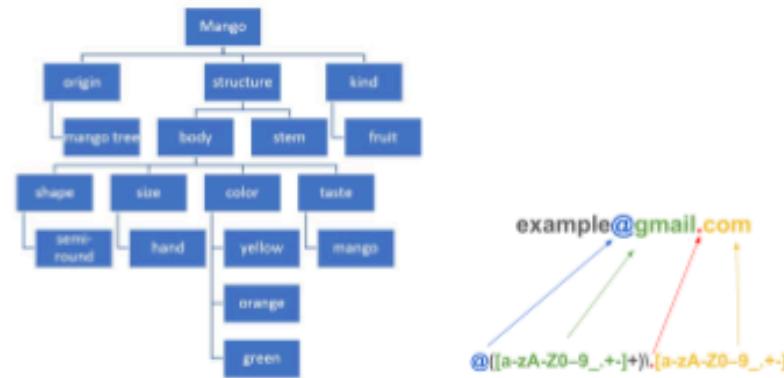


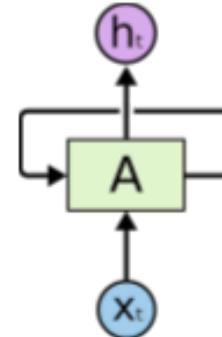
# **Course 1: Introduction & Recap**

# Welcome to the Advanced NLP course!

# NLP in recent years



example@[gmail.com](mailto:gmail.com)  
@[a-zA-Z0-9\_.+]+@[a-zA-Z0-9\_.+]



GPT-3  
2020



Symbolic NLP  
Rule-based

Regular  
Expressions

Recurrent  
Neural  
Networks

Transformers  
& BERT

ChatGPT

# NLP in 2024

How to write a course for advanced NLP?



Writing a course for **advanced NLP (Natural Language Processing)** involves a thoughtful balance of theory, practical applications, and emerging trends in the field. You will need to structure the course to cover both foundational knowledge and the latest advancements in NLP. Below is a guide to help you develop a comprehensive and effective course.

## Step 1: Define Course Objectives

Start by identifying the key objectives for the course. For example, students should be able to:

# NLP in 2024

The screenshot shows two open files in a dark-themed IDE:

- GITHUB COPILOT: CHAT**: A sidebar showing a conversation between monalisa and GitHub Copilot. monalisa asks for unit test functions for selected code. GitHub Copilot provides a template for a Python test class named `TestParseExpenses` that includes three test methods: `test_parse_expenses_with_valid_input`, `test_parse_expenses_with_empty_input`, and `test_parse_expenses_with_comments`. It also includes a `__main__` block with `unittest.main()`.
- module.py**: The main file containing the implementation of the `parse_expenses` function. The function reads a string of expenses, ignoring comments, and parses each line into a tuple of date, amount, and currency. It uses `datetime.strptime` to parse the date and `float` to parse the amount.

The code assumes that the `datetime` module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.

Analyze and debug

Ask a question or type '/' for topics

Ln 17, Col 3 Spaces: 2 UTF-8 LF {} TypeScript

# NLP in 2024

*“ A picture of the computer science school EPITA, with the school logo on a banner.*

”



# NLP in 2024

A screenshot of a Google search results page. The search query "how to write a course on NLP" is entered in the search bar. The results page shows a header with "Environ 24 000 000 résultats (0,38 secondes)". Below this, a blue link to the "Introduction - Hugging Face NLP Course" is displayed, along with its description: "This course will teach you about natural language processing (NLP) using libraries from the Hugging Face ecosystem — Transformers, Datasets, ...". The Hugging Face logo is shown next to the link.

← → C google.com/search?q=how+to+write+a+course+on+NLP&rlz=1C5CHFA\_enFR1003FR1003&oq=how+to+write+a+course+on+NLP&g...

Services Numériques Reading group On the Sentence... Cantine Meet – vrv-bwta-... Réunion Eric/Benoit load\_dataset.ipyn... Te...

