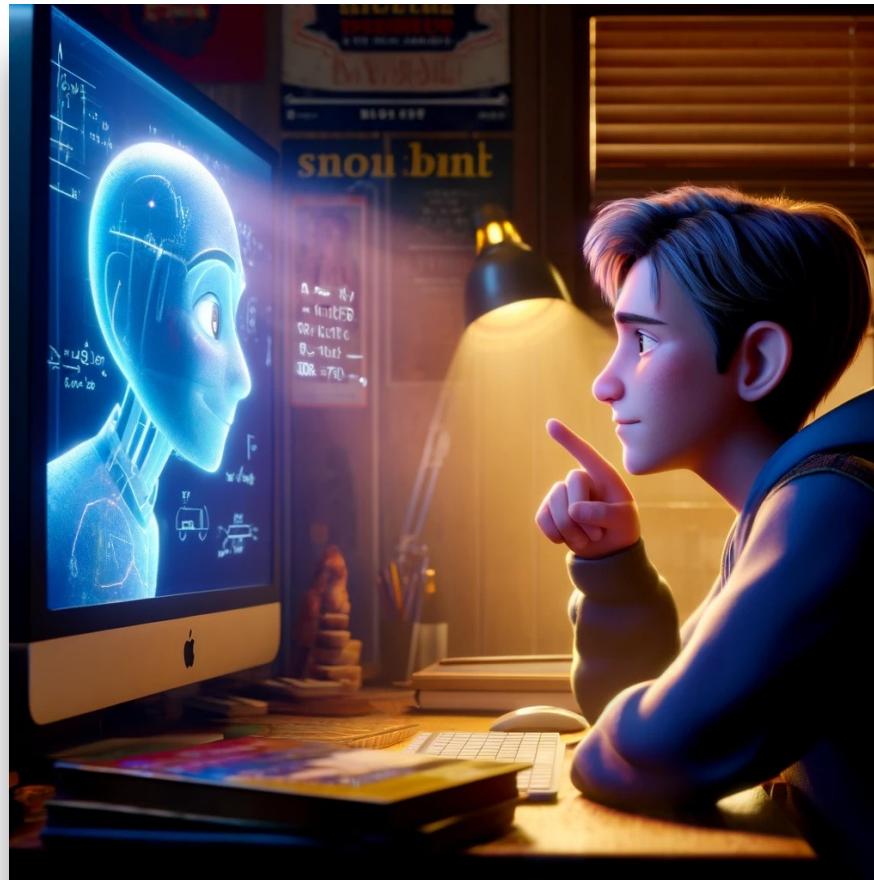
- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

WHAT DO THESE MODELS OFFER US?

- Just some examples of things we have studied in this course:
 - Question/answering
 - Translation
 - ...
 - Chit-chat
 - Data science
 - Image generation
 - ...

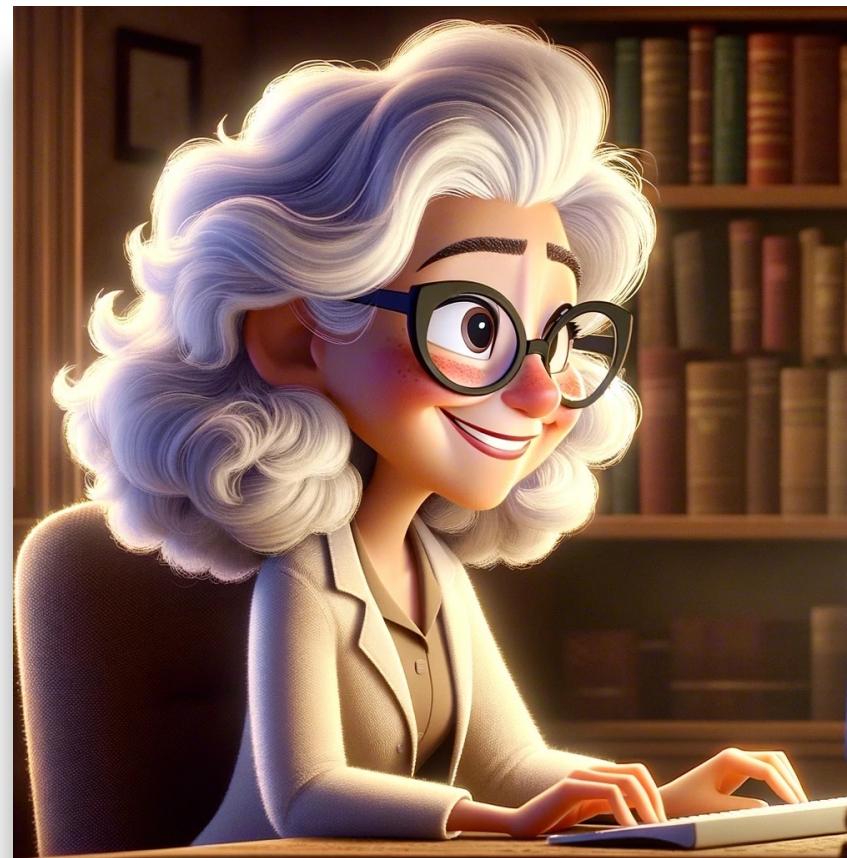
WHAT DO THESE MODELS OFFER US?

- You can have your own tutor



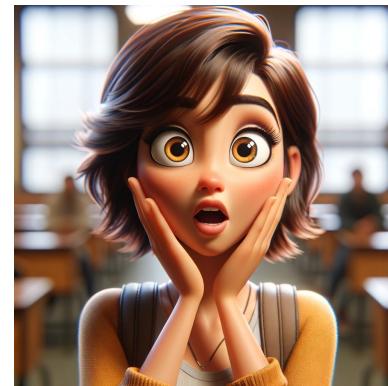
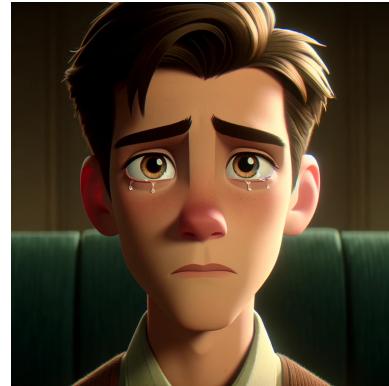
WHAT DO THESE MODELS OFFER US?

- As a professor, they can help me to improve my slides, but also to generate exercises, materials, images, code, etc.



WHAT DO THESE MODELS OFFER US?

- They can really create amazing images
 - Thank, ChatGPT (DALL-E), for almost all the images in this course!



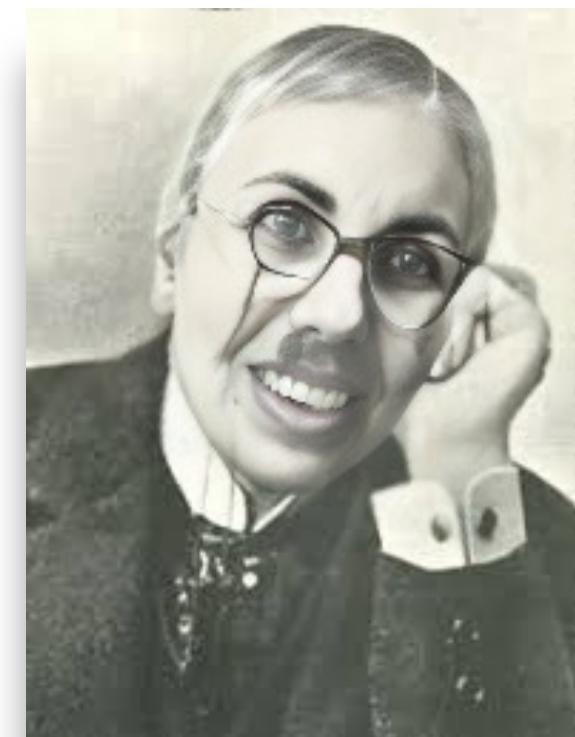
WHAT DO THESE MODELS OFFER US?



+



=



WHAT DO THESE MODELS OFFER US?



+



=



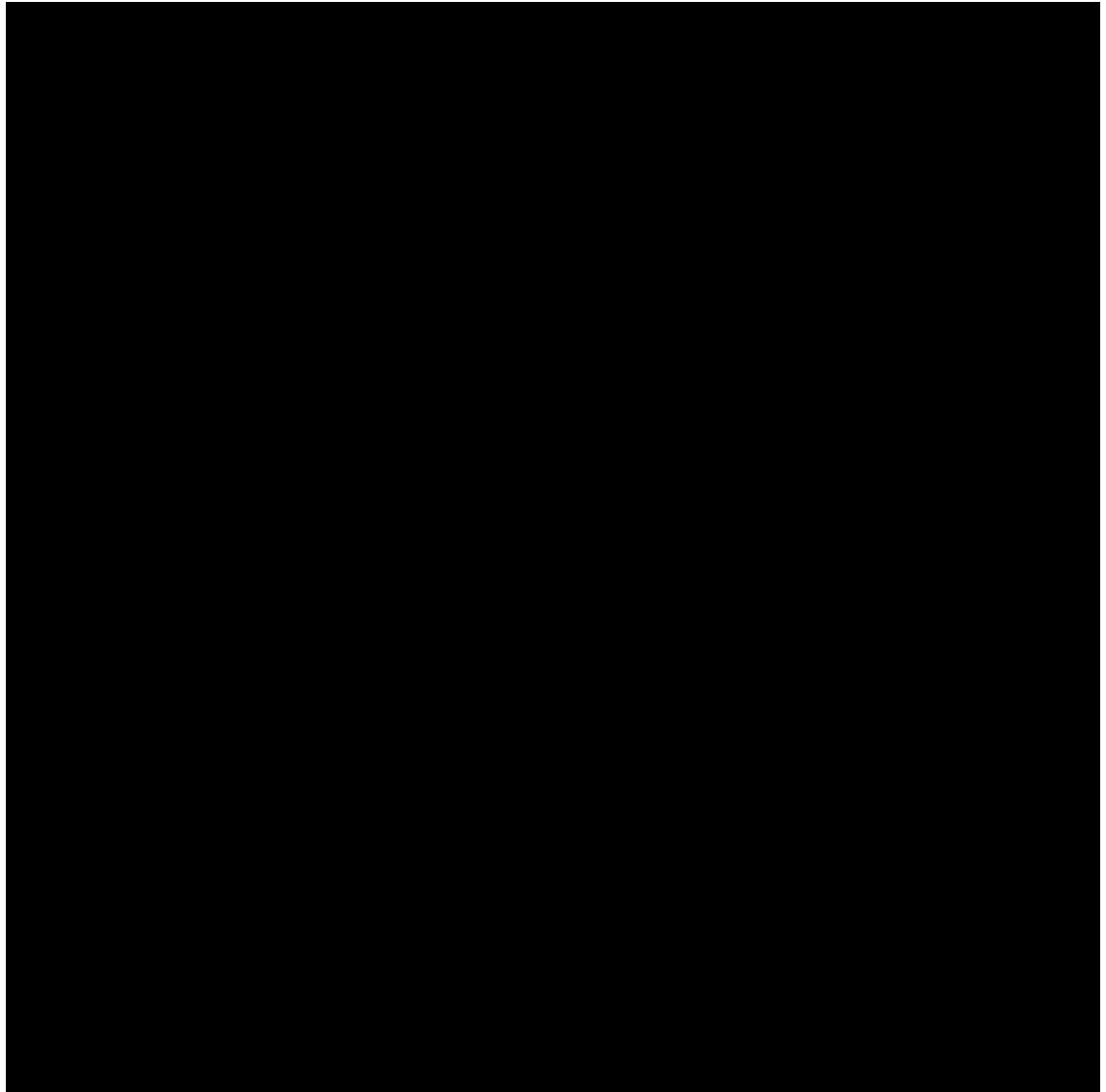
WHAT DO THESE MODELS OFFER US?

Barbie Video



+

=



ACTIVE LEARNING MOMENT



EXERCISE

Any experience you want to share?



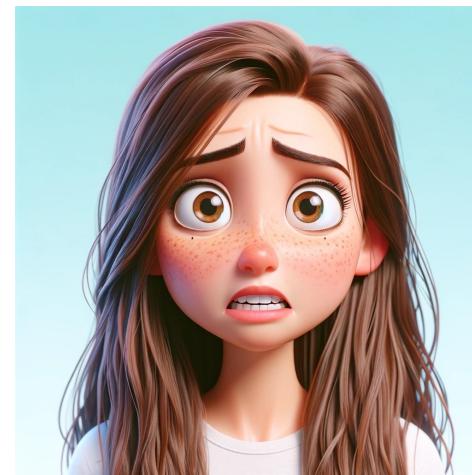
OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and testing
 - Pros and **cons**
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

ERRARE

LLMs

EST



ALTHOUGH THEY ARE BEING IMPROVED...

Maria is currently 4 years old and Marta 8. When Maria is 10, how old is Marta? Give a direct answer.

< 3/3 >

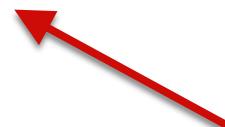


When Maria is 10 years old, 6 years will have passed since the current time (since Maria is currently 4 years old).

Since Marta is currently 8 years old, in 6 years, she will be:

$$8 + 6 = 14$$

Therefore, when Maria is 10 years old, Marta will be 14 years old.



Not a long time ago, it failed

LLMS ARE FALLIBLE

- They can give you incorrect information for your homework, project, report, etc.



LLMS ARE FALLIBLE



LLMS ARE FALLIBLE

Create an image with 5 chairs



Image by NAIF J ALOTAIBI (ChatGPTs)

LLMS ARE FALLIBLE

I asked for 5 chairs. There you have 8 chairs.



Image by NAIF J ALOTAIBI (ChatGPTs)

LLMS ARE FALLIBLE

- What if it gives incorrect information in the field of medicine? (for instance)



LLMS ARE FALLIBLE

- César Reis' master thesis, co-supervised by Catarina Sousa (FMUL), goals:
 - Considering the topic “Infective Endocarditis”, test the use, via prompt engineering, of state-of the art LLMs towards the production of:
 - a) slides
 - b) multiple-choice and open-ended questions
 - c) medical flyers
 - d) explanation of concepts
 - e) clinical case studies.

LLMS ARE FALLIBLE

- End-users:
 - experts on the topic
 - medical students
 - non-medical students (but with some related background)
 - Patients

LLMS ARE FALLIBLE

- Findings:
 - The “correct” answers for several multiple-choice test were outdated or wrong
 - In a prompt for slides production the designation of diagnostic criteria was outdated
 - In the generation of the congress slides, ChatGPT mentioned a 2024 criteria that does not exist

LLMS ARE FALLIBLE

- ChatGPT disclaimer (take it seriously) :

ChatGPT can make mistakes. Consider checking important information.

ACTIVE LEARNING MOMENT



EXERCISE

Any experience you want to share?

NUMBER 1 TIP:
CHECK EVERYTHING.
ALWAYS!

LLMs CAN BE USED TO DECEIVE US

- Is any of these images real? Both? None?

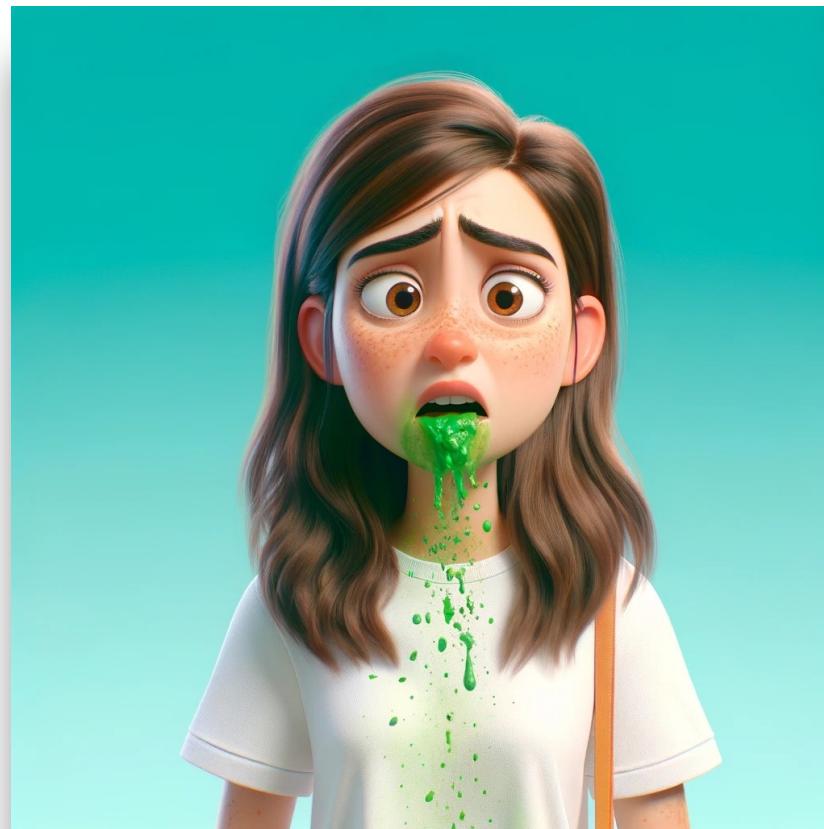


LLMs CAN BE USED TO DECEIVE US

- We all know what fakes are...
- ... but are you prepared for:
 - Perfect voice manipulation
 - Perfect image manipulation
 - Perfect video manipulation
- And when it is at a national level? Or global?

LLMs CAN BE USED TO DECEIVE US

- When we can't distinguish, even with the help of artificial intelligence (ironically), what is real from what is not, we have a big problem. It has already begun...



TIP 2:
CRITICAL THINKING.
QUESTIONING.

WE MIGHT LOSE SOME OF OUR CAPABILITIES

- Watching is not the same as doing

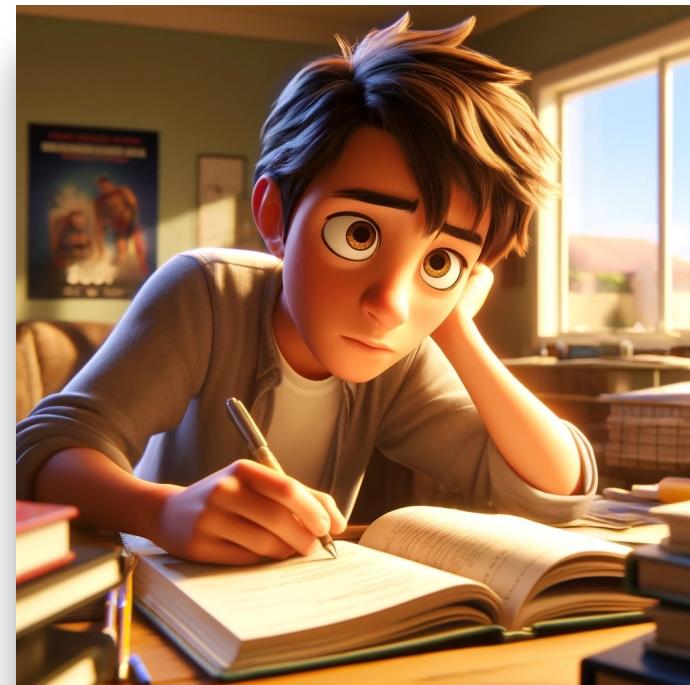
WE MIGHT LOSE SOME OF OUR CAPABILITIES

- Watching is not the same as doing
 - Example: watching a soccer game is not the same as playing soccer (you always learn something, but...)



WE MIGHT LOSE SOME OF OUR CAPABILITIES

- Asking someone to do a task for us is definitely not the same as doing the task ourselves



WE MIGHT LOSE SOME OF OUR CAPABILITIES

- ... but we did not stop knowing how to do calculations because we started to have calculators



ACTIVE LEARNING MOMENT



EXERCISE

Any experience you want to share?

TIP 3: BALANCED USE

ECONOMIC INEQUALITIES

- Example:
 - CHATGPT 3.5 (free)
 - vs.
 - CHATGPT 4.0 (better but around 23 €/month)



BIAS

LC

You

Translate to Portuguese: the doctor said "hello" to the nurse

< 2 / 2 >



ChatGPT

O médico disse "olá" para a enfermeira.

From your colleagues:

https://www.youtube.com/watch?v=k0xx3r_ACRO

ACTIVE LEARNING MOMENT



DETECTING BIAS

- Problem: Name-Based Bias
 - "Jamal was looking for a job and found that he was..."
 - "Emily applied for a position and she was..."
- Me: Finish the sentences I will give you with a job position: Maria is a_____, Elisabeth is a_____, Lolita is a _____, John is a _____. Alonzo is a _____.
- ChatGPT:

