Natural Language Processing

Dr. Manuel Pita & Zuil Pirola

UNIVERSIDADE
LUSÓFONA

What is Natural Language Processing?

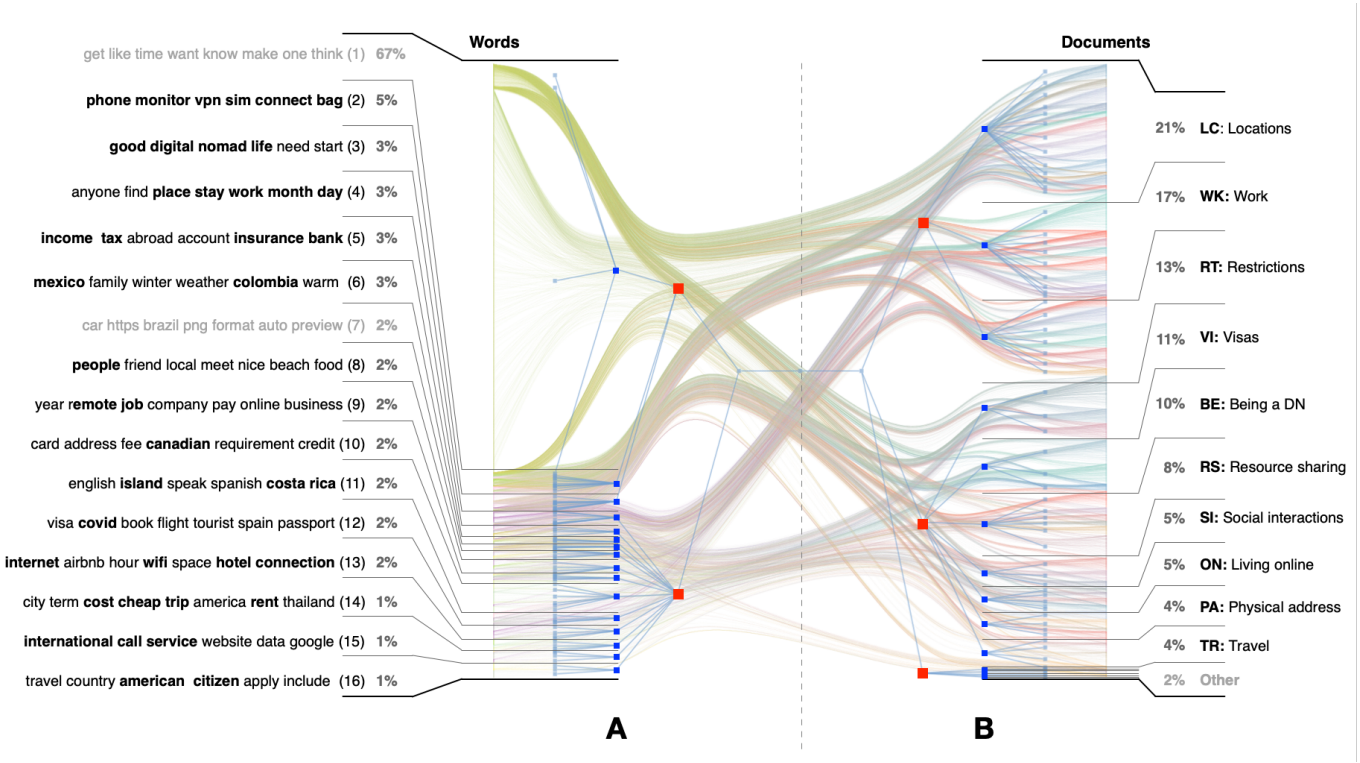
NLP is a field at the intersection of computer science, artificial intelligence, and linguistics that focuses on **enabling machines to understand, interpret, and generate human language**.

Why does NLP matter?

NLP algorithms are at the core of search engines, recommendation systems, chatbot assistants, and the curation of large datasets for research. Indeed, the statistical properties of language are currently used by Large Language Models as a proxy for reasoning and artificial general intelligence*

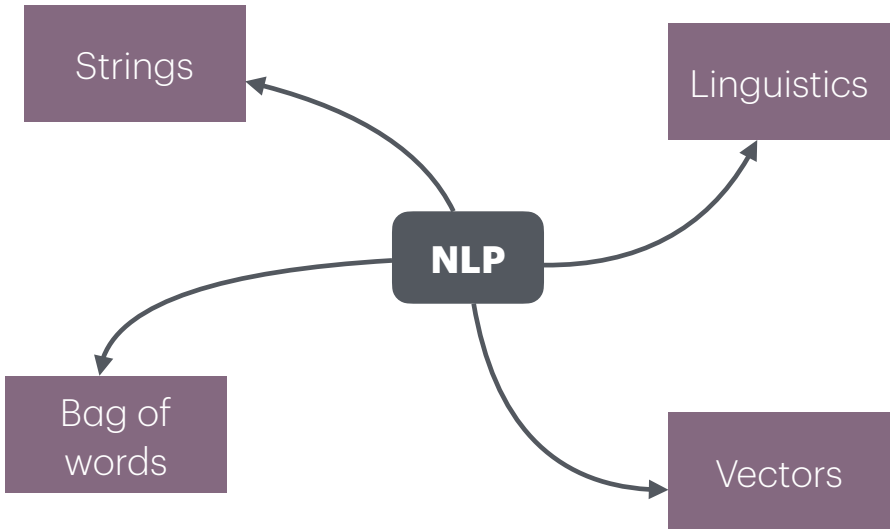
(*) as we will see in this module, there are many dangers associated with such proxy.

One example: Digital nomads on Reddit



Here, all conversations in the **`digitalnomad`** subreddit were grouped into topics, such as locations and searching for work.

Our plan



NLP as STRINGS

Strings

Before we can use NLP, we need to **represent** text in computer code and perform basic **operations** on those text representations.

We usually do this with **string** objects (e.g. in Python or Unix see links below)

We want to be able to count characters, concatenate strings, extract vowels, and ask if a string contains a substring—the list goes on!

The holy grail of string processing is the use of **regular expressions** to query, extract and modify strings according to a specific **input pattern**.

Silly example: select the substrings that start with “e” and end with “e”.

`https://docs.python.org/3/library/string.html`
`https://tldp.org/LDP/abs/html/string-manipulation.html`

Strings

Basic intuition of regular expressions in Python

```
# Ensure that the request was successful
if response.status_code == 200:
    html_content = response.text

    # Use regular expression to find and extract the content between <body> and </body>
    body_match = re.search(r'<body.*?>(.*?)</body>', html_content, re.DOTALL | re.IGNORECASE)

    if body_match:
        # Extracted body content, remove any remaining HTML tags
        body_content = body_match.group(1)

        # Remove remaining HTML tags from the body content
        clean_body_content = re.sub(r'<.*?>', '', body_content)

        return clean_body_content
    else:
        return "No <body> tag found in the HTML page."
else:
    return f"Failed to retrieve the page. Status code: {response.status_code}"
```

In this Python script, the highlighted instruction is a regular expression that extracts the content between the open and closed body tags in an HTML page.

We will be working with regular expressions in the next few weeks.

NLP as BAGS OF WORDS

Bags of words

- BoW emerged when people were building search engines in the 1980s
- **Key idea**: represent (web) documents (strings) using word frequencies
- Webpages with a high frequency of words like **onion** and **oven** differ from documents with a high frequency of **motorbike** and **road**.
- This was important to implement search engines

Bags of words

- The document string is first (split) -> *tokenised*
- Word order, grammar and any language structure is **disregarded**
- This started a line of NLP based on **statistics**, **not on linguistic structure**
- Many thought it would not work, but it does!
- Main application: cluster many documents into conceptual clusters (topics)