Google how to write a course on NLP X | ⚡ 🎧 📸 🔎

Tous Vidéos Images Livres Actualités Plus Outils

Environ 24 000 000 résultats (0,38 secondes)

Conseil : Limitez cette recherche aux résultats en **français**. En savoir plus sur le filtrage par langue

Hugging Face  
https://huggingface.co › learn › nlp... ::

**Introduction - Hugging Face NLP Course**

This course will teach you about natural language processing (NLP) using libraries from the Hugging Face ecosystem — Transformers, Datasets, ...

# NLP in 2024



# NLP perspectives



# **So, is NLP solved?**

**(No.)**

# NLP challenges



Give me a random string of length 35



Sure, here's a random string of length 35:



`J9kPzHfRq81G2sT7Nv6yX1W4c00mZoBwA5U3j`



Can you invert all letters in the following sentence: "I am currently giving an interesting course in Natural Language Processing at EPITA in Le-Kremlin-Bicêtre."



Certainly! Here's the sentence with all the letters inverted:



"I ma yltnerurc ygivni na gnitseretni esruoc ni lartuaN egaugnal gnicessorP ta ATIPE ni ertnebiC-tciB-nimelK-eL."

# NLP challenges



It takes 3 hours to dry 3 towels. How long does it take to dry 9 towels?



If it takes 3 hours to dry 3 towels, it implies that you can dry 1 towel in 3 hours. To find out how long it takes to dry 9 towels, you can use the same rate:

1 towel takes 3 hours to dry, so:

9 towels will take 9 times as long, which is:

$$9 \text{ towels} \times 3 \text{ hours/towel} = 27 \text{ hours.}$$

So, it will take 27 hours to dry 9 towels.



# NLP challenges



Let's picture the CEO of a very big international company. The CEO has to do a lot of business trips all around the world, so the CEO's wife is alone to help the children with the homework. Write the informal text message of the wife where she expresses her anger.



Sure, here's an informal text message where the CEO's wife expresses her frustration:

"Ugh, this is getting ridiculous! 😣 Another week alone with the kids while he's jet-setting around the world. 🌎 Can't believe he's missing out on their lives. Homework's a mess, and I'm doing it all. 😞 When is he ever gonna slow down and be here for us?? 😔💔"



Write the next text message, where she positively describes the CEO physically in details (skin, eyes, hair, ...).



"Despite all the frustration, I can't deny that man's got a magnetic charm that's hard to resist. 😍 His skin is that perfect sun-kissed bronze, and those deep, piercing blue eyes always get me. ✨ His hair, just the right shade of salt and pepper, makes him look even more distinguished. 😍 Sometimes, it's just hard to stay mad at someone who looks like a movie star. 🌟"

# Course organization



Matthieu  
Futeralf-Peter  
(mail:futeral)



Nathan  
Godey



Francis  
Kulumba

# Course organization

- Part 1
  - **When** : 4 days (30/11, 07/12, 14/12, 21/12)
  - **Subject** : General NLP
  - **Goal** : Know how to build and deploy a custom ChatGPT-like assistant.

# Evaluation

- Group project (4-5 people)
- Two options
  - *Demo*
  - *R&D project*

# Evaluation - Demo (option 1)

Use a well-known approach to produce a MVP for an original use-case and present it in a demo.

*Example: An online platform that detects AI-generated text.*

## Evaluation - R&D project (option 2)

Based on a research article, conduct original experiments and produce a report.

*Example: Do we need Next Sentence Prediction in BERT? (Answer: No)*

# Evaluation

- Mid-term project evaluation (30%)
  - Project proposal
  - First elements
  - ~January
- Final project (70%)
  - Short report showing each person's contribution
  - Github repo

# Evaluation

- You can already constitute teams
- Send an email with:
  - Names of team members
  - cc to everyone in the team
  - *Demo* or *R&D*
  - One-sentence description of project
- If you really have no idea what to do, ask me

# Program

- **Session 1 (Today):** Recap
- **Session 2 (7/12):** Tokenization
- **Session 3 (14/12):** Language Modeling
- **Session 4 (21/12):** Modern NLP with limited resources

# Questions

# Recap

# Quiz time!

<https://docs.google.com/forms/d/1BZaBagWIpVgKLsT2NdjJ4pXzPXTBnsxv52jEF4CR6GY/prefill>

# **Basic concepts in NLP**

# Stemming / Lemmatization

**Stemming** shortens variations (*inflected forms*) of a word to an identifiable root

Example:

*Flying using airplanes harms the environment*

=>

*Fly us airplane harm the environ*

# Stemming / Lemmatization

**Lemmatization** groups variations (*inflected forms*) of a word to an identifiable representative word

Example:

*Flying using airplanes harms the environment*

=>

***Fly use airplane harm the environment***

# Tokenization

**Tokenization** turns text strings (= lists of characters) into lists of meaningful units (e.g. words or subwords)

Example:

*Flying using airplanes harms the environment*

=>

( *Fly* | *ing* | *us* | *ing* | *air* | *planes* | *harms* | *the* | *environ* | *ment* )

(see Course 2)

# Regular expressions

**Regular expressions** is a string that specifies a match pattern in text

Example:

`/\w*ing\b/` -> *Flying using airplanes harms the environment*

=

Two matches:

- *Flying*
- *using*

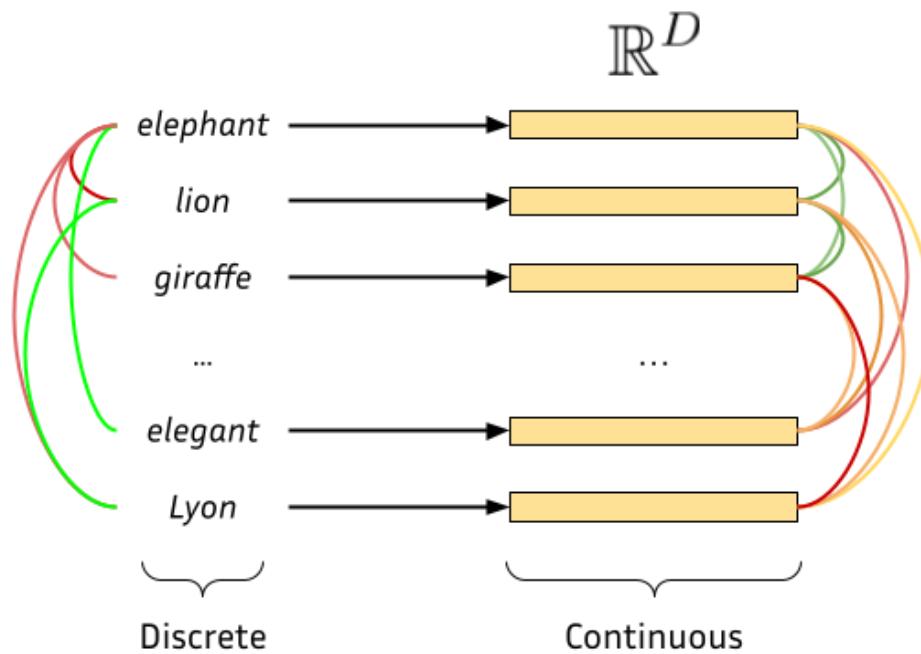
# Zipf's law



# Machine Learning & NLP

# Embeddings

- Vectors that represent textual entities (words, sentences, documents, ...)



# Bag-of-Words Embeddings

Represent a sentence/document by counting words in it.

Example:

*John likes to watch movies. Mary likes movies too.*

=>

```
{John: 1, likes: 2, to: 1, watch: 1, movies: 2, Mary: 1, too: 1}
```

=>

```
[1, 2, 1, 1, 2, 1, 1]
```

# TF-IDF Embeddings

Represent a sentence/document by comparing how frequent a word is in the extract vs. how frequent the word is in general.

