Assignment 3: Tiền xử lý văn bản với tiếng Việt

Tổng quan: ở bài tập này chúng ta xây dựng chương trình tiền xử lý văn bản với tiếng Việt.

Import các thư viện cần dùng. Lưu ý ở đây ta dùng thư viện underthesea để dùng tokenize tiếng Việt, để cài đặt các bạn làm theo hướng dẫn sau (link)

```
import os,glob
import codecs
import sys
import re
from underthesea import word_tokenize
```

Câu hỏi 1: Tạo corpus và khảo sát dữ liệu

Dữ liệu trong phần này được trích một phần từ bộ dữ liệu VNTC. VNTC là bộ dữ liệu về tin tức tiếng Việt với nhiều chủ đề khác nhau. Trong nội dung phần này ta chỉ xử lý cho chủ đề khoa học trong VNTC. Ta sẽ tạo một corpus từ cả thư mục train và test. Hoàn thiện đoạn chương trình:

- Ghi sentences_list ra 1 file với tên dataset_name.txt, mỗi phần tử là một doc, sẽ trên 1 dòng.
- Kiểm tra xem trong corpus có bao nhiêu docs

```
dataset_name = "VNTC_khoahoc"
path = ['./VNTC_khoahoc/Train_Full/', './VNTC_khoahoc/Test_Full/']

if os.listdir(path[0]) == os.listdir(path[1]):
    folder_list = [os.listdir(path[0]), os.listdir(path[1])]
    print("train labels = test labels")
else:
    print("train labels differ from test labels")

doc_num = 0
sentences_list = []
meta_data_list = []
for i in range(2):
    for folder_name in folder_list[i]:
        folder_path = path[i] + folder_name
        if folder_name[0] != ".":
            for file_name in glob.glob(os.path.join(folder_path, '*.txt')):
                # Đọc nội dung file vào f
                f = codecs.open(file_name, 'br')
                # Chuyển sang định dạng utf-16 cho dữ liệu tiếng Việt
                file_content = (f.read().decode("utf-16")).replace("\r\n", " ")
                sentences_list.append(file_content.strip())
                f.close
                # Đếm số lượng docs
                doc_num += 1
```

```
#### YOUR CODE HERE ####
# Đọc nội dung của tên dataset_name.txt vào f
f = open(dataset_name + '.txt', 'w')
sentences_str = '\n'.join(sentences_list)
# Ghi ra file
f.write(sentences_str)
f.close

print("Doc number: ", doc_num)
#### END YOUR CODE #####
```

```
train labels = test labels
Doc number:  3916
```

Câu hỏi 2: Viết các hàm tiền xử lý

Câu hỏi 2.1: Viết hàm làm sạch văn bản.

Gợi ý:

- Biết rằng văn bản chỉ giữ lại các ký tự sau: aAàÀảẢãÃáÁạẠăĂằẰẳẲẵẴắẮặẶâÂầẦẩẨẫẪấẤậẬbBcCdDđĐeEèÈẻẺẽẼéÉẹẸêÊềỀểỂễỄếẾệỆfFgGhHiIìÌỉỈĩĨíÍịỊjJkKlLmMnNoOòÒỏỎõÕóÓọỌôÔồỒổỔỗỖốỐộỘơƠờỜởỞỡỠớỚợỢpPqQrRsStTuUùÙủỦũŨúÚụỤưƯừỪửỬữỮứỨựỰvVwWxXyYỳỲỷỶỹỸýÝỵỴzZ0-9(),!?'`
- Sau đó cắt các khoảng trắng của văn bản đầu vào.

```
def clean_str(string):
    #### YOUR CODE HERE ####
    # Làm sạch văn bản chỉ giữ lại các ký tự như đã định
    string = re.sub(r"[^aAàÀảẢãÃáÁạẠăĂằẰẳẲẵẴắẮặẶâÂầẦẩẨẫẪấẤậẬbBcCdDđĐeEèÈẻẺẽẼéÉẹẸêÊềỀểỂễỄếẾệỆfFgGhHiIìÌỉ
    return string.strip() # Cắt các khoảng trắng

    #### END YOUR CODE #####
```

Câu hỏi 2.2: Viết hàm chuyển văn bản về dạng thường

```
# make all text lowercase
def text_lowercase(string):
#### YOUR CODE HERE ####

    return string.lower()

#### END YOUR CODE #####
```

Câu hỏi 2.3: Tách từ

Gợi ý: sử dụng hàm word_tokenize() đã import ở trên với hai tham số lần lượt là strings, và format="text".

```
def tokenize(strings):
    #### YOUR CODE HERE ####

    return word_tokenize(strings, format="text")

    #### END YOUR CODE #####
```

Câu hỏi 2.4 Loại bỏ từ dừng

Ở đây để loại bỏ từ dừng, ta dùng một danh sách các từ dừng tiếng Việt. Chúng được lưu trong file './vietnamese-stopwords.txt'. Hoàn thiện chương trình sau:

- Đối chiếu từng từ trong văn bản (strings) nếu từ nào không có trong từ dừng thì thêm vào doc_words

```
def remove_stopwords(strings):
    #### YOUR CODE HERE ####
    strings = strings.split()
    f = open('./vietnamