TIỀN XỬ LÝ DỮ LIỆU VĂN BẢN (EXPLORING AND PREPROCESSING TEXT DATA)

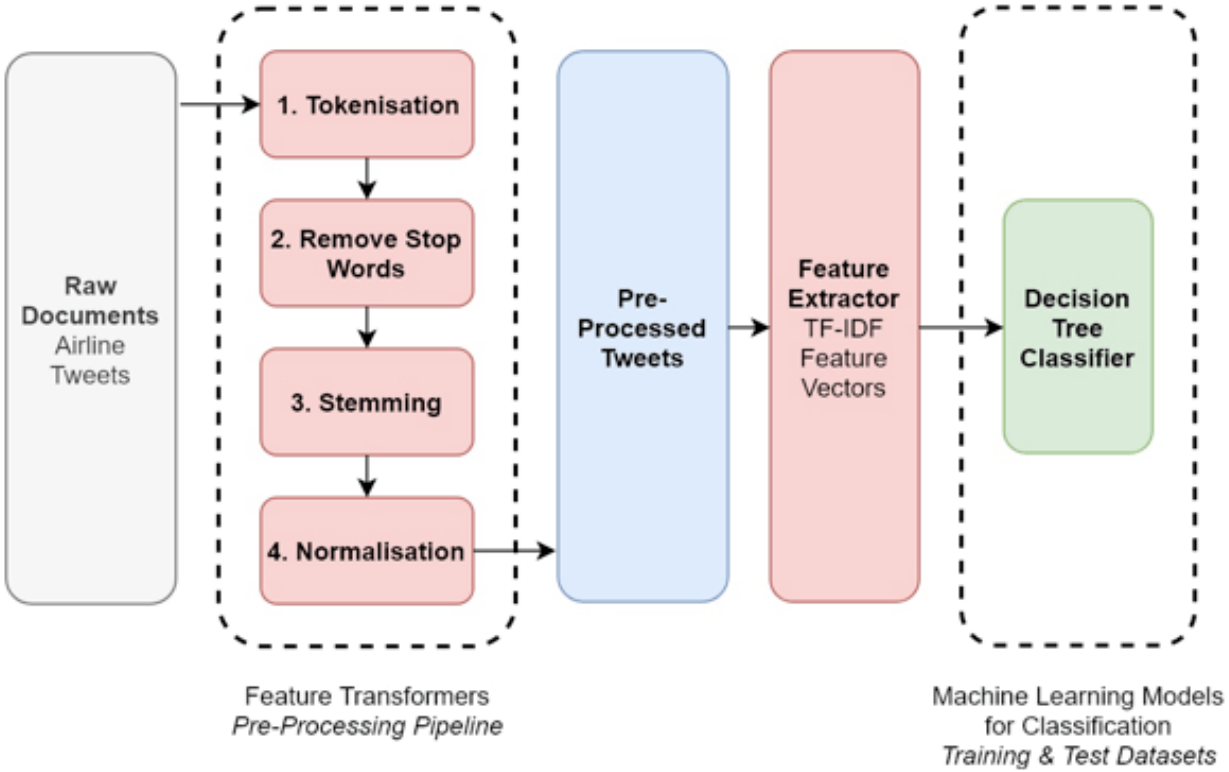
AI Academy Vietnam

Nội dung

1. Tiền xử lý dữ liệu

2. Các kỹ thuật tiền xử lý dữ liệu văn bản
   - Chuyển chữ thường
   - Loại bỏ dấu chấm (Removing Punctuation)
   - Loại bỏ từ dừng (Removing stop words)
   - Chuẩn hoá văn bản(Standardizing text)
   - Chuẩn hoá chính tả (Correcting spelling)
   - Tách từ (Tokenizing)
   - Đưa về từ gốc (stemming and lemmatization)
   - Khám phá dữ liệu văn bản
   - Xây dựng chương trình tiền xử lý

Tiền xử lý dữ liệu



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

- Tiền xử lý dữ liệu là bước quan trọng
- 70%-80% công sức là xử lý dữ liệu

Tại sao cần tiền xử lý văn bản?