$$w_{x,y} = tf_{x,y} \times \log \left( \frac{N}{df_x} \right)$$

**TF-IDF**

Term  $x$  within document  $y$

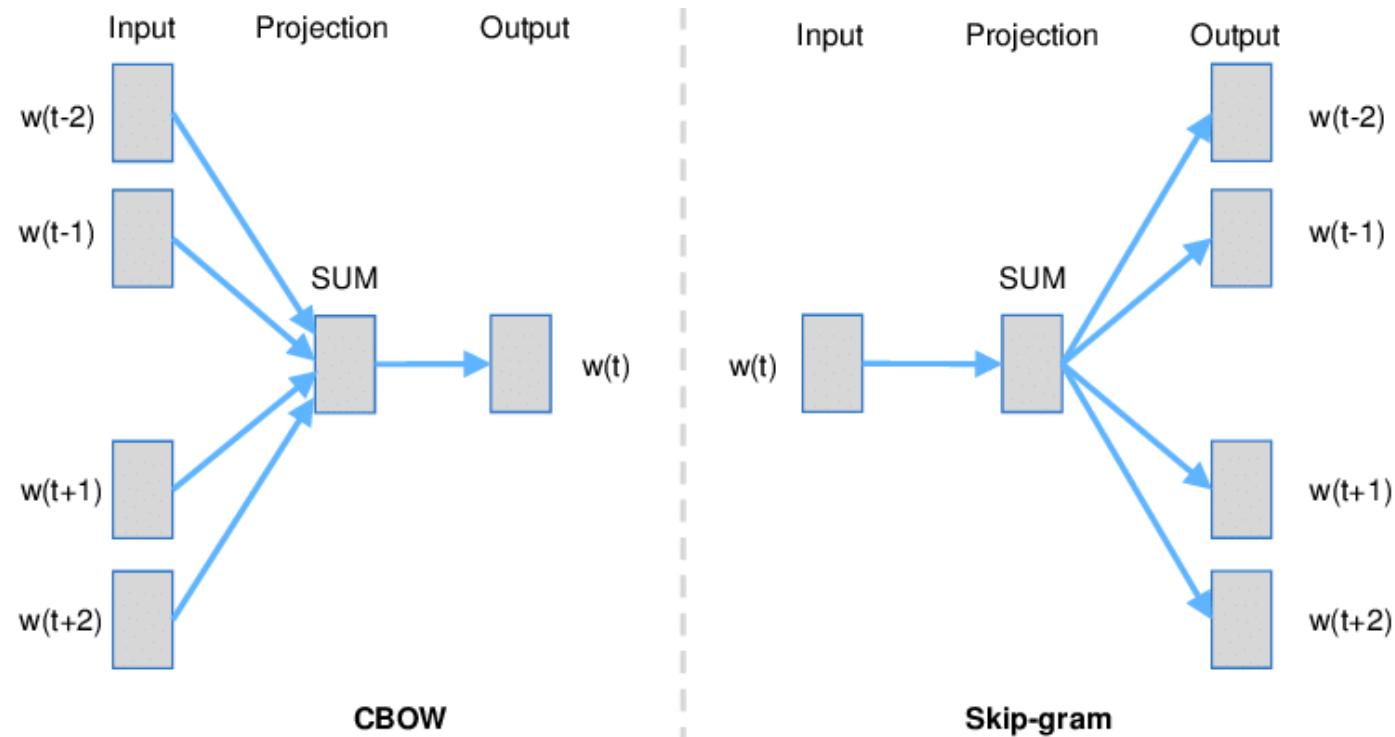
$tf_{x,y}$  = frequency of  $x$  in  $y$

$df_x$  = number of documents containing  $x$

$N$  = total number of documents

# Skip-gram & CBoW

Learn embeddings **without supervision/counting.**

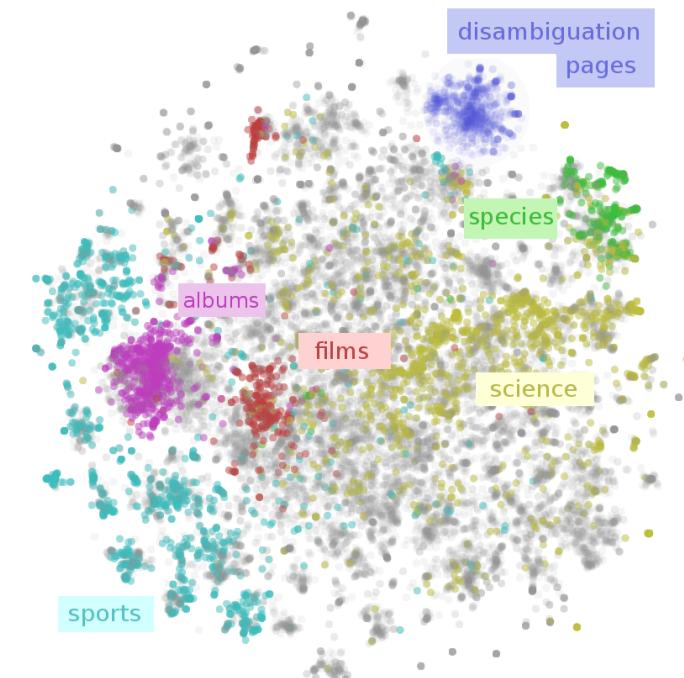


# Static word embeddings

Allow semantically meaningful spaces and can be used as features.

