EDOARDO TOPPETTI

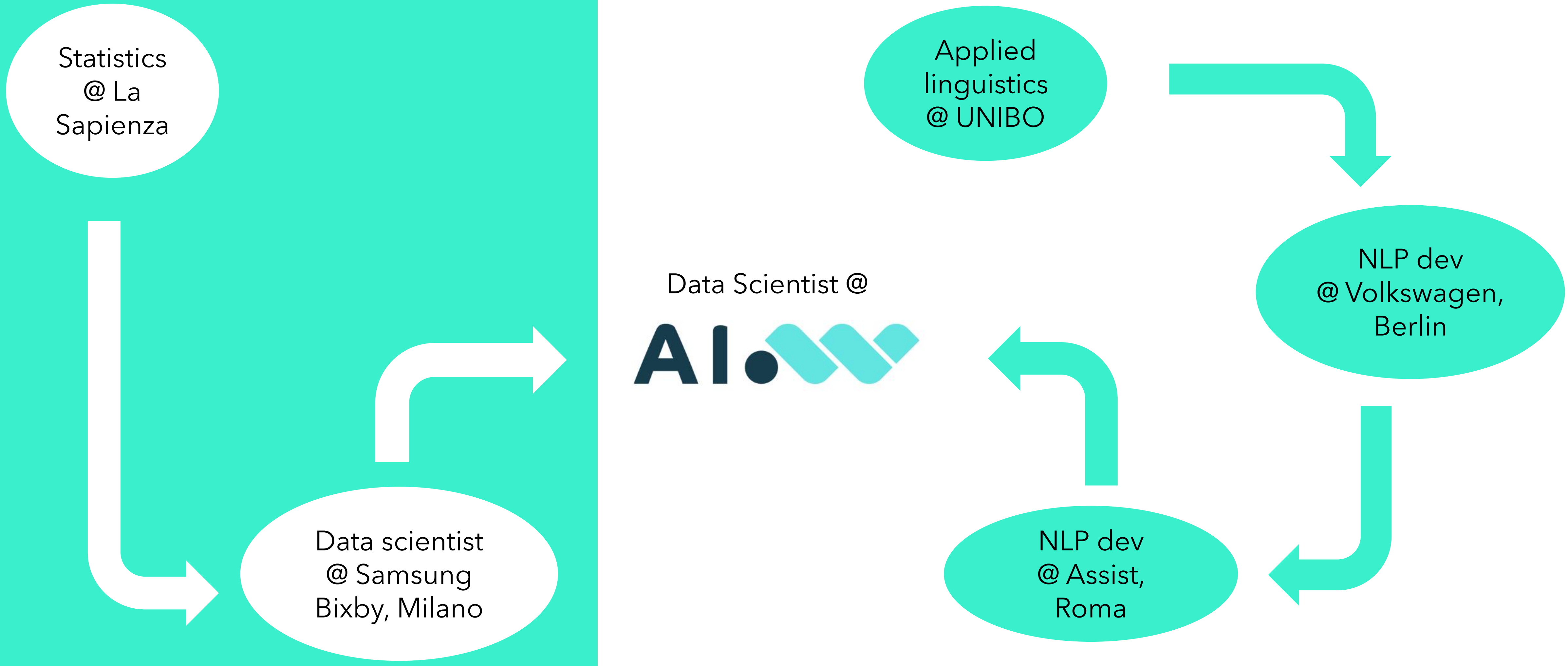
FILIPPO BONORA

# PRACTICAL NATURAL LANGUAGE PROCESSING

# ABOUT US

## EDOARDO

## FILIPPO



Statistics @ La Sapienza

Data scientist @ Samsung Bixby, Milano

Data Scientist @

Applied linguistics @ UNIBO

NLP dev @ Volkswagen, Berlin

NLP dev @ Assist, Roma

# INDEX

**1) The main concepts of NLP**

**2) Math with words**

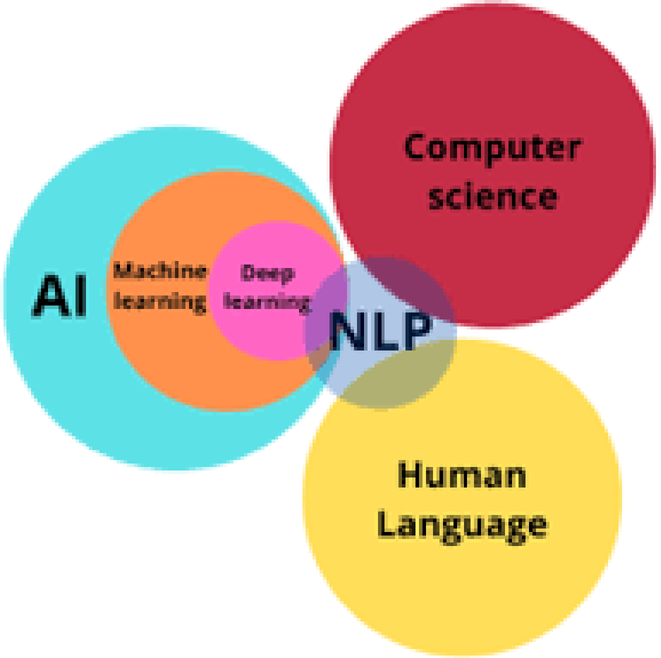
- Word Embeddings
- Big language models

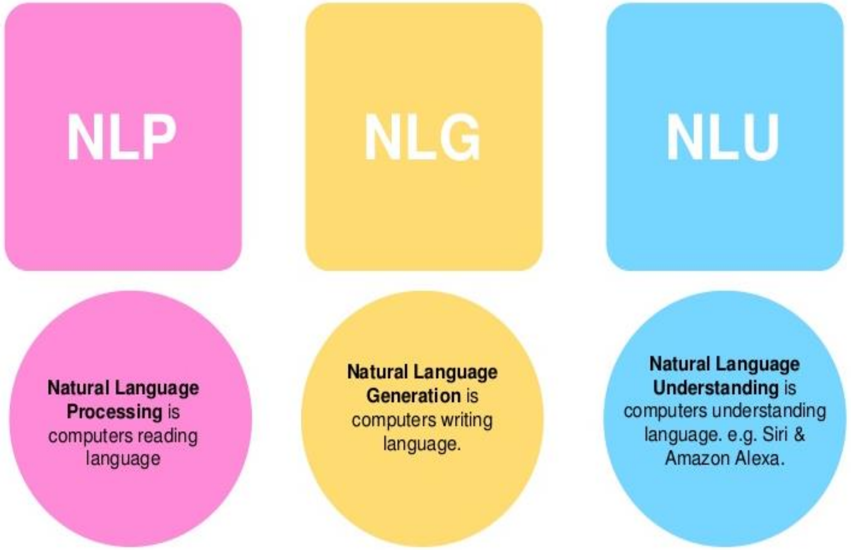
**3) NLP in practice**

- NLP main tasks
- Use case: Reputational Risk Index

# WHAT IS NLP

**Natural language processing (NLP)** is a set of **techniques** borrowed from different disciplines such as **linguistics**, **computer science**, and **artificial intelligence,** concerned with the ***interactions between computers and human language***, in particular how to program computers to process and analyse large amounts of natural language data.




