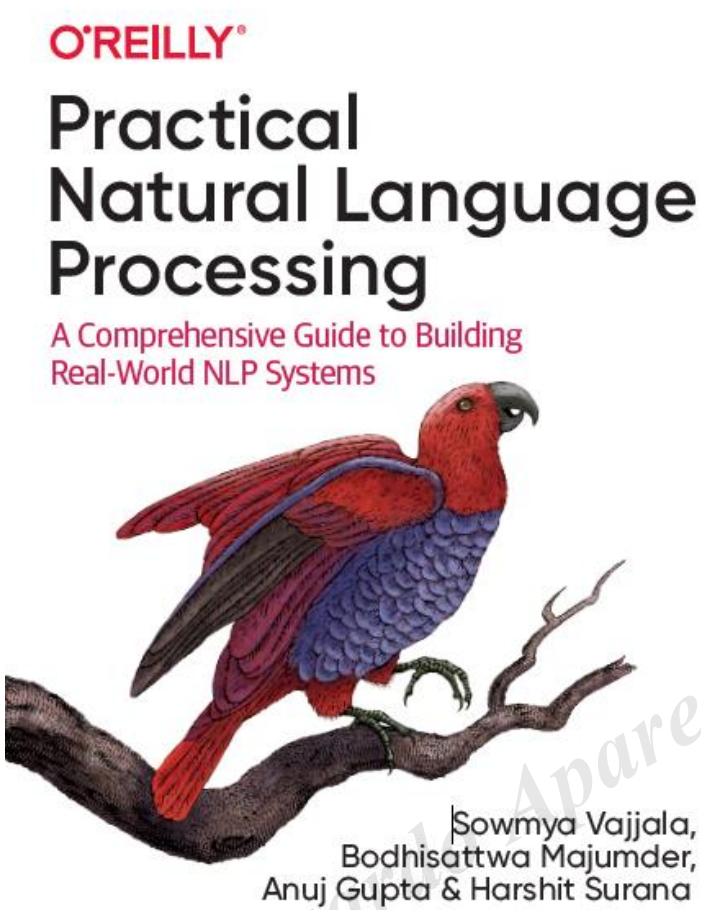
MBA USP ESALQ

# TEXT MINING, SENTIMENT ANALYSIS AND NLP

Prof. Dr. Jeronymo Marcondes

# Introduction




# Plan of attack

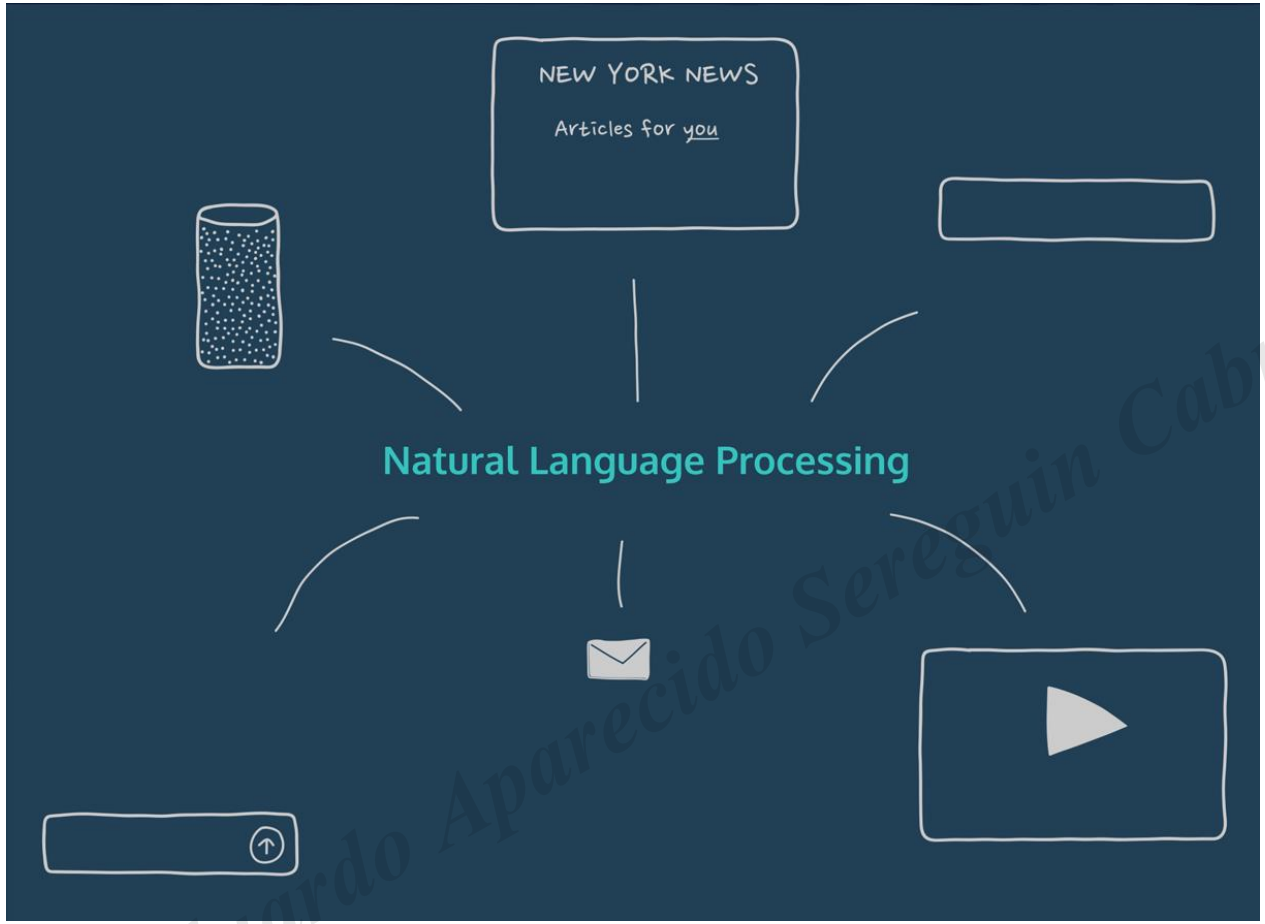
1. Basic concepts of text mining:
- Text extraction techniques;
- Tidytext use;
- Bag of words
- Important concepts: stop words, stemming, etc.
- N grams
- Topic modelling

2. TF-IDF and Sentiments analysis

# Introduction

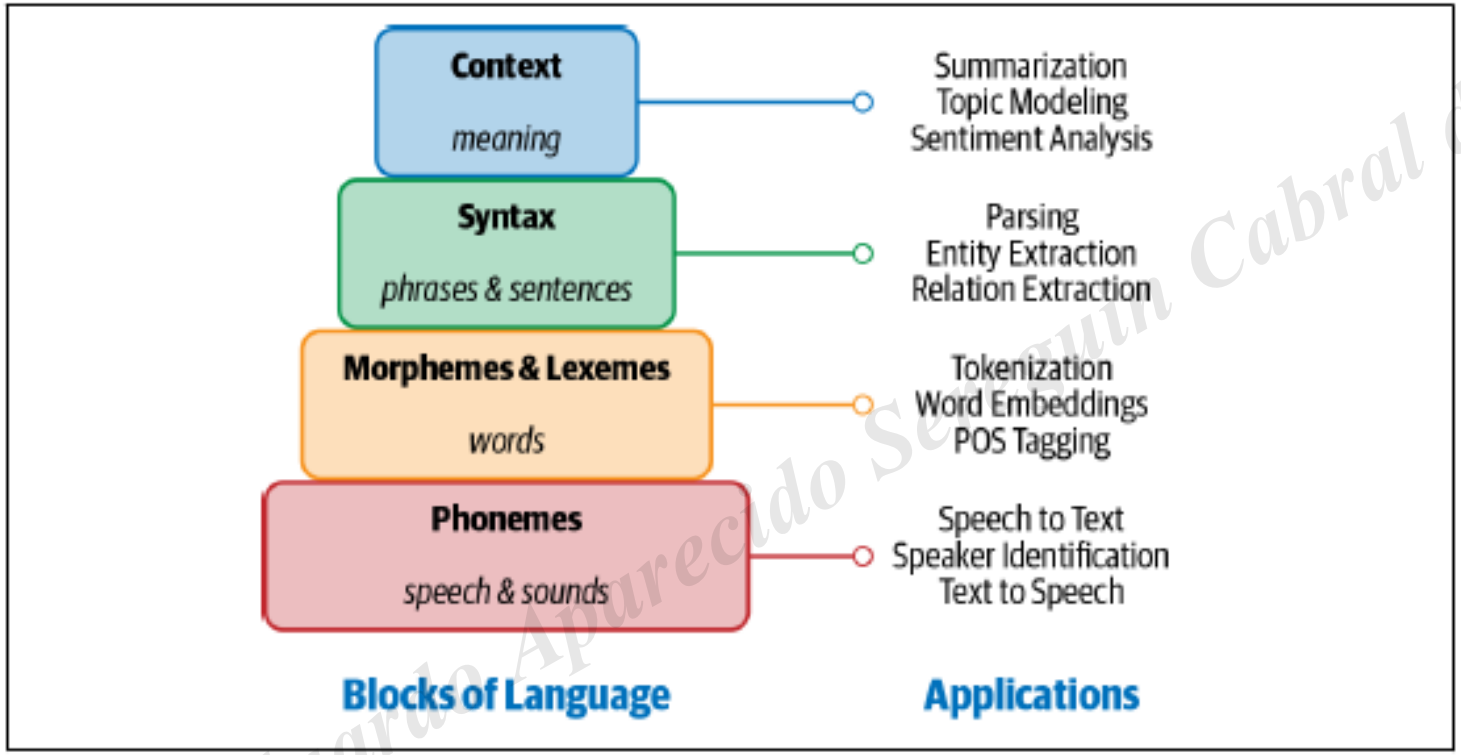
- What are we talking about?
- What is NLP?
- Why NLP? Difference regarding the Computer Programming Language
- What is the idea of functioning?

# Sources for NLP




Source: https://medium.com/swlh/nlp-text-preprocessing-techniques-ea34d3f84de4

# Introduction



# Introduction

- Difficulties with NLP:

1. Sarcasm
2. Common domain knowledge
3. Ambiguity

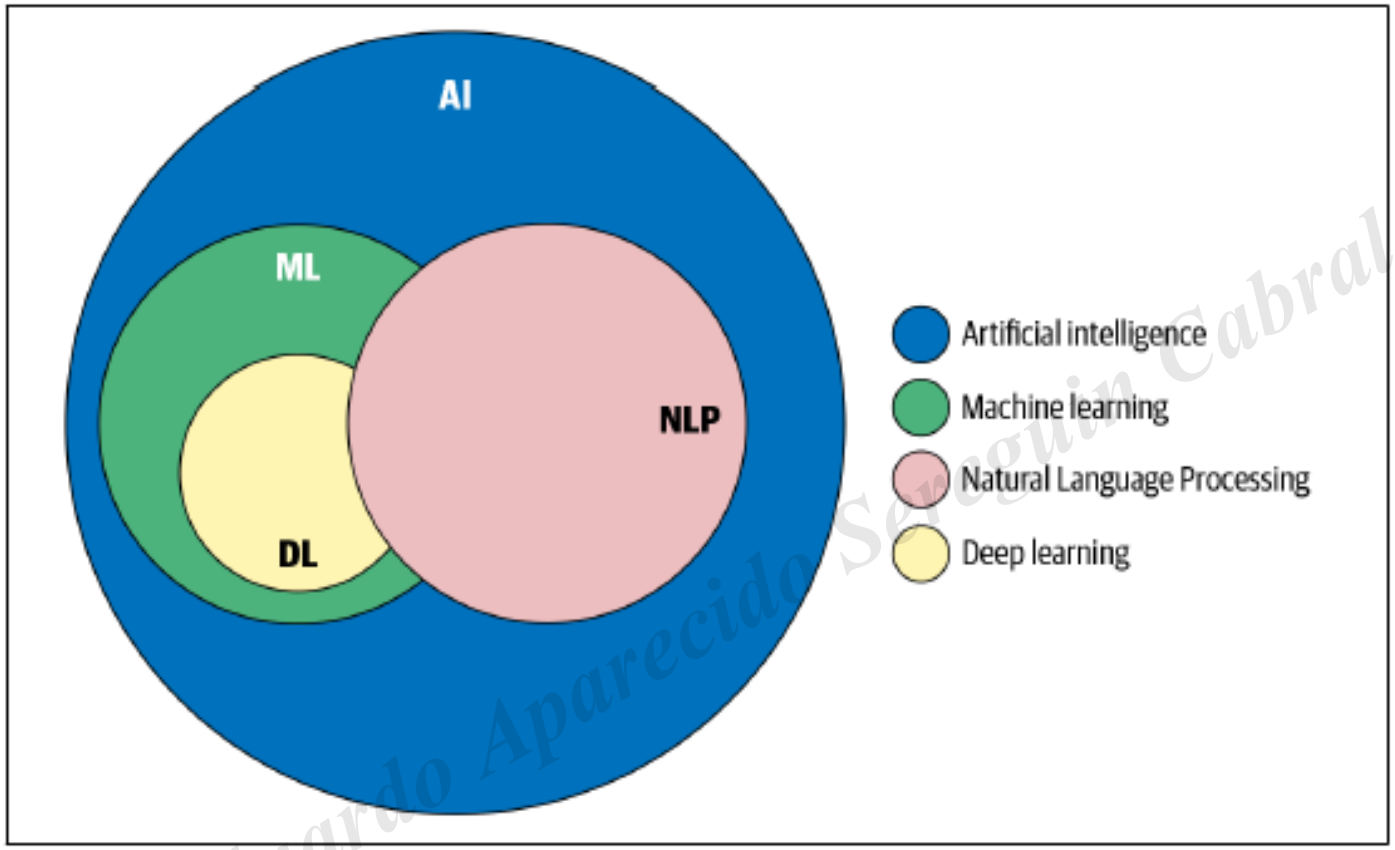
# Introduction

- Approaches :

1. Heuristic

2. Machine Learning and Deep Learning

# Introduction



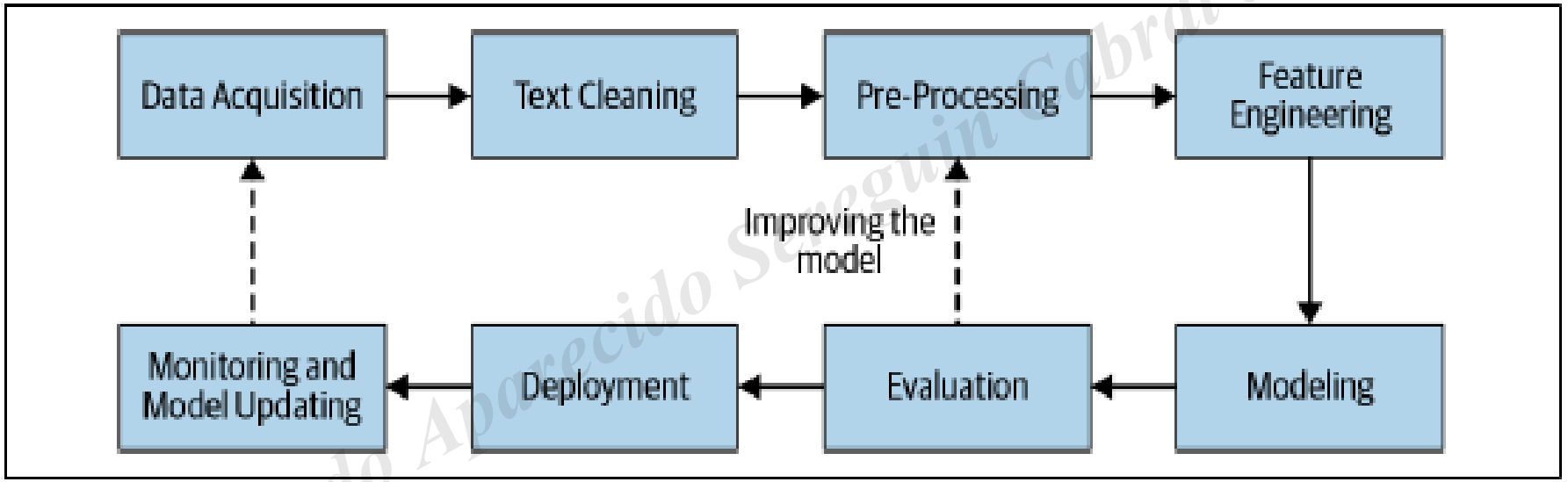
# Origin

- Alan Turing
- How to identify a machine?
- Until the 80s – several rules
- Begining of Machine Learning

# Uses

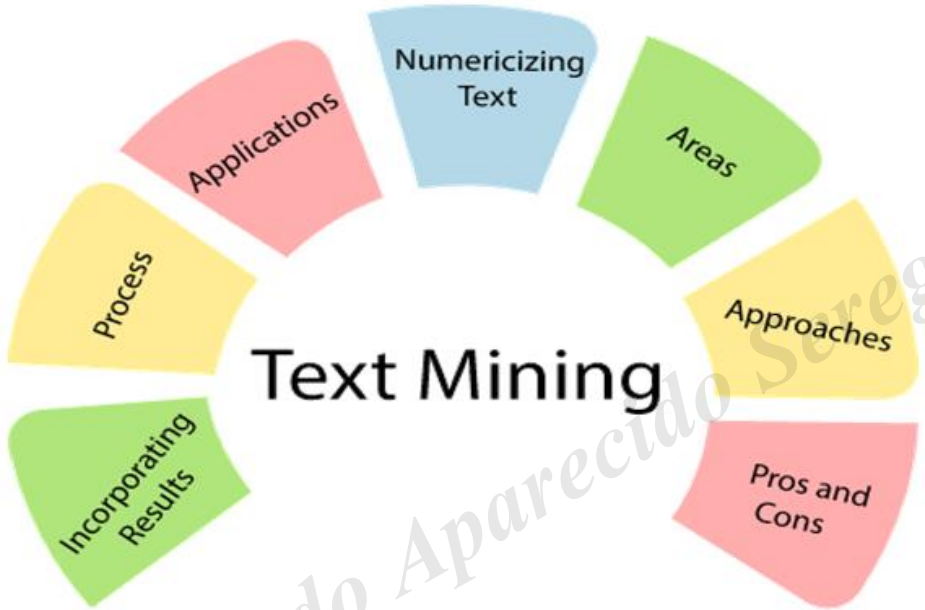
- Internet search engines
- Chatbots
- Virtual assistants
- Text analysis - article example
- Spam filter

# NLP Pipeline



# Text Mining

- What is text mining?




Source: https://www.javatpoint.com/text-data-mining

# Tidytext and Tidydata

- Hadley Wickham:

1. Each variable is a column
2. Each observation is a line
3. Each observational unit is a chart

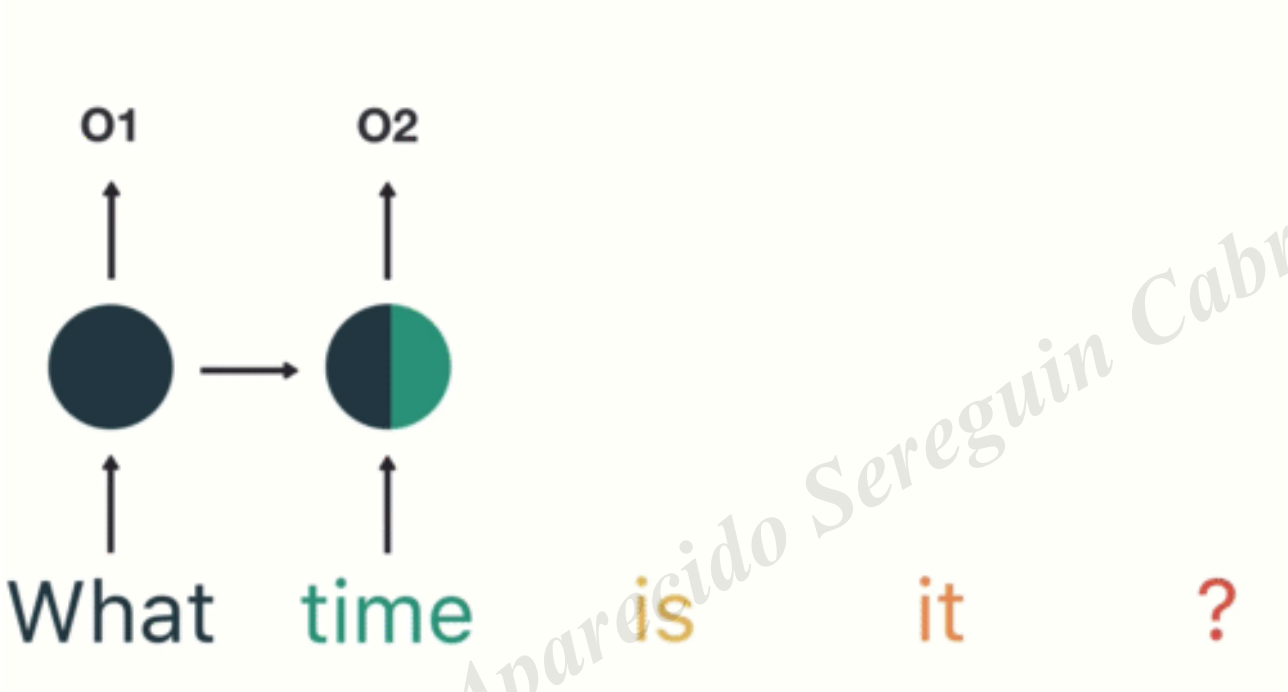
- A token by line!

# Token

TOKEN = For organized text mining, the stored token on each line is a single word, but it can also be a n-gram, sentence, or paragraph.

Text's lowest unit that matters!

# Tokening




Source: https://aiplusinfo.com/

# Token

- "I will be a data scientist"

| | |
|---|---|
| 1 | I |
| 2 | will |
| 3 | be |
| 4 | a |
| 5 | data |
| 6 | scientist |

- R has very good functions for this process in the tidytext package

# Unnest Token

```
text <-c("Because I could not stop for Death -",
"He kindly stopped for me -".
"The Carriage held but just Ourselves -",
"and Immortality")

text_df <-tibble(line = 1:4, text = text)

text_df %>% unnest_tokens(word, text)
```

```
text_df %>%
  unnest_tokens(word, text)
#> # A tibble: 20 × 2
#>     line word
#>    <int> <chr>
#>  1     1 because
#>  2     1 i
#>  3     1 could
#>  4     1 not
#>  5     1 stop
#>  6     1 for
#>  7     1 death
#>  8     2 he
#>  9     2 kindly
#> 10     2 stopped
#> # … with 10 more rows
```