Known static embeddings include:

- GloVe
- FastText
- Word2Vec
- ...



# Deep Learning & NLP

# RNNs

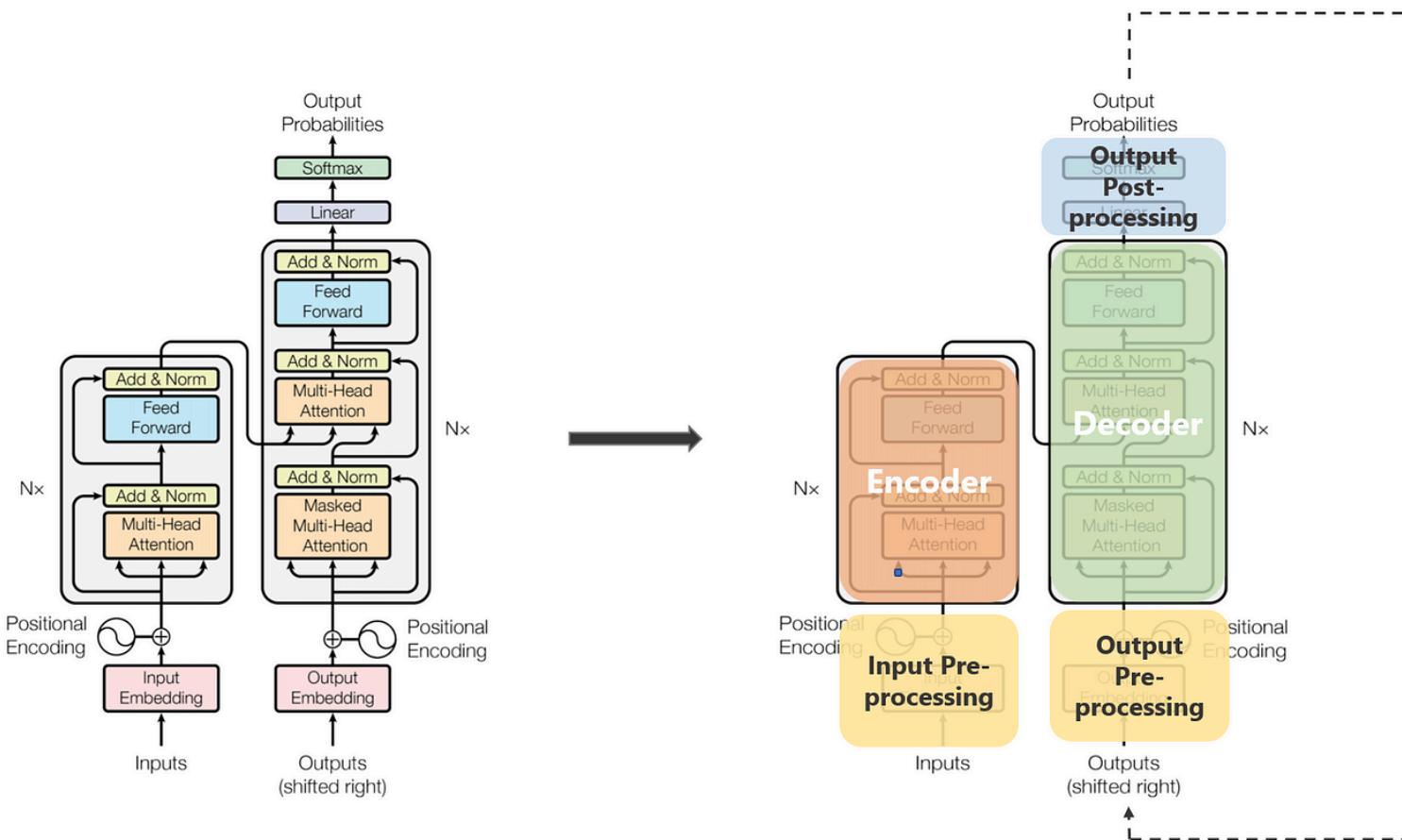
## Recurrent Neural Networks

Neural Networks that are able to process input sequences recurrently.