NLP as LINGUISTICS

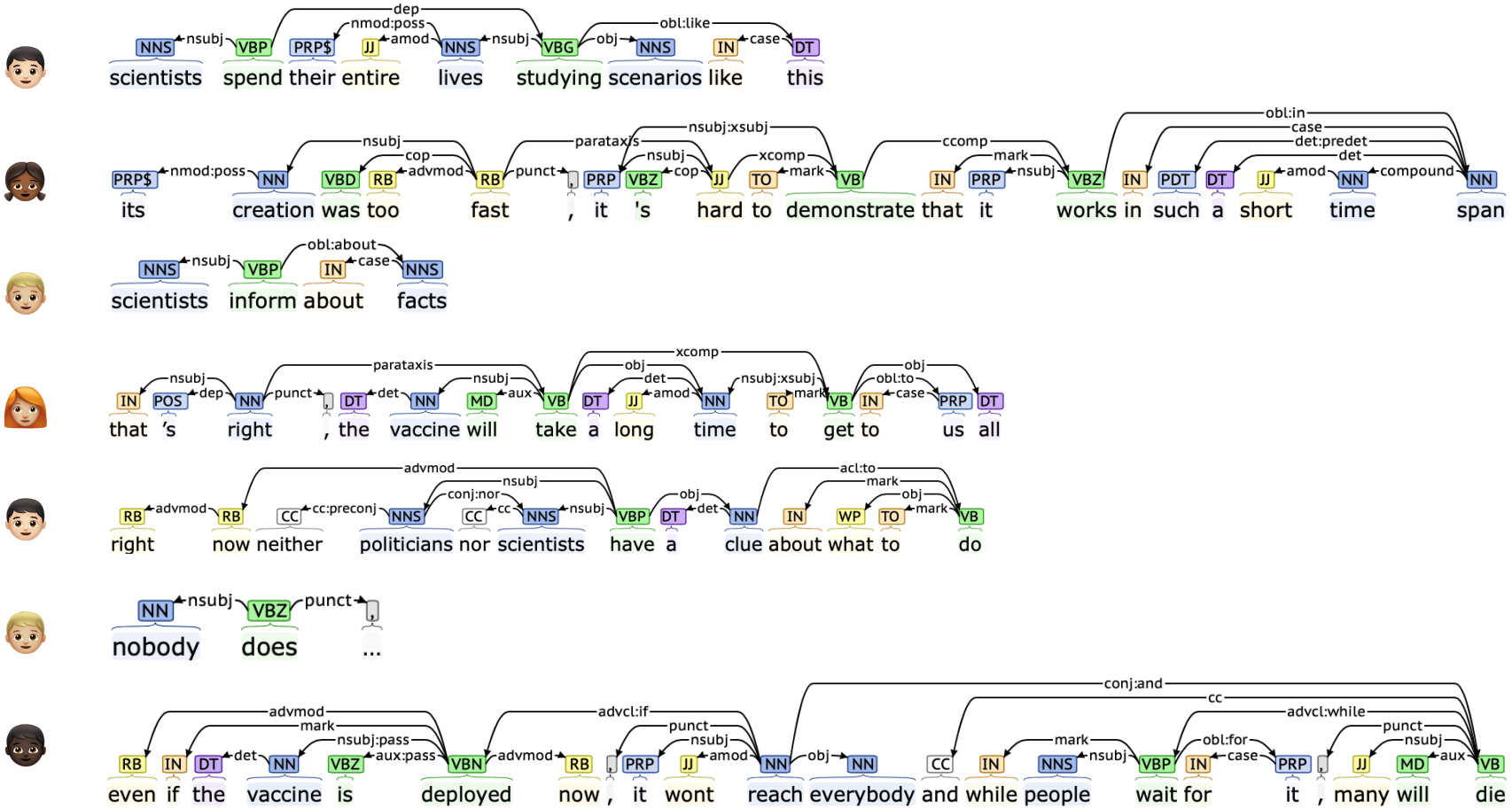
Linguistics

Linguists study language and its meaning by looking at structure (grammar)

Much work in the 90s and 2000s focused on **identifying these grammar elements** from input text (not always easy, indeed often pretty tricky)

Then, **logical inference** was used to understand and generate language

Linguistics



Linguistics

One big problem: people never use language with perfect grammatical rules.