- Nhiều nguồn: web, HTML, documents....
- Chứa nhiều nhiễu (noise) và chưa được làm sạch (clean)

$\longrightarrow$ **Understandable format**

Nội dung

1. Tiền xử lý dữ liệu

2. Các kỹ thuật tiền xử lý dữ liệu văn bản
   - Chuyển chữ thường
   - Loại bỏ dấu chấm (Removing Punctuation)
   - Loại bỏ từ dừng (Removing stop words)
   - Chuẩn hoá văn bản(Standardizing text)
   - Chuẩn hoá chính tả (Correcting spelling)
   - Tách từ (Tokenizing)
   - Đưa về từ gốc (stemming and lemmatization)
   - Khám phá dữ liệu văn bản
   - Xây dựng chương trình tiền xử lý

Chuyển chữ thường

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

Bài toán: tất cả dữ liệu văn bản cần được định dạng giống nhau, để chắc chắn rằng "NLP" và "nlp" là như nhau.

- **Đầu vào**: Một văn bản chứa nội dung
- **Đầu ra**: Dữ liệu văn bản đã được chuyển về chữ thường
- **Giải pháp**: Sử dụng hàm **lower()** trong python

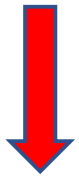
Loại bỏ dấu chấm (Removing Punctuation)

	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

- Loại bỏ punctuation:
  - Quan trọng vì punctuation không mang thêm thông tin
  - Giảm kích thước dữ liệu, đồng thời tăng hiệu quả tính toán
- Bài toán:
  - **Đầu vào**: Một văn bản chứa nội dung
  - **Đầu ra**: Văn bản đã được loại bỏ các dấu chấm
  - **Giải pháp**: Sử dụng biểu thức chính quy và hàm **replace()** trong python

Loại bỏ từ dừng (Removing stop words)

Ngay cả khi trời mưa, trận đấu vẫn diễn ra



Removing stop words

Trời mưa, trận đấu diễn ra

Từ dừng: phổ biến, nhưng không mang nhiều ý nghĩa. Việc loại bỏ giúp: giảm kích thước dữ liệu, có thể cải thiện hiệu năng của mô hình.

- **Đầu vào**: Một văn bản
- **Đầu ra**: Văn bản đã được loại bỏ các từ dừng (stop words)
- **Giải pháp**:
  - Sử dụng thư viện **NLTK**
  - Xây dựng một danh sách các từ dừng, sau đó sử dụng nó để loại bỏ các từ dừng có trong văn bản. Ví dụ stop words cho tiếng Việt [1]

[1] https://github.com/stopwords/vietnamese-stopwords

Chuẩn hoá văn bản (Standardizing text)

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

- **Đầu vào**: Một văn bản
- **Đầu ra**: Văn bản đã được chuẩn hoá
- **Giải pháp**: Tạo một từ điển để tìm kiếm các từ ngắn hoặc các từ viết tắt

Sửa lỗi chính tả (Correcting spelling)

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

- Dữ liệu có chứa lỗi chính tả (đánh gía của người dùng, blogs, tweets...)
- Giảm số bản sao của các từ. Ví dụ: nếu không sửa thì "studing" và "studying" được coi là 2 từ khác nhau.
- **Bài toán**:
  - *Đầu vào*: Một văn bản chứa nội dung
  - *Đầu ra*: Văn bản đã được sửa lỗi chính tả
  - *Giải pháp*: Sử dụng thư viện **TextBlod**
- Ví dụ minh hoạ

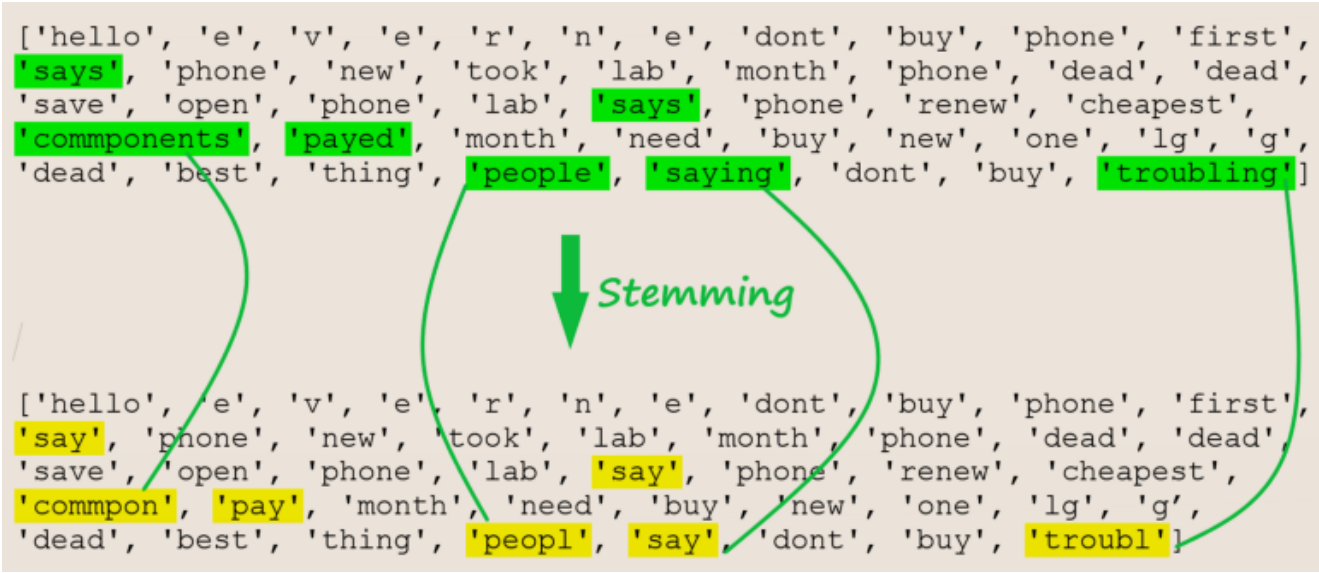
Tách từ (Tokenizing)