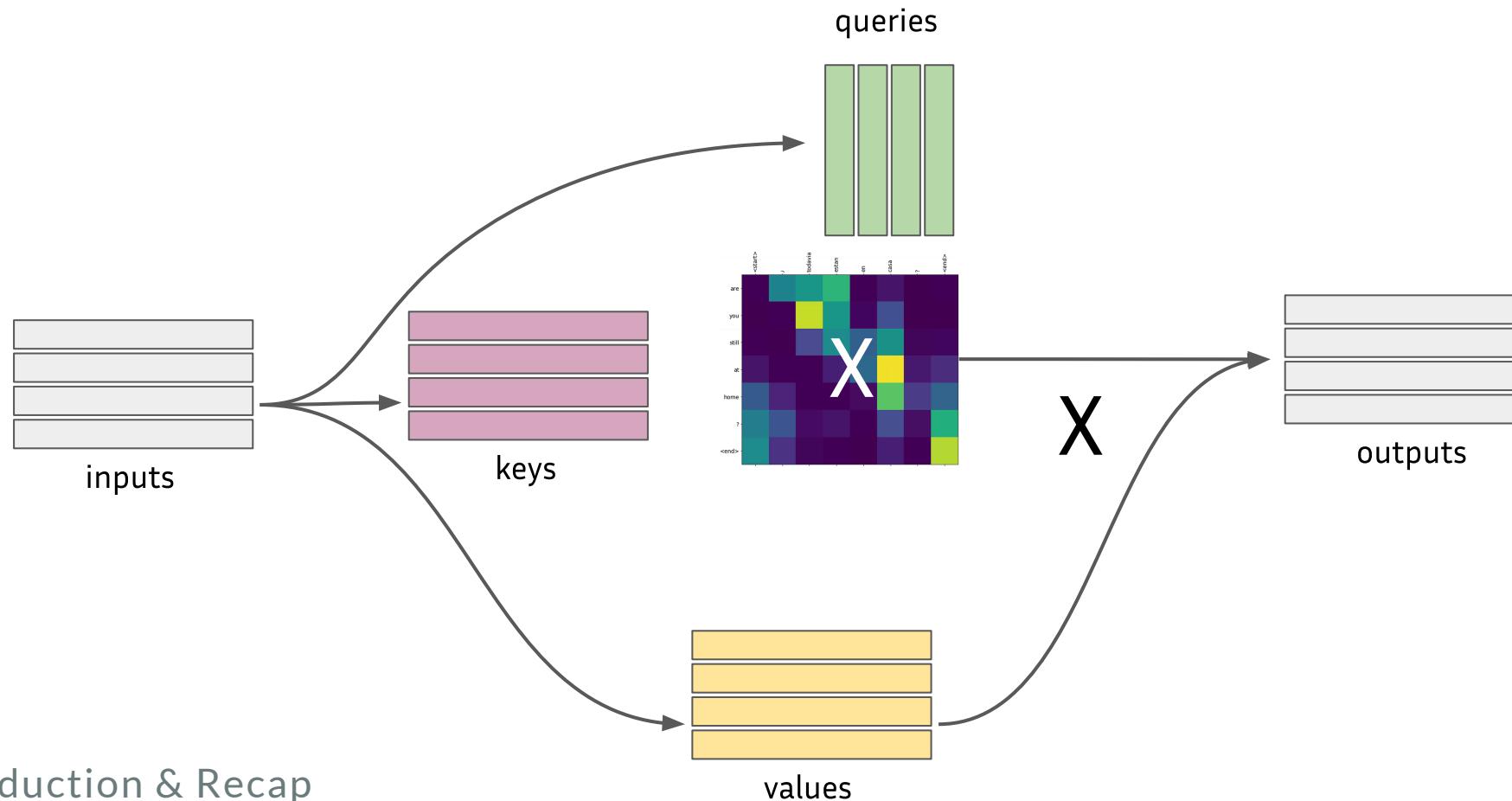
# Transformers

Neural Networks that are able to process input sequences directly.