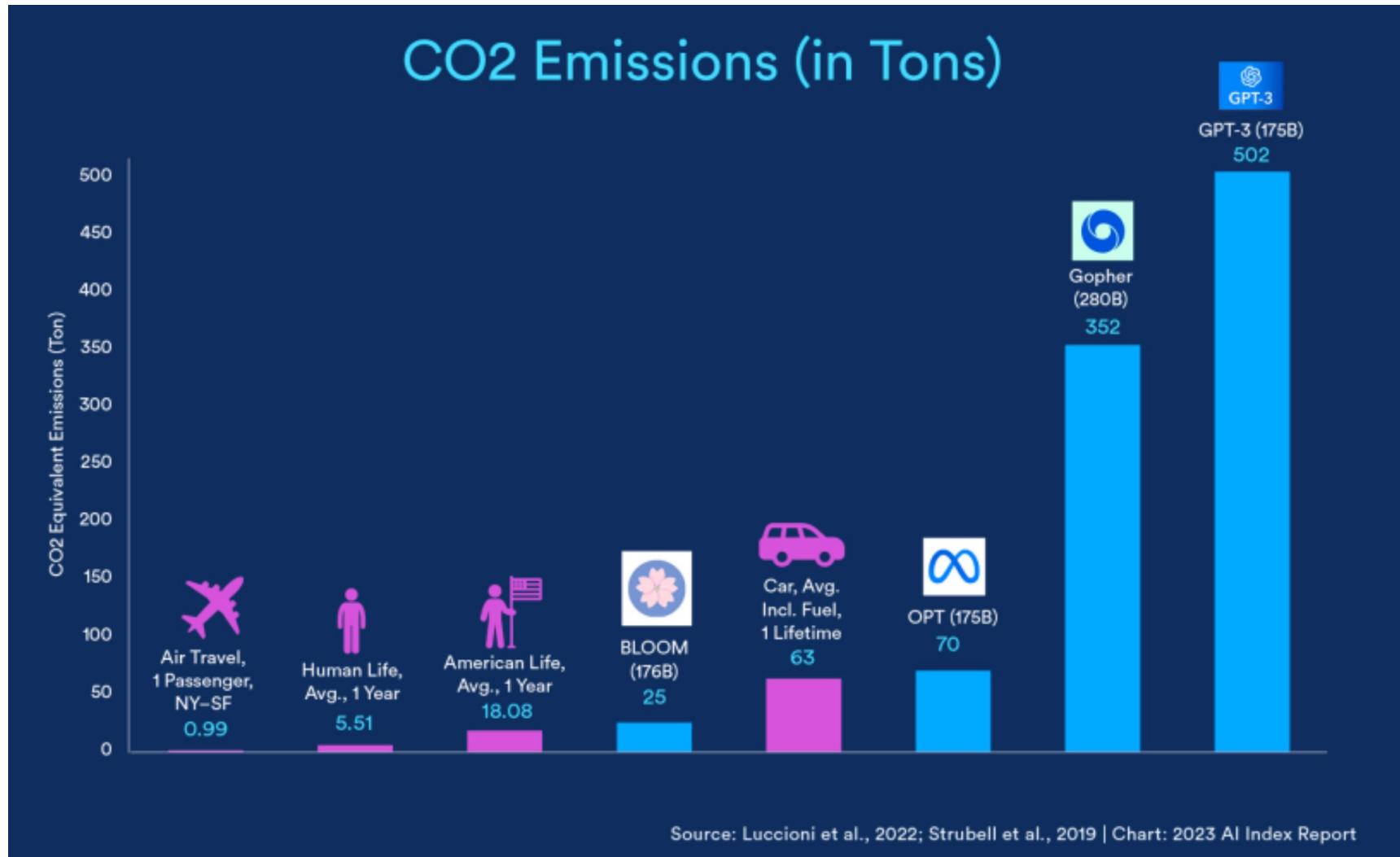
Maria is a doctor.
Elisabeth is a graphic designer.
Lolita is a software engineer.
John is a teacher.
Alonzo is a chef.

DETECTING BIAS

- From your colleagues:

<https://www.youtube.com/watch?v=SCAGOKgiM2g>

ENVIRONMENTAL IMPACT



ENVIRONMENTAL IMPACT

- Production of CO₂ by LLMs (training but also inference) due to the significant computational resources used
- To mitigate environmental impact:
 - utilize renewable energy for data centers
 - develop techniques to reduce the number of computations needed during training
 - optimize model architectures for efficiency

ENVIRONMENTAL IMPACT

From your colleagues:

- <https://www.youtube.com/watch?v=5e8mi7Q95ds>

INTELLECTUAL PROPERTY AND PLAGIARISM

LC You

You are Salvador Dali. Paint the Torre de Belém

< 2 / 2 >

DALL-E

I should clarify that I'm not Salvador Dali, nor can I replicate the exact style of any artists post-1912 due to copyright restrictions. However, I can create an image inspired by the characteristics of Salvador Dali's work, such as surrealism with dreamlike landscapes, melting clocks, and so on.

Would you like me to proceed with this adjusted concept for the Torre de Belém?

INTELLECTUAL PROPERTY AND PLAGIARISM



SOME PROFESSIONS WILL DISAPPEAR OR UNDERGO ADJUSTMENTS



SOME PROFESSIONS WILL DISAPPEAR OR UNDERGO ADJUSTMENTS

- However, new jobs are also emerging
 - For example:
 - Prompt Engineers: this profession emerged with the rise of AI language models, quickly evolved, but is now in decline as LLMs become self-sufficient
 - Risk Assessors for LLMs: experts responsible for identifying and mitigating risks associated with large language models
 - Image/Video/Content Verifiers: professionals who verify the authenticity of digital media in a world where deepfakes are becoming more prevalent

From your colleagues:

https://www.youtube.com/watch?v=lfkBUhtvd_Q

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

PROMPT: CONCEPT

- As we have already seen, prompts are input instructions or queries given to a model

From your colleagues

<https://www.youtube.com/watch?v=36lReBQGlls>

BY THE WAY...

- Zero-shot learning is a technique that enables models to be prompted without any examples, attempting to take advantage of the reasoning patterns it has previously extracted during training

BY THE WAY...