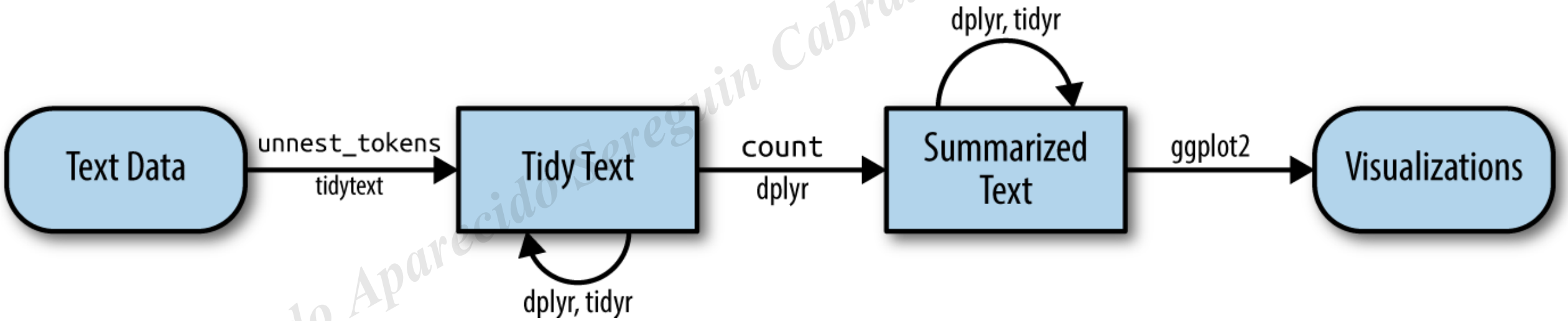
# Unnest Token

- Notice how useful the function is:

1. Transforms all letters into lower case;
2. Removes punctuation;

- Other packages - must be done manually.

# Unnest Token



# Bag of words

- Very interesting for understanding the text
- Verifies and counts the terms frequency, or binary variable that indicates presence or not
- It is not the approach in question – we can get into something similar to dplyr

Bag-of-words (BoW) is a statistical language model used to analyze text and documents based on counting of words. The model does not consider the order of words in a document.

# Bag of Words

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 |

https://www.medium.com/

# Bag of words

- This is very useful counting for several problems:

1. Verifying more common terms;
2. Verifying less common terms;
3. Verifying similarity between terms and their amount in the text.

- Bag - ignores any order
- We will show the TF-IDF on next class

# Data Cleaning

- Many words must be removed from the text, depending on what is being extracted (data cleaning):

1. Numbers;
2. Emoji;
3. Special characters
4. Blank space
5. etc

# Stop Words

- Do every word have meaning? Some have little or any: stop words
- https://gistgithubub.com/alopes/5358189
- from, to, what, etc

# Stop Words

- Large amount of available lists: using premade lists

```
library(stopwords)
length(stopwords(source = "smart"))
length(stopwords(source = "snowball"))
length(stopwords(source = "stopwords-iso"))
```

# Stop Words

- Acceptable: done list with adjustments
- Does every stop word have to be removed?
- Anti join

```
Data_frame %>%  anti_join(get_stopwords(source = "snowball"))
```

# Stemming

- Stemming is the process of reducing flexed words (or sometimes derived words) to their stem, base, or roots, usually a written word's form.

| | |
|---|---|
| amigos | amig |
| amigas | amig |
| amizade | amizad |
| carreira | carr |
| carreiras | carr |

Source: https://www.alura.com.br

Here are some examples in English:

| | |
|---|---|
| information | inform |
| informative | inform |
| connect | connect |
| connected | connect |
| connection | connect |

# Stemming

- Difference from lemmatization


Here is an example in English:


https://www.computersciencemaster.com.br/como-reduzir-uma-palavra-ao-seu-radical-em-python-stemming/

# Stemming

- Different approaches
- Should we make stemming?

# Token n-gram

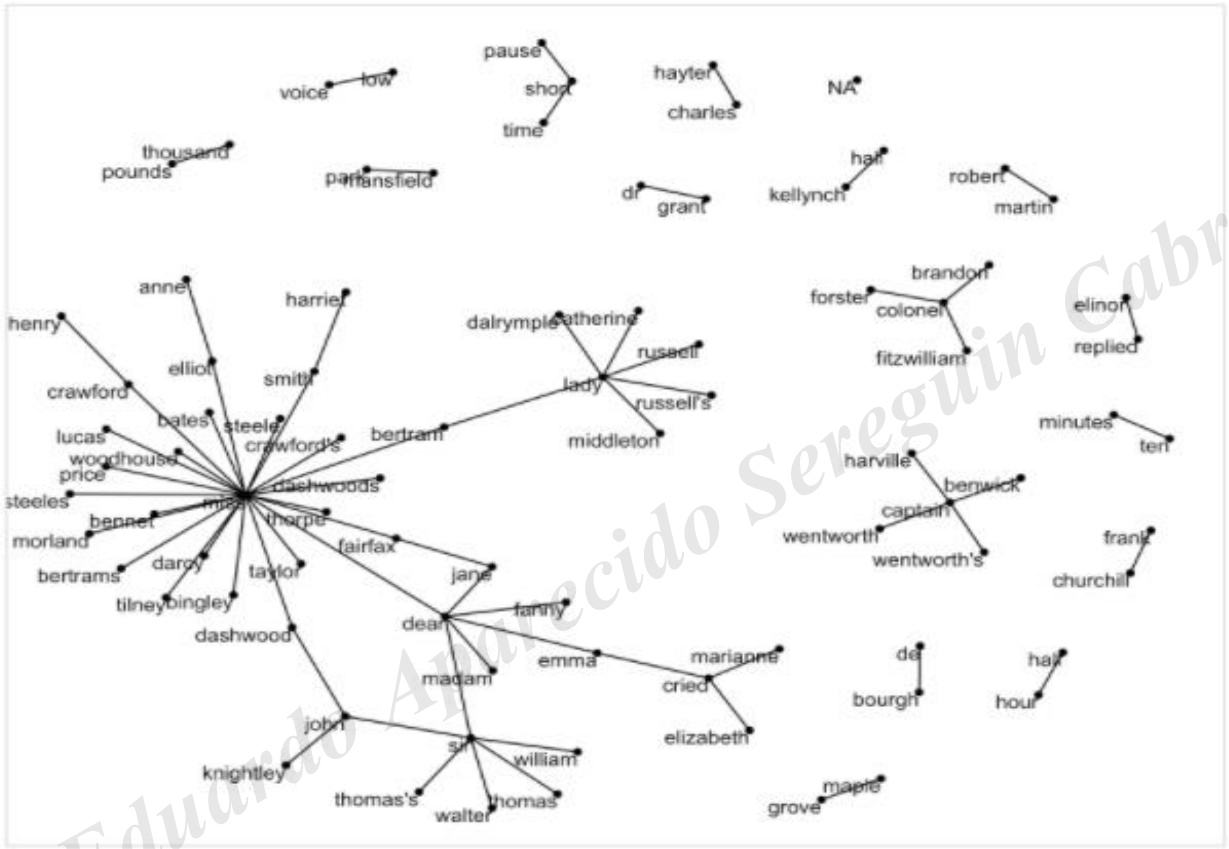
- What is bigram and n-gram?

| | | | | | | |
|---|---|---|---|---|---|---|
| *Uni-Gram* | This | Is | Big | Data | AI | Book |
| *Bi-Gram* | This is | Is Big | Big Data | Data AI | AI Book | |
| *Tri-Gram* | This is Big | Is Big Data | Big Data AI | Data AI Book | | |

https://devopedia.org/n-gram-model

# Token n-gram

- What is the importance?



# Correlation

- Is there a way of measuring if two words tend to simultaneously occur in a document?
- Tidytext is very useful for this
- Pairwise *correlation*
- *Phi* value - Pearson correlation

# Token n-gram - Correlation

| | Has word Y | No word Y | Total |
|---|---|---|---|
| Has word X | $n_{11}$ | $n_{10}$ | $n_{1.}$ |
| No word X | $n_{01}$ | $n_{00}$ | $n_{0.}$ |
| Total | $n_{.1}$ | $n_{.0}$ | n |

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1.}n_{0.}n_{.0}n_{.1}}}$$

https://www.spss-tutorials.com/pearson-correlation-coefficient/

# Topic Modelling

- Unsupervised model that allows to bring documents parts closer by similar topics.
- Practical use
- LDA method

# Final

- Practical part
- Next class

https://www.linkedin.com/in/jeronymo-marcondes-585a26186