NLP

NLG

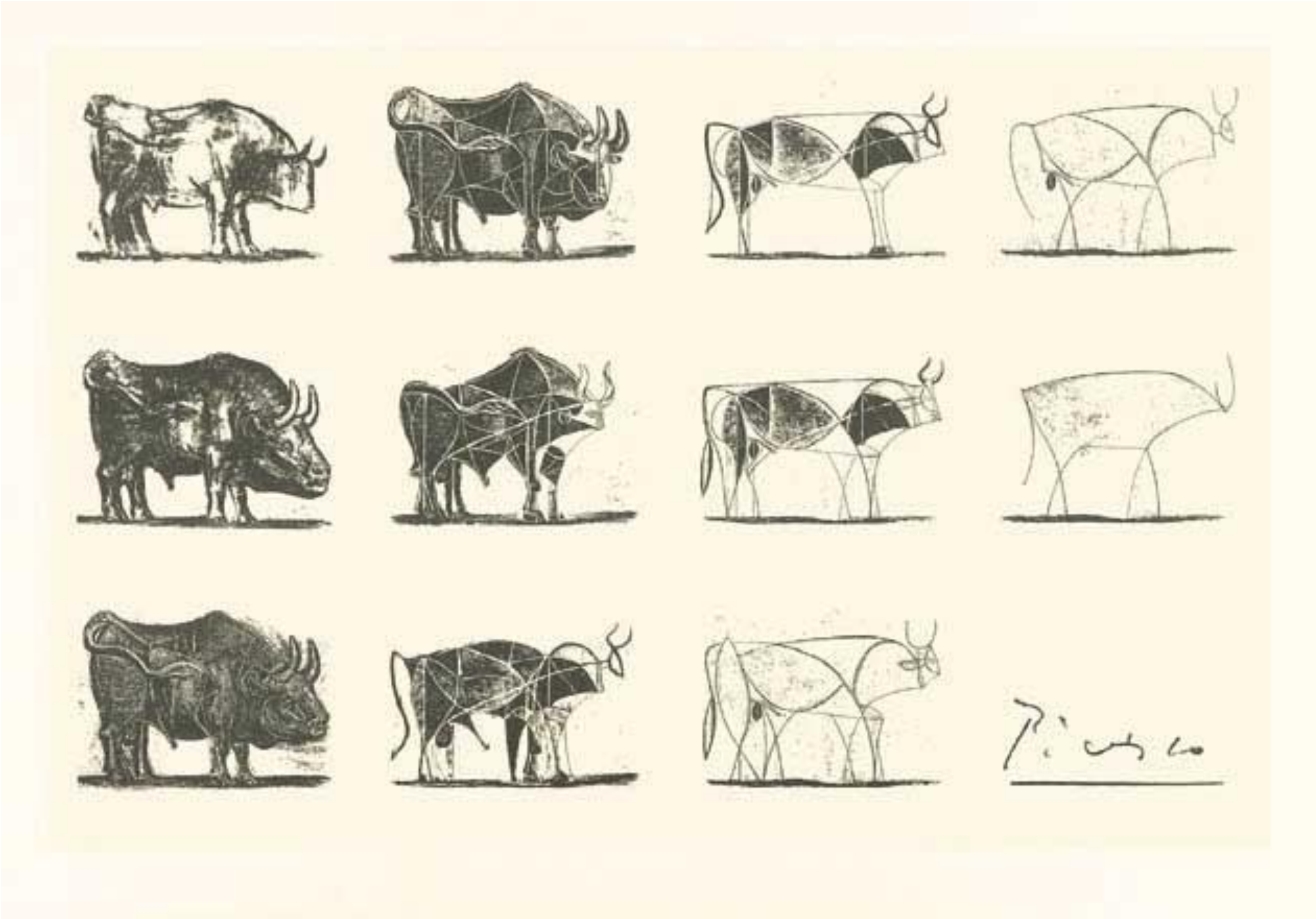
NLU

**Natural Language Processing** is computers reading language

**Natural Language Generation** is computers writing language.

**Natural Language Understanding** is computers understanding language. e.g. Siri & Amazon Alexa.

# ENCODING



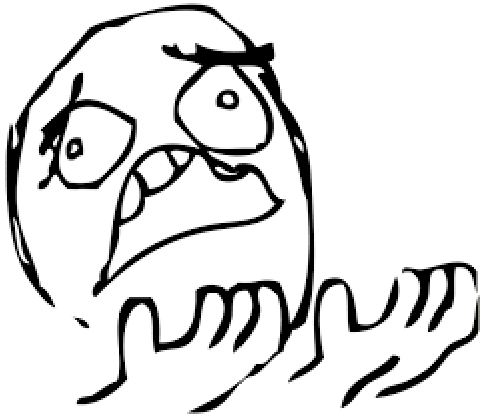
"$$f(\mathbf{Bull}) = \textit{stylized representation of the bull}$$
$$\textit{s.t. } f \textit{ is the creative genius of Picasso}$$

***“Encoding involves the use of a code to change original data into a form that can be used by an external process”***

*Picasso needs was to catch the bull essence... Same way we need to catch words or sentences meanings and relations between each other words or sentences*