[captured from medium.com]

- **Đầu vào**: Một văn bản chứa nội dung
- **Đầu ra**: Các câu hoặc các từ được tách ra từ văn bản gốc
- **Giải pháp**:
  - Với tiếng Anh: Sử dụng thư viện NLTK, SpaCy, TextBlod
  - Với tiếng Việt: VnCoreNLP [2], underthesea [3], coccoc-tokenizer [4]

[2] https://github.com/vncorenlp/VnCoreNLP
[3] https://github.com/undertheseanlp/underthesea
[4] https://github.com/coccoc/coccoc-tokenizer

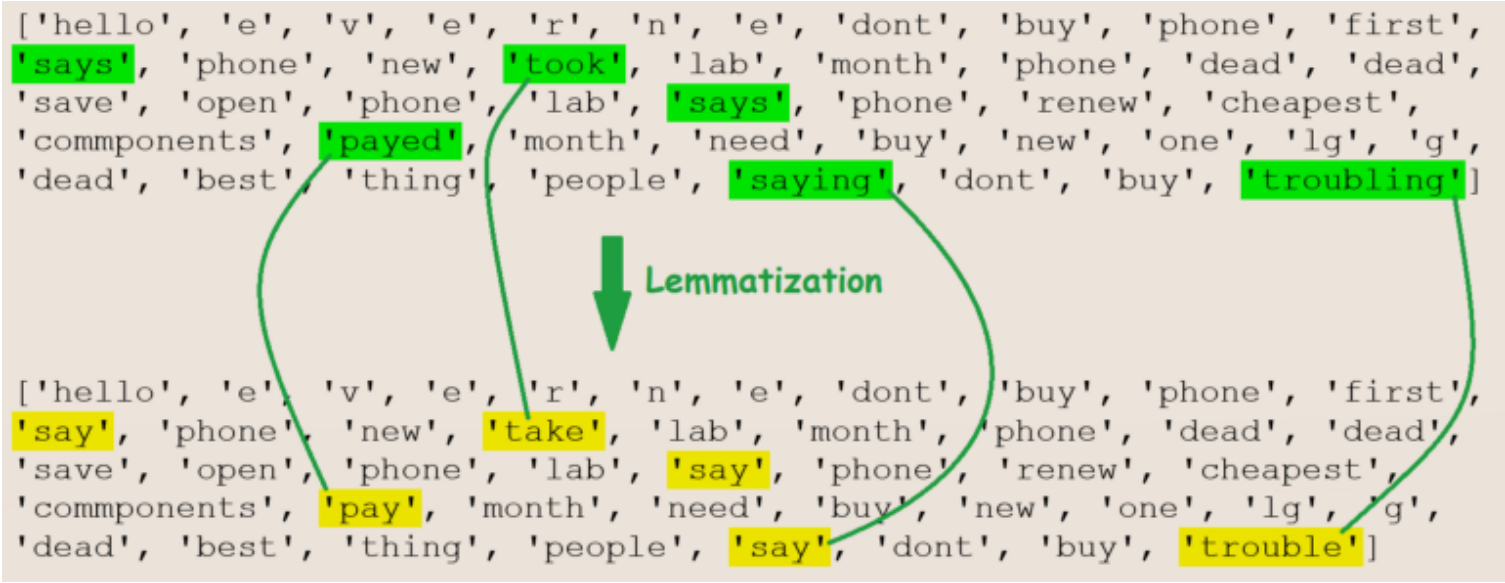
Đưa về từ gốc (stemming)



