Lecture 2

EXPLORING and PREPROCESSING Text Data

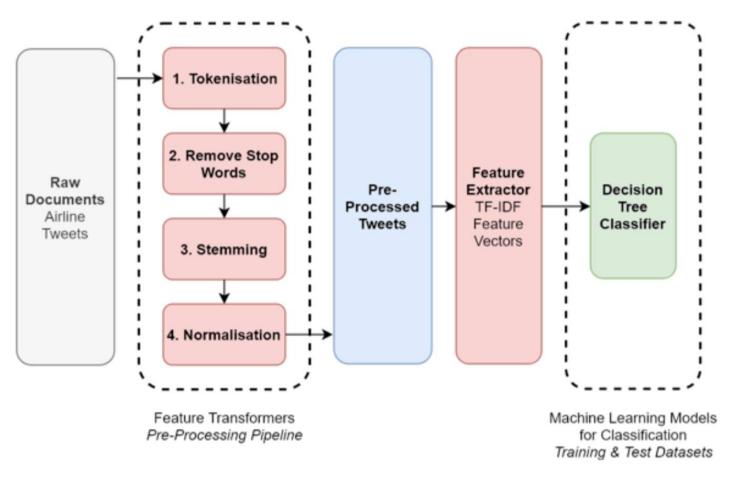
Bùi Thị Mai Anh

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 understandable format

Raw Data



Cleaned Data

```
laud Spool arm. thing
stuck robot product
   blender bend
```

Text Data Preprocessing Techniques 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

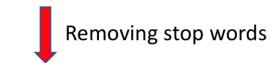
- Punctuation has no meaning!
- Reducing data dimension as well as improving the computational performance
- Solution: Regular Expression
 + replace() function in
 python

4	А	В	С	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>			Blueberry
8	Dash-test			Dashtest
9	TEST~!#\$%^&*()_+{}[]"":;<>?.,			TEST
40				

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

Ngay cả khi trời mưa, trận đấu vẫn diễn ra



Trời mưa, trận đấu diễn ra

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

```
\operatorname{studing} \longrightarrow \operatorname{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 preprocessing 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

'hello e v e r y o n e dont buy this phone at all first o f all that says the phone in new i took it to the lab after month the phone is dead dead you can save it they open the phone in the lab and say s the phone is renew and its cheapest commponents i payed for on ly month now i need to buy new one this lg g is dead not a best thing people are saying to me dont buy from at all it s troubling '



['hello', 'e', 'v', 'e', 'r', 'y', 'o', 'n', 'e', 'dont', 'buy', 'this', 'phone', 'at', 'all', 'first', 'of', 'all', 'that', 'says', 'the', 'phone', 'in', 'new', 'i', 'took', 'it', 'to', 'the', 'lab', 'after', 'month', 'the', 'phone', 'is', 'dead', 'dead, 'you', 'can', 'save', 'it', 'they', 'open', 'the', 'phone', 'in', 'the', 'lab', 'and', 'says', 'the', 'phone', 'is', 'renew', 'and', 'its', 'cheapest', 'commponents', 'i', 'payed', 'for', 'only', 'month', 'now', 'i', 'need', 'to', 'buy', 'new', 'one', 'this', 'lg', 'g', 'is', 'dead', 'not', 'a', 'best', 'thing', 'people', 'are', 'saying', 'to', 'me', 'dont', 'buy', 'from', 'at', 'all', 'it', 's', 'troubling']

2.7. Stemming

 Extract the base form of a word by removing affixes

from them

- Solution:
 - NLTK
 - TextBlod

```
['hello', 'e', 'v', 'e', 'r', 'n', 'e', 'dont', 'buy', 'phone', 'first',
'says', 'phone', 'new', 'took', 'lab', 'month', 'phone', 'dead', 'dead',
'save', 'open', 'phone', 'lab', 'says', 'phone', 'renew', 'cheapest',
'commponents', 'payed', 'month', 'need', 'buy', 'new', 'one', 'lg', 'g',
'dead', 'best', 'thing', 'people', 'saying', 'dont', 'buy', 'troubling']
['hello', /e', 'v', 'e', 'r', 'n', 'e', 'dont', 'buy', 'phone', 'first'
'say', 'phone', 'new', 'took', 'lab', 'month', 'phone', 'dead', 'dead'
'save', /'open', 'phone', \'lab', 'say', 'phone', 'renew', 'cheapest',
'commpon', 'pay', 'month', 'need', 'buy', 'new', 'one', 'lg', 'g',
'dead', 'best', 'thing', 'peopl', 'say', 'dont', 'buy', 'troubl'
```

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

```
['hello', 'e', 'v', 'e', 'r', 'n', 'e', 'dont', 'buy', 'phone', 'first', 'says', 'phone', 'new', 'lab', 'month', 'phone', 'dead', 'dead', 'save', 'open', 'phone', 'lab', 'says', 'phone', 'renew', 'cheapest', 'month', 'need', 'buy', 'new', 'one', 'lg', 'g', 'dead', 'best', 'thing', 'people', 'saying', 'dont', 'buy', 'troubling']

Lemmatization

['hello', 'e', 'v', 'e', 'r', 'n', 'e', 'dont', 'buy', 'phone', 'first', 'say', 'phone', 'new', 'take', 'lab', 'month', 'phone', 'dead', 'dead', 'save', 'open', 'phone', 'lab', 'say', 'phone', 'renew', 'cheapest', 'commponents', 'pay', 'month', 'need', 'buy', 'new', 'one', 'lg', 'g', 'dead', 'best', 'thing', 'people', 'say', 'dont', 'buy', 'trouble']
```

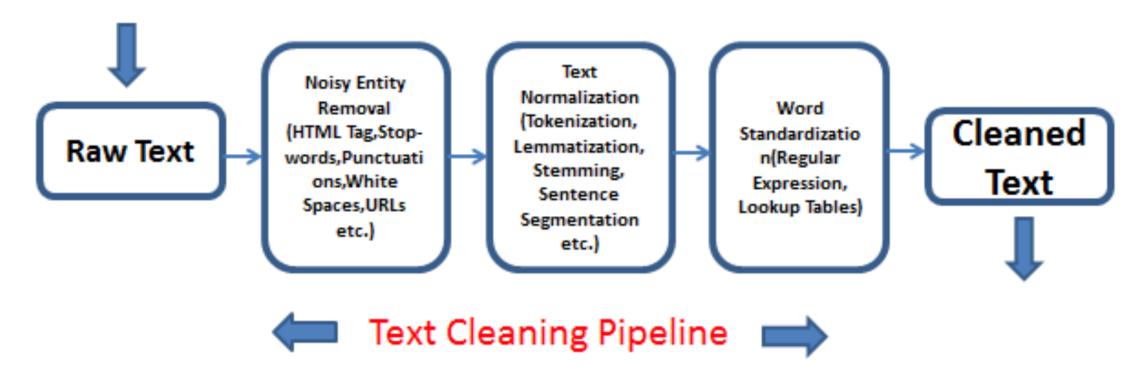
2.9. Statistical Analysis of Text Document

- 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