Lecture 2

EXPLORING and PREPROCESSING Text Data

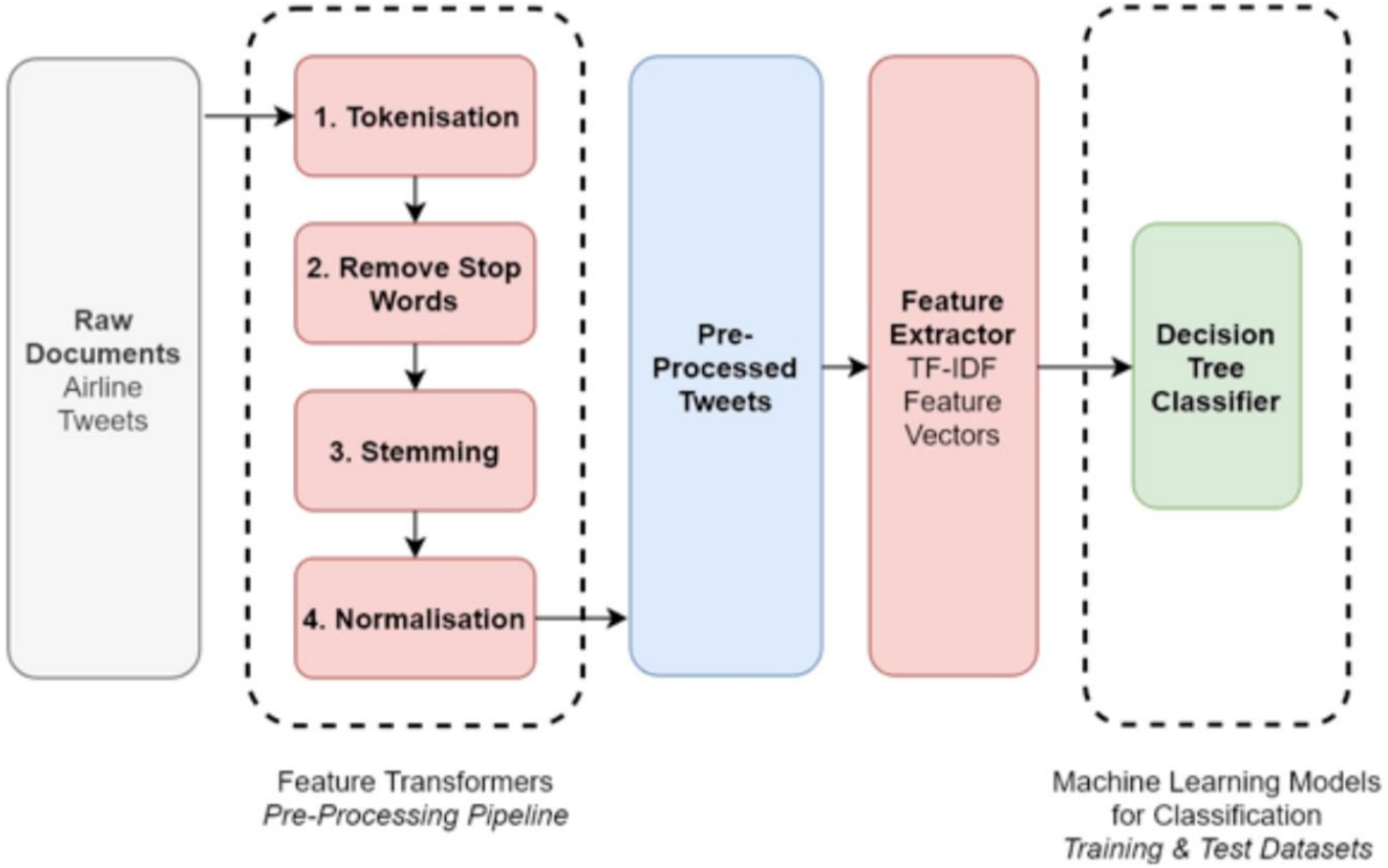
Bùi Thị Mai Anh

Trường Công nghệ Thông tin và Truyền thông, ĐHBKHN

Content

- Data Preprocessing
- Text Data Preprocessing Techniques
  - Uncapitalizing
  - Removing Punctuation
  - Removing Stopwords
  - Standardizing text
  - Correcting spelling
  - Tokenizing
  - Stemming and Lemmatization
- Text Data Exploring
- Building pre-processing model for text data

1. Data Preprocessing

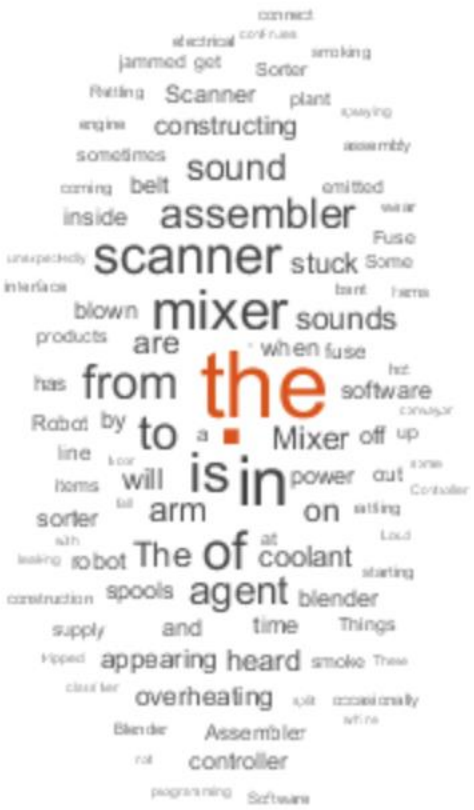


Source: https://www.mlanalytics.in/how-does-text-preprocessing-in-nlp-work/

Motivation

- Text data come from many resources (and *heterogeneous*!)
  - Web, social network, documents, etc.
  - Noise, redundancy
- ==> Need to be transformed into **<u>understandable format</u>**

Raw Data



Cleaned Data

2. Text Data Preprocessing Techniques
2.1. Uncapitalizing

- Problem: *“NLP” and “nlp” is different or not?*

- All the text needs to be represented in the same format
- Solution: using `lower()` function in `python`

Lệnh trừng phạt của Mỹ lên Huawei không chỉ tác động đến các công ty công nghệ Trung Quốc mà còn kéo theo nhiều hệ lụy tới ngành công nghiệp toàn cầu.

→

lệnh trừng phạt của mỹ lên huawei không chỉ tác động đến các công ty công nghệ trung quốc mà còn kéo theo nhiều hệ lụy tới ngành công nghiệp toàn cầu

2.2. Removing Punctuation

- *Punctuation has no meaning!*
- Reducing data dimension as well as improving the computational performance
- Solution: Regular Expression + `replace()` function in `python`

	A	B	C	D
1	Text with Punctuation			Remove Punctuation
2	"Apple"			Apple
3	(Pear). 5			Pear 5
4	{[Orange]}			Orange
5	Lemon;;; :::			Lemon
6	Lychee!			Lychee
7	<Blueberry>			Blueberry
8	Dash-test			Dashtest
9	TEST~!#$%^&*()_+{}[]"":;<>?.,			TEST

2.3. Removing Stop words

- Stop words are commonly used in text document but meaningless
- Removing stop words allows to reduce the data dimension and to improve the model performance
- Solution:
  - Using the library NLTK
  - Building a list of stop-words and removing them from the input document
  - Example: https://github.com/stopwords/vietnamese-stopwords

2.4. Standardizing Text

- Transforming acronym and abbreviation in the document text
- Solution:
  - Building a dictionary for common acronym and abbreviation in the text

Raw	Normalized
2moro 2mrrw 2morrow 2mrw tomrw	tomorrow
b4	before
otw	on the way
:) :-) ;-)	smile

2.5. Correcting Spelling

studing $\longrightarrow$ studying
intresting $\longrightarrow$ interesting
aquire $\longrightarrow$ acquire

- Data comes from social network (e.g., user comments, blogs, tweets etc.) may contain spelling errors
- Correcting spelling allows to remove redundant words
- Solution:
  - Using library TextBlod

2.6. Tokenizing

- Motivation: split the input text into "terms" ➔ numerical representation
- Input: document after some pre-processing techniques
- Output: A list of terms
- Solution
  - For English documents: NLTK, TextBlod, Spacy
  - For Vietnamese documents: VnCoreNLP, underthesea, coccoc-tokenizer