[captured from medium.com]

- **Đầu vào**: Một một từ, như fishing, fishes
- **Đầu ra**:
  - Từ gốc của từ đó, như là fish (bỏ đi các phần tiền hoặc là hậu tố)
  - Hình thái học của từ
- **Giải pháp**:
  - Sử dụng thư viện: **NLTK**
  - Sử dụng thư viện: **TextBlod**

Đưa về từ gốc (lemmatization)



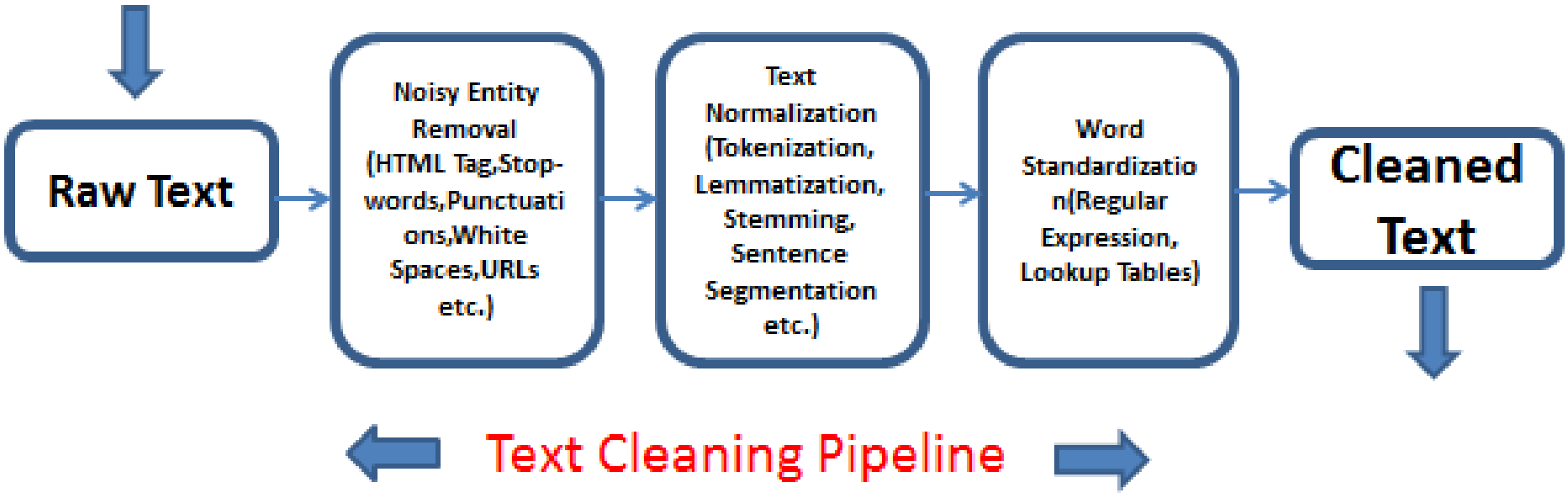
[captured from medium.com]

- **Đầu vào**: Một một từ, như good, best, better
- **Đầu ra**:
  - Từ gốc của từ đó, như là good.
  - Ngữ nghĩa của từ
- **Giải pháp**:
  - Sử dụng thư viện: **NLTK**
  - Sử dụng thư viện: **TextBlod**

Phân tích và hiểu văn bản

- **Đầu vào**: Một văn bản
- **Đầu ra**:
  - Đếm số từ của một văn bản
  - Đếm tần số xuất hiện của các từ
  - Đếm từ với độ dài lớn hơn 3 và vẽ phân bố
  - Xây dựng đám mây từ (word cloud)
- **Giải pháp**:
  - Sử dụng thư viện: **NLTK**
  - Sử dụng thư viện: **TextBlod**

Xây dựng một chương trình tiền xử lý dữ liệu văn bản



[captured from medium.com]

- **Đầu vào**: Một văn bản
- **Đầu ra**: Văn bản đã được tiền xử lý
- **Giải pháp**: Tạo một hàm tiền xử lý văn bản với tất cả các kỹ thuật tiền xử lý đã được trình bày ở trên.

THANK YOU

Q&A