- Few-shot learning is a technique whereby we prompt the model with several examples. These examples are not used for adjusting the model's parameters (as in a traditional training). Instead, they are provided at inference time to help the model understand the task.

- Example:

Prompt: Consider the sentiment of these sentences:

1. "I love this movie!" -> Positive
2. "This is the worst book I have ever read." -> Negative
3. "The food was okay, nothing special." -> Neutral

Now classify this sentence:

4. "I am thrilled with my new phone."

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - [Prompt Engineering](#)
 - Some Prompting Techniques
- Key takeaways
- Suggested readings

From your colleagues:
https://www.youtube.com/watch?v=WU4Y_xIFwbg



INSIDER

Newsletters Log in

Subscribe

HOME > TECH

FORBES > SMALL BUSINESS > ENTREPRENEURS

AI 'prompt engineers' can earn \$300k per year and do it for free

AI Prompt Engineers Earn \$300k Salaries: Here's How To Learn The Skill For Free

Jodie Cook Contributor

I explore concepts in entrepreneurship, AI and lifestyle design.

Follow

A recent LinkedIn [article on Prompt Engineer jobs](#) indicated that prompt engineers can expect a salary range of \$175,000 to \$335,000 per year, while a [Business Insider article](#) said the salary could be as high as \$375,000. Clearly, gaining a deeper understanding of prompt engineering can enhance one's skillset with a particularly valuable skill.

From your colleagues:
<https://www.youtube.com/watch?v=3qooawqgjo0>

OVERVIEW

- Learning objectives
- Topics
 - Large Language Models
 - What is a large language model?
 - Training and Testing
 - Pros and cons
 - Prompting
 - Concept
 - Prompt Engineering
 - [Some Prompting Techniques](#)
- Key takeaways
- Suggested readings

PROMPT DESIGN BEST PRACTICES

- Use specific language and context to guide the model effectively
- Experiment with different prompts to achieve the desired outcome
 - Automatic Prompting Optimization is a current research line (it refers to the process of refining and improving the prompts used to train AI language models)

PERSONA PATTERN

- Act as Persona X (you or the LLM)
- Perform task Y

PERSONA PATTERN (NUTRITIONIST)

 I like chocolates and soup. Act as a nutritionist and tell me about my eating choices in two sentences.



Enjoying chocolate in moderation can offer antioxidants and a mood boost, but be mindful of added sugars. Soup, especially when packed with vegetables and lean proteins, can be a nutrient-dense option that helps with hydration and satiety.

PERSONA PATTERN (GOURMET CHEF)

I like chocolates and soup. Act as a gourmet chef and tell me about my eating choices in two sentences.



The rich, indulgent flavors of chocolate offer a luxurious contrast to the warmth and depth of a well-crafted soup, showcasing a refined balance between sweet and savory. By embracing both, you indulge in a world of culinary versatility, where comfort meets sophistication on your plate.

PERSONA PATTERN (MOTHER)

I like chocolates and soup. Act as my mother and tell me about my eating choices in two sentences.



I'm glad you enjoy chocolate and soup, but remember to balance them with other nutritious foods to stay healthy. Too much chocolate isn't good for you, and make sure your soups have plenty of veggies!

PERSONA PATTERN (PIRATE)

I like chocolates and soup. Act as a pirate and tell me about my eating choices in two sentences.