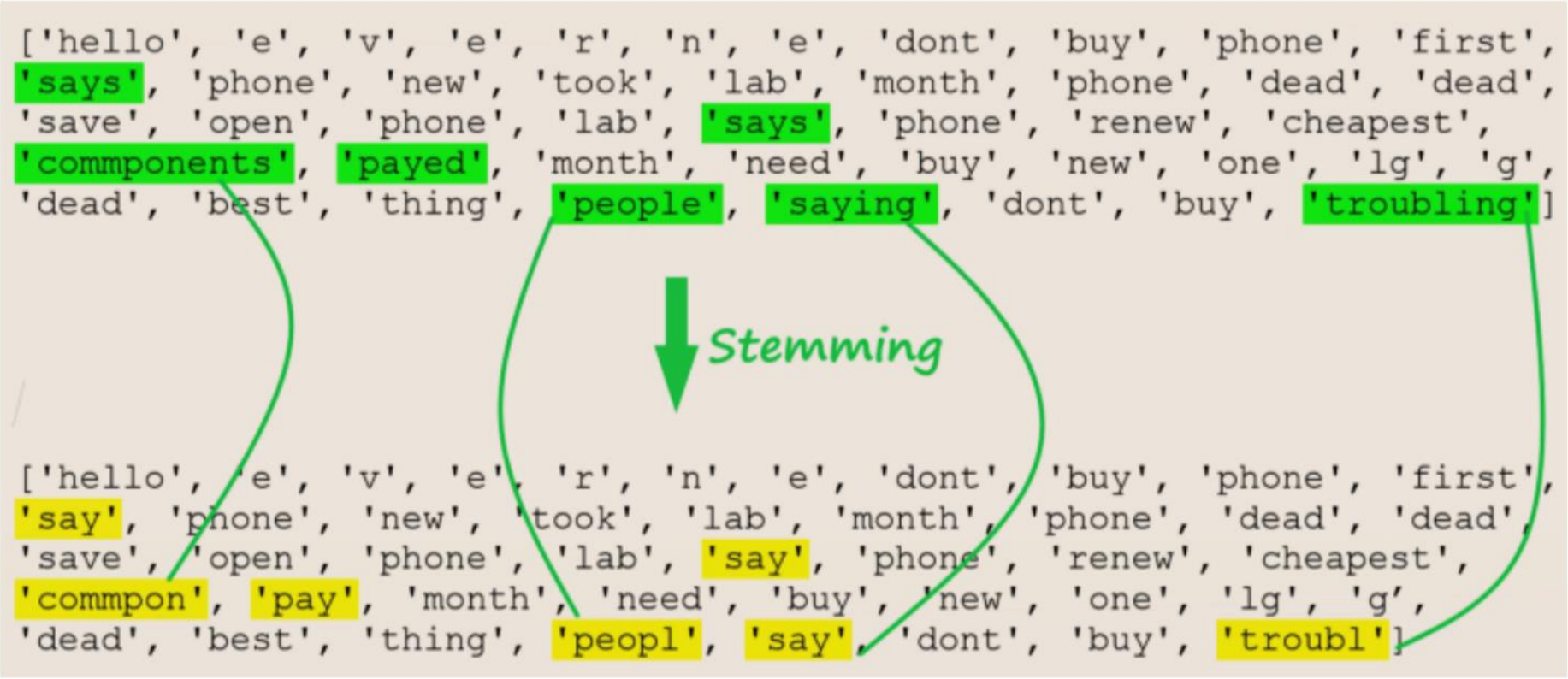
VnCoreNLP: https://github.com/vncorenlp/VnCoreNLP