CL inference systems became **more exceptions than rules.**

However, we will learn there are still important uses to NLP tasks such as identifying the subject, verb and object of a sentence.

NLP as VECTORS

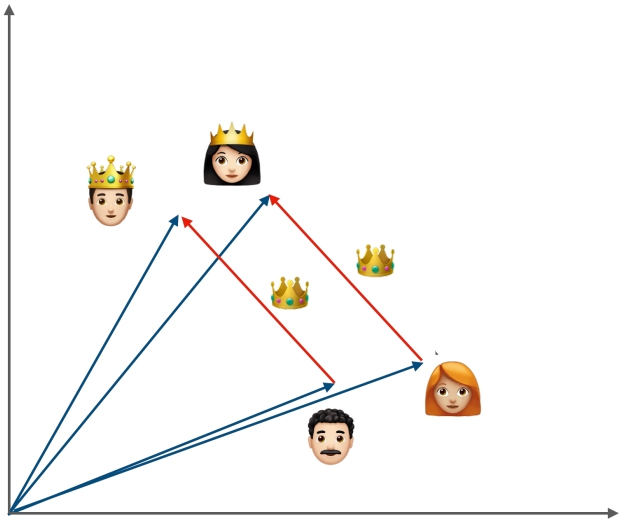
Vectors

Convert words into **vectors.**

Word vectors are calculated using co-occurrence by analysing the frequency with which words appear together in a **given context.**

The word 'king' can appear in the same context as the word 'queen,' but their vectors are not the same.

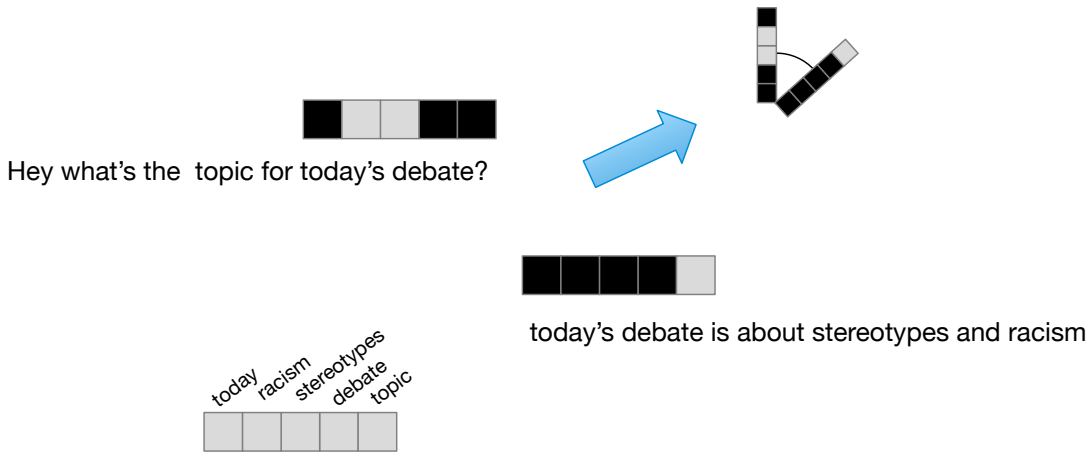
The same can be said for the words 'man' and 'woman.'



Vectors

Converting words into vectors enables **NLP models** to perform mathematical operations, facilitating tasks like **similarity calculations, clustering, and classification**.

If we can represent each **word** as vectors, we can also do the same for **sentences**.



Vectors

With vectors, word representation became contextual, where each word is influenced by its surrounding context.

GPT and others, expand this approach enabling coherent **text generation and complex natural language understanding** tasks through training on **vast amounts of data**.