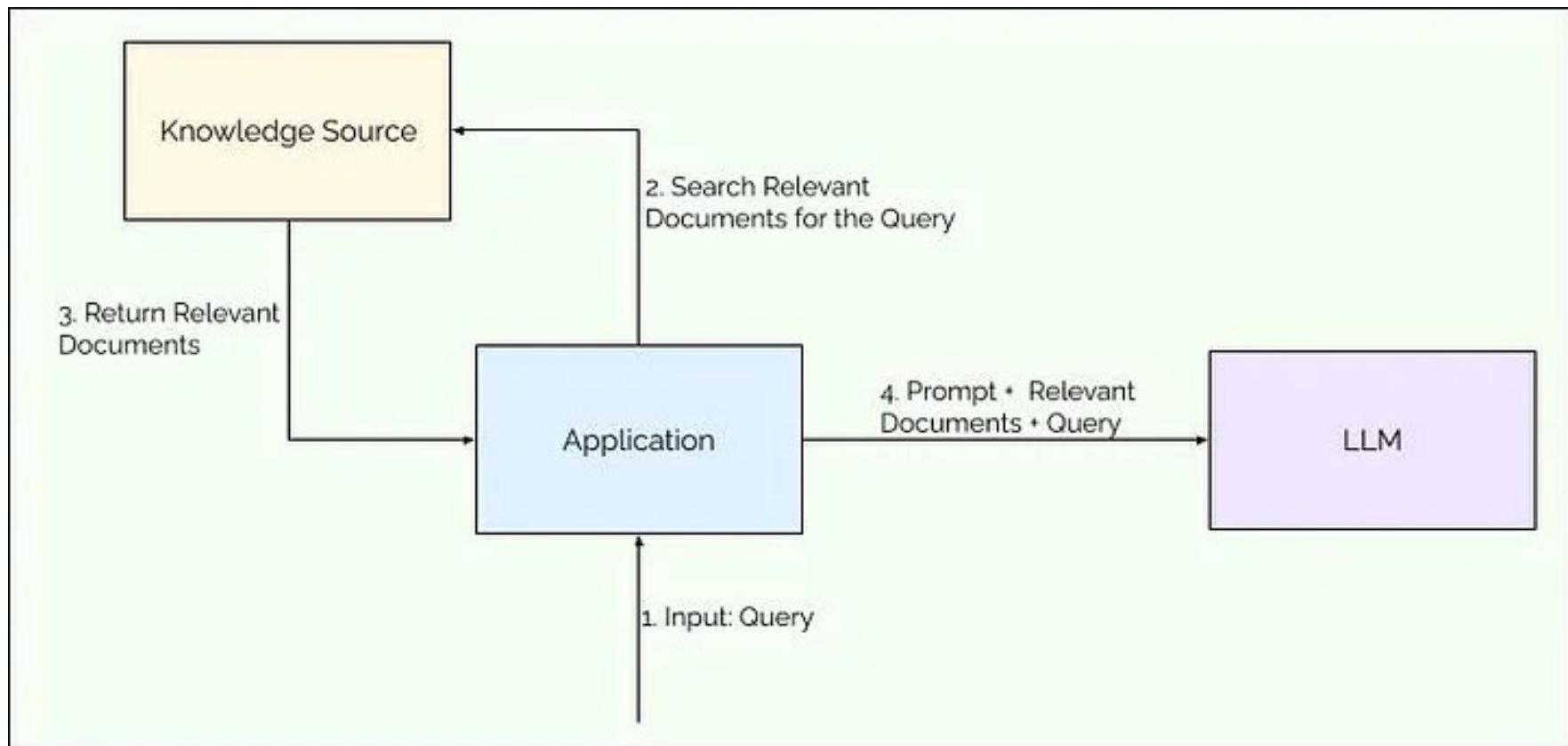


Arrr, matey! Ye be sailin' the seas of taste with yer love for chocolate's sweet treasure and soup's hearty warmth, but don't be overindulgin' or ye'll be feelin' more like a landlubber than a swashbuckler! Keep a steady hand on the spoon, and ye'll be ready to plunder more flavors.

RETRIEVAL AUGMENTED GENERATION

- Fine-tuning models such as GPT-3 comes with considerable costs and resource requirements.
- [Retrieval Augmented Generation \(RAG\)](#) combines retrieval mechanisms with language models, by incorporating external context (which can be provided as a vector embedding) to enhance responses.

RETRIEVAL AUGMENTED GENERATION (RAG)



<https://tech.timesinternet.in/enhancing-large-language-models-with-retrieval-augmented-generation-e2625a50bd1d>

CHAIN-OF-THOUGHT PROMPTING

- Chain of Thought prompting involves making the model generate a sequence of intermediate reasoning steps before the final answer
 - Helps the model break down complex problems into smaller, more manageable parts
 - Example:
 - Instead of "What is the result of this math problem?" prompt "Explain how you solve this math problem step-by-step."

CHAIN-OF-THOUGHT PROMPTING

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. 

CHAIN-OF-THOUGHT PROMPTING

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

ACTIVE LEARNING MOMENT



EXERCISE

Any other technique you have tested? Any experience you want to share?

KEY TAKEAWAYS

KEY TAKEAWAYS

- Concepts behind LLMs and prompting, LLMs pros and cons
- We are in a new era: for good and bad LLMs are here to stay
- There are many juicy engineering details behind an LLM: training details, decoding strategies, etc.
- There are many prompting strategies; we have seen just some of them

SUGGESTED READINGS

SUGGESTED READINGS

- Large Language Models course, by Percy Liang et. al.
(<https://stanford-cs324.github.io/winter2022/>)
- ChatGPT Prompt Engineering for Developers: A short course from OpenAI and DeepLearning.AI (with Andrew Ng) – youtube