# HOW TO REPRESENT WORDS?

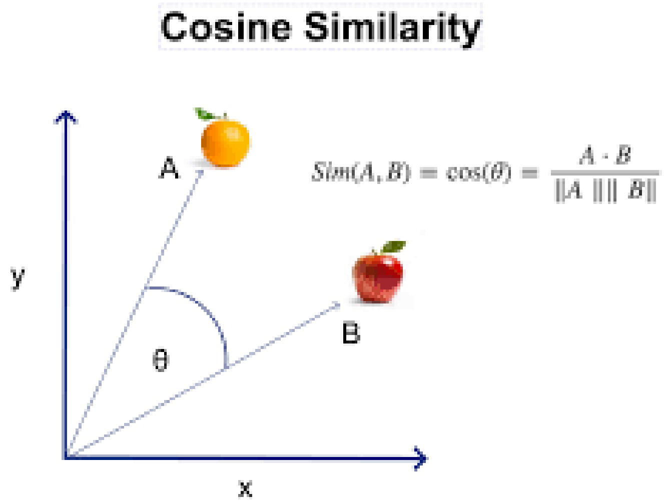
## EMBED THEM INTO VECTORS!!



## OK… BUT WHY?

Because vectors are **multidimensional** and **comparable entities**, while with words is tough for a computer.

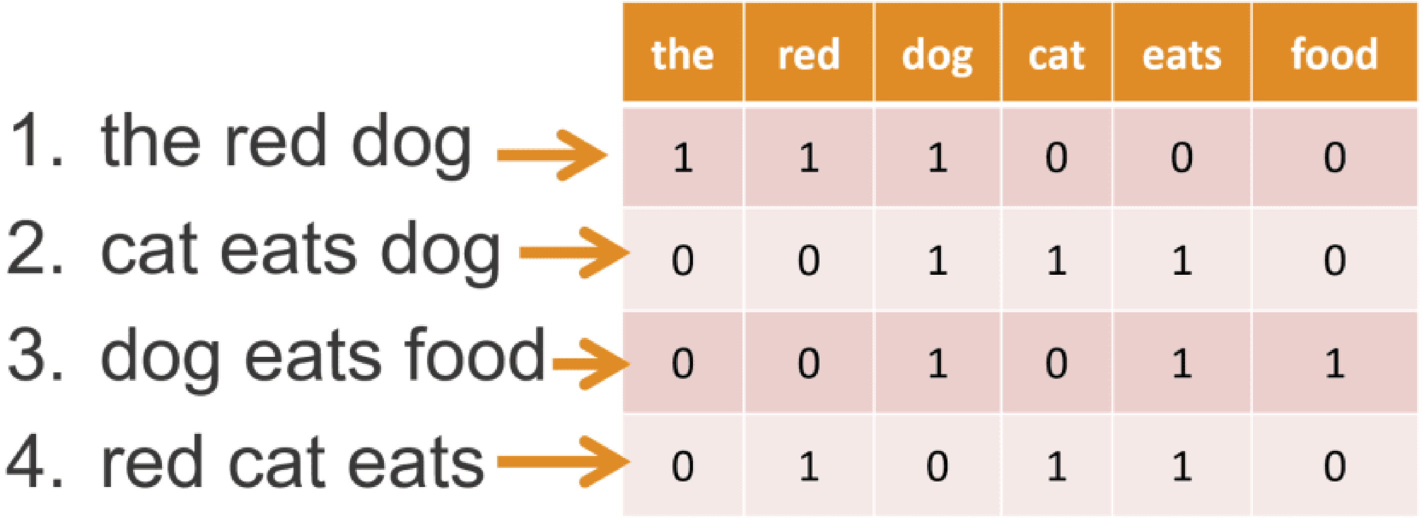
We want to vectorize words to measure the **semantic similarity** of each other



# HOW TO REPRESENT WORDS?
# BAG OF WORDS!

**Distributional hypothesis:**

*"You shall know a word by the company it keeps"* J.R.Firth 1957



1. the red dog →
2. cat eats dog →
3. dog eats food →
4. red cat eats →

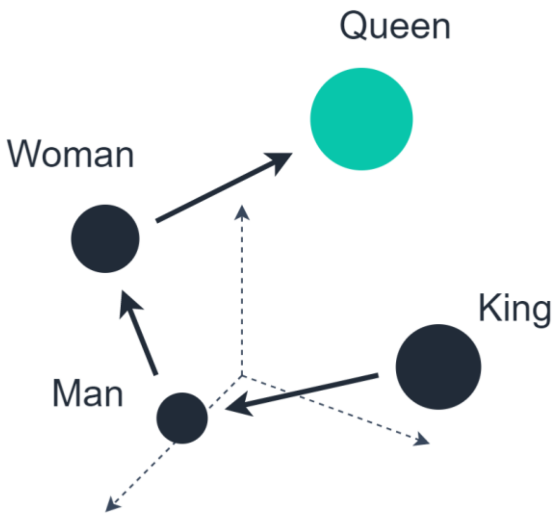
| the | red | dog | cat | eats | food |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |

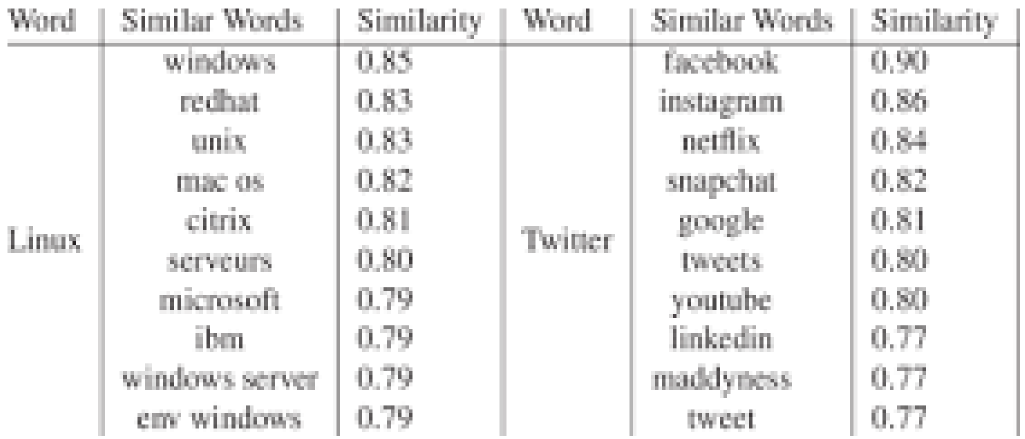
# DENSE VECTORS: WORD2VEC

But a **sparse vector is not good** as a feature for ML. We need DENSE one!

The word2vec algorithm uses a simple neural network to learn word associations from a large corpus of text (es. Wikipedia). Dimensionality is up to the dev, depends on needs

$$Queen \sim King \; - Man + Woman$$





| Word | Similar Words | Similarity | Word | Similar Words | Similarity |
|---|---|---|---|---|---|
| Linux | windows | 0.85 | Twitter | facebook | 0.90 |
| | redhat | 0.83 | | instagram | 0.86 |
| | unix | 0.83 | | netflix | 0.84 |
| | mac os | 0.82 | | snapchat | 0.82 |
| | citrix | 0.81 | | google | 0.81 |
| | serveurs | 0.80 | | tweets | 0.80 |
| | microsoft | 0.79 | | youtube | 0.80 |
| | ibm | 0.79 | | linkedin | 0.77 |
| | windows server | 0.79 | | maddyness | 0.77 |
| | env windows | 0.79 | | tweet | 0.77 |

# LIMITS OF W2V APPROACH

Every word has its vector, but words are **POLYSEMIC !** In addition, every language is full of **IDIOMS!**

**It’s raining cats and dogs!!** **I really love cats ♥**

[0.43, 0.976, 0.882, …, 0.921]

How can be efficient to represent the word "cats" in the same way, every time it comes up?