# Transformers - Self-Attention

Self-attention allows comparing inputs and representing interactions.



# Fine-tuning

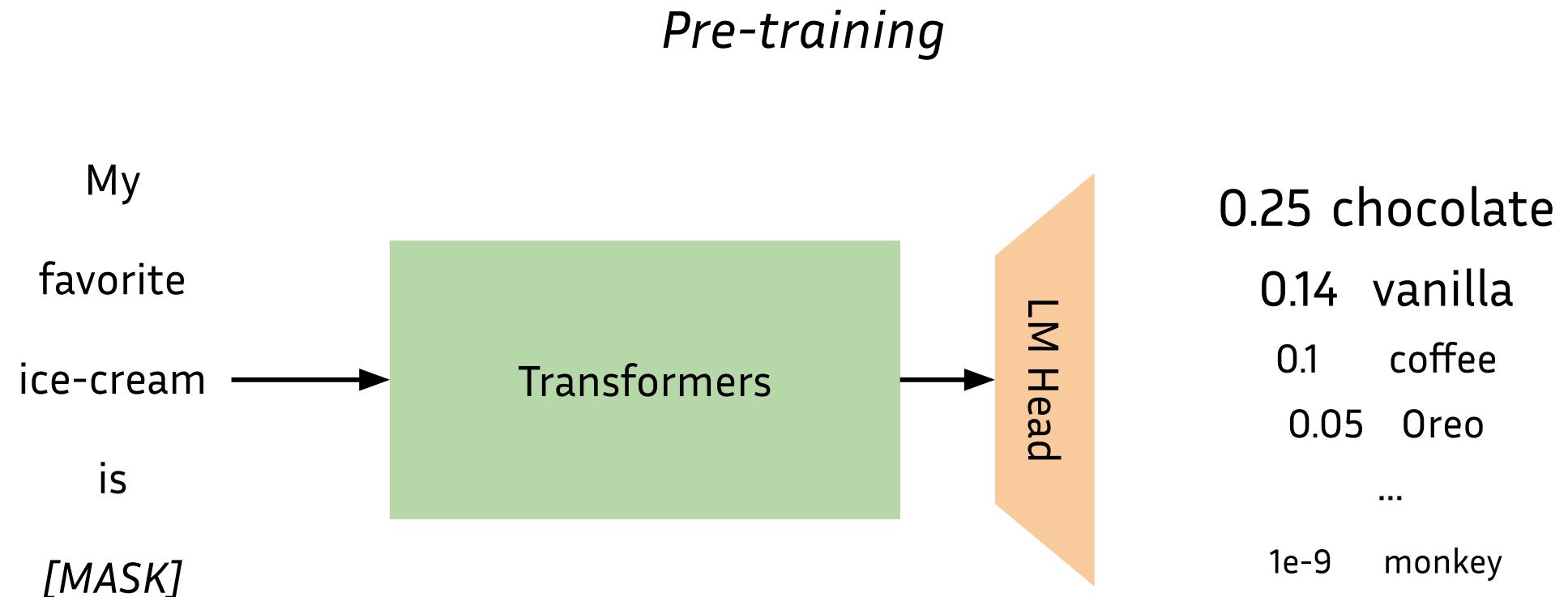
Modern NLP models are built in two separate steps:

- **Pretraining:** models are trained **without supervision** on raw text data (e.g.: next word prediction on all text from Wikipedia).
- **Fine-tuning:** pretrained models are *retrained* on a smaller annotated dataset that matches the final task.

# Fine-tuning - example

1. Download a Transformers language model that was pretrained on data from the Internet (books, Wikipedia, ...) to predict future words
2. Take a list of tweets and mark them as *suspicious* or *safe*
3. Add a classifier on top of the LM that predicts a float in  $[0, 1]$
4. Train the whole model (LM + classifier) to predict *suspicious* or *safe* tweets

# Fine-tuning - example



# Fine-tuning - example



# Questions?

# Lab session