Underthesea: https://github.com/undertheseanlp/underthesea

Coccoc-tokenizer: https://github.com/coccoc/coccoc-tokenizer

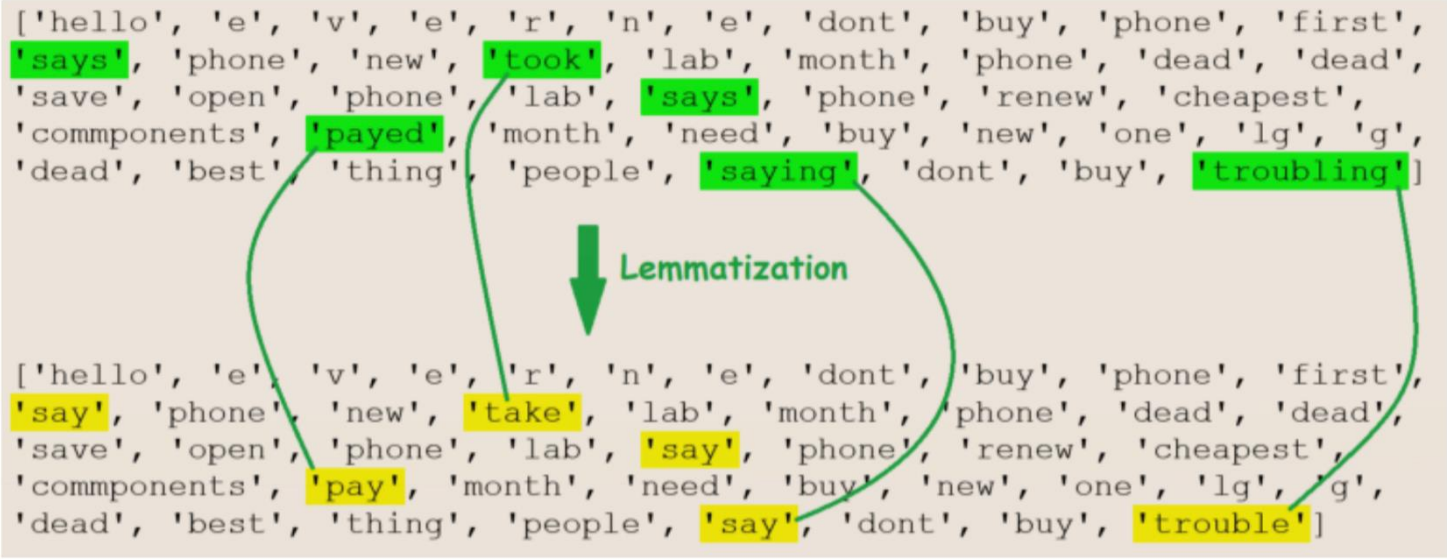
2.7. Stemming

- Extract the base form of a word by removing affixes from them
- Solution:
  - NLTK
  - TextBlod



2.8. Lemmatization

- Identify the derived forms of a word then convert them to the base form
- Input: a word
- Output: base form of this word
- Solution
  - NLTK
  - TextBlod

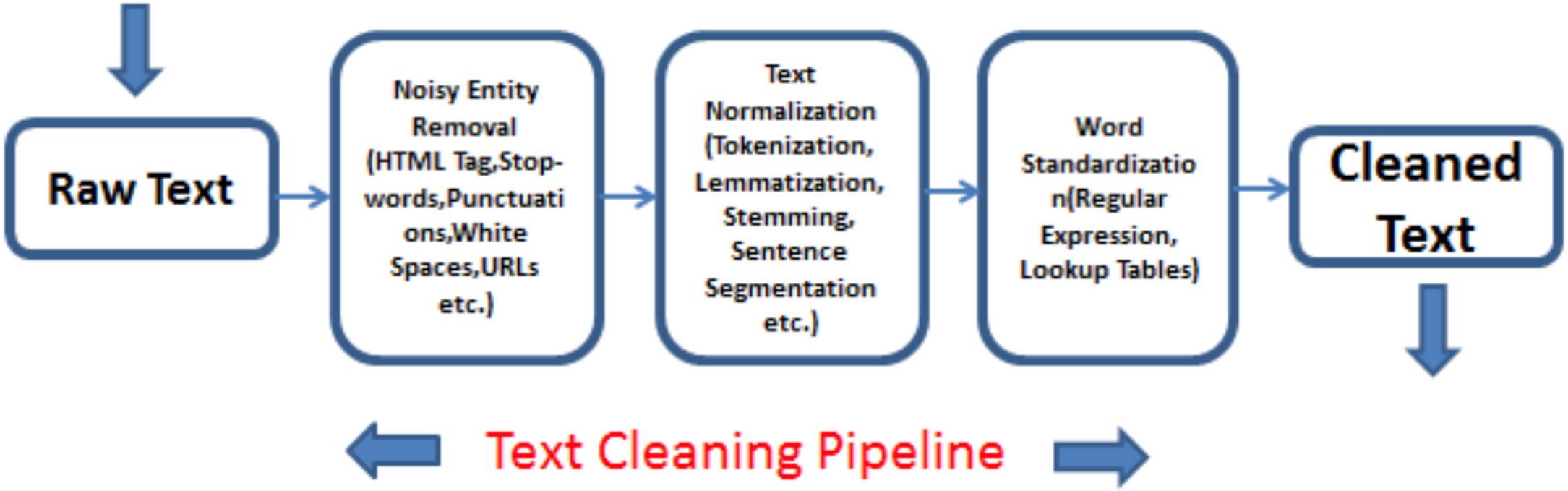


2.9. Statistical Analysis of Text Document

- Input: a text document
- Output:
  - Counting the number of words of this document
  - Counting the frequency of each word
  - Counting words with specified constraint on length
  - Building word cloud

- Solution:
  - NLTK
  - TextBlod

3. Pre-processing Text Data Model



Solution: Build a step-by-step preprocessing model with all the steps