# BERT

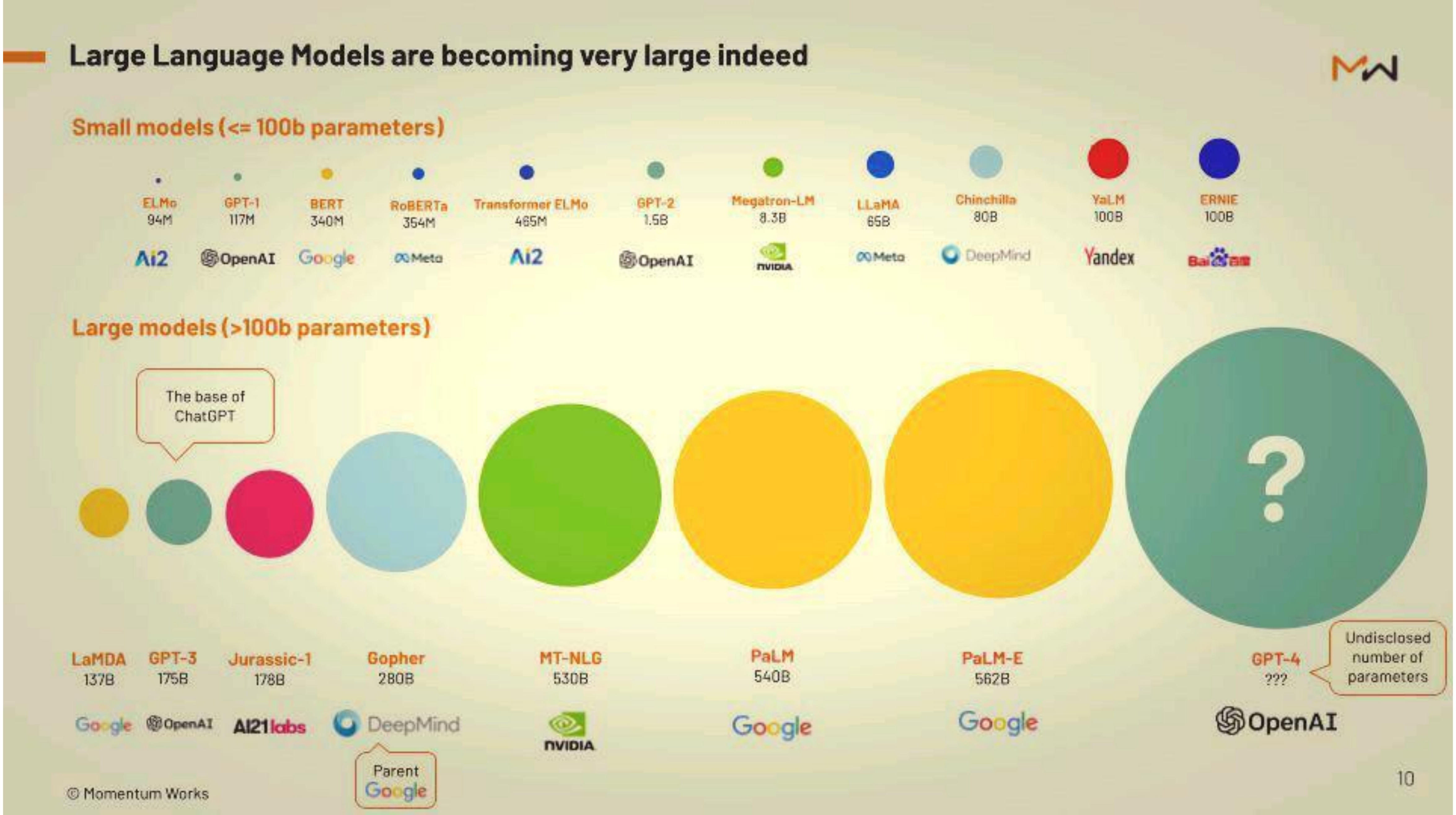
## BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

**Transformers architectures** and **attention mechanism**, developed around 2018, have constituted a breakthrough in NLP community.

**It's rainings cats and dogs!!**
[0.43, 0.976, 0.882, …, 0.921]

**I really love cats ♥**
[0.697, 0.161, 0.292, …, 0.001]

Compared to previous methods, the model produces **contextual embeddings,** say, it **takes polisemy into account**.

