

Section 1:

1. <https://www.nltk.org/data.html>

NLTK 3.4.5 documentation

PREVIOUS | NEXT | MODULES | INDEX

Installing NLTK Data

NLTK comes with many corpora, toy grammars, trained models, etc. A complete list is posted at: http://nltk.org/nltk_data/

To install the data, first install NLTK (see <http://nltk.org/install.html>), then use NLTK’s data downloader as described below.

Apart from individual data packages, you can download the entire collection (using “all”), or just the data required for the examples and exercises in the book (using “book”), or just the corpora and no grammars or trained models (using “all-corpora”).

Interactive installer

For central installation on a multi-user machine, do the following from an administrator account.

Run the Python interpreter and type the commands:

```
>>> import nltk
>>> nltk.download()
```

2. Interactive installer

```
In [3]: import nltk

In [*]: nltk.download()

showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

NLTK Downloader

File View Sort Help

CollectionsCorporaModelsAll Packages

Identifier	Name	Size	Status
all	All packages	n/a	not installed
all-corpora	All the corpora	n/a	not installed
all-nltk	All packages available on nltk_data gh-pages branch	n/a	not installed
book	Everything used in the NLTK Book	n/a	not installed
popular	Popular packages	n/a	not installed
tests	Packages for running tests	n/a	not installed
third-party	Third-party data packages	n/a	not installed

Download

Refresh

Server Index: https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

Download Directory: C:\Users\platon\AppData\Roaming\nltk_data

- Sentence Segmentation (*sent_tokenize*)
- Word Segmentation (*word_tokenize*)

- POS Tagging (*pos_tag*)
- Chunking (*chunk*)
- Dependency Parsing (*dependency_parse*)
- Named Entity Recognition (*ner*)
- Text Classification (*classify*)

5. Sử dụng thư viện Pyvi để tách từ tiếng Việt

from pyvi import ViTokenizer, ViPosTagger

```
[18] def clean_document(doc):
    doc = ViTokenizer.tokenize(doc) #Pyvi Vitokenizer library
    doc = doc.lower() #Lower
    tokens = doc.split() #Split in_to words
    tokens = [word for word in tokens if word]
    return tokens
clean_document('Bác sĩ bây giờ có thể thản nhiên báo tin bệnh nhân bị ung thư?')
```

```
['bác_sĩ',
 'bây_giờ',
 'có_thể',
 'thản_nhiên',
 'báo',
 'tin',
 'bệnh_nhân',
 'bị',
 'ung_thư',
 '?']
```

Section 2:

import một số thư viện, nếu chưa có thì setting thư viện

```
from pyvi import ViTokenizer, ViPosTagger
import gensim
import os
import pickle
import codecs
import re
from tqdm import tqdm
import regex
import pandas as pd
```

6. Xóa các thẻ HTML

```
def remove_html(txt):
    return regex.sub(r'<[^>]*>', '', txt)
```

7. Xóa stopwords

```

)## ----- Xóa stop words -----
)## -----
stopword = set()
with open('stopwords.txt', 'r', encoding='utf-8') as _fp:
    word = _fp.readlines()
    stopword = [n.replace('\n', '') for n in word]

def remove_stopwords(line):
    words = []
    for word in line.strip().split():
        if word not in stopword:
            words.append(word)
    return ' '.join(words)

```

8. Tiền xử lý dữ liệu văn bản.

[illegible]