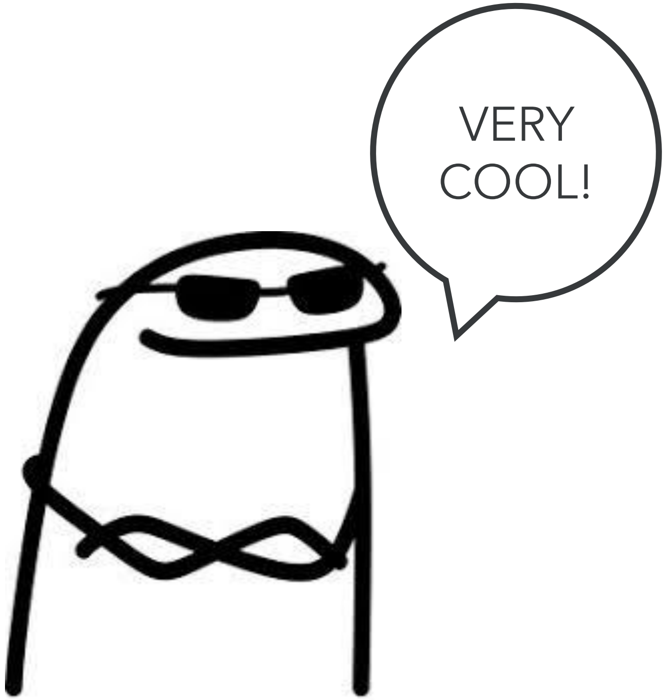


## Large Language Models are becoming very large indeed

# WELL.. NOW I GOT VECTORS WHAT I CAN DO?

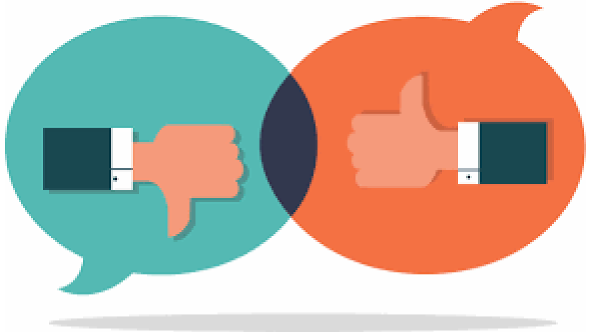
## HERE SOME NLP POPULAR TASKS:

- TEXT CLASSIFICATION
  (E.G. SENTIMENT ANALYSIS)
- MACHINE TRANSLATION
- SUMMARIZATION
- INFORMATION RETRIEVAL
- NAME ENTITY RECOGNITION



*EXCUSE ME SIR, I DON'T WANT TO BE UNPOLITE BUT WE ARE ECONOMISTS… DOES THIS 🍌 TRASH INTEREST US?*

# NLP & ECONOMICS: REPUTATIONAL RISK



**Reputational risk** is the **damage** that can occur to a business when it fails to meet the **expectations of its stakeholders**

***"It takes many good deeds to build a good reputation, and only one bad one to loose it."***

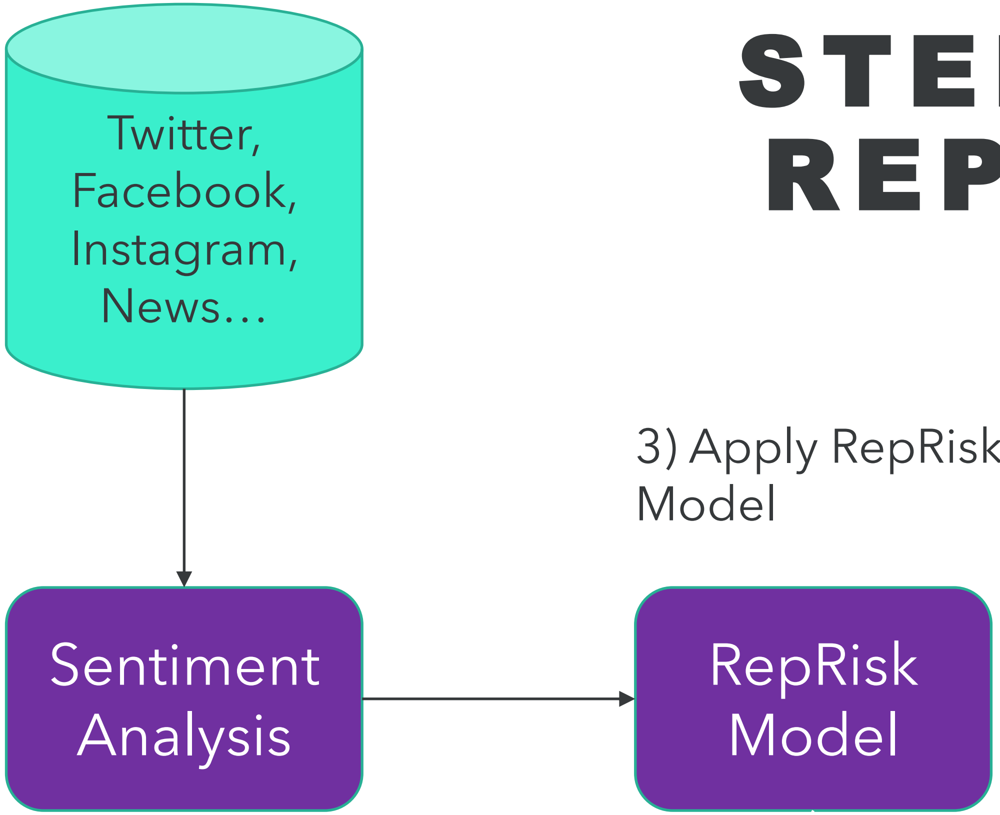
*cit. Benjamin Franklin*

*Seems to be important measure the company reputation! How can we do it?*

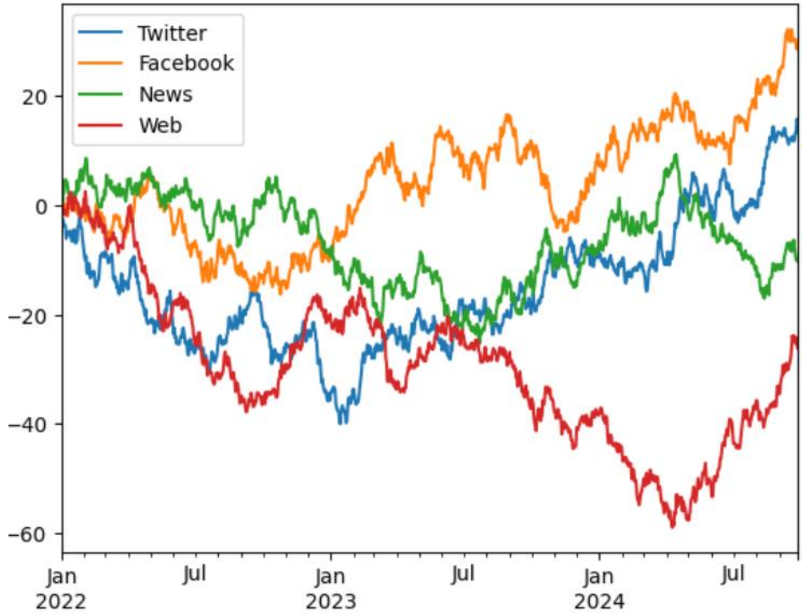
# STEP BY STEP – REPUTATIONAL RISK

1) *Gather text company related contents*

2) Compute sentiment on each content



3) Apply RepRisk Model



**Reputational Risk Model - Some maths**

Normalization

$$S(x) = u(x) = \begin{cases} x^{\alpha}, & x \geq 0 \\ -\lambda(-x)^{\alpha}, & x < 0 \end{cases}$$

Avarage scores

$$AS(S_t) = \frac{\sum_{i=1}^{n} S_t}{n}$$

Compute index for day t and source m

$$Rep_m(t) = \sum_{t' < t} Rep_m(t') + a * AS_m(S_t)$$

ANY QUESTIONS???

# AND NOW… HANDS-ON 🤓

**For further questions:**

**e.toppetti@almawave.it**

**f.bonora@almawave.it**

**Find the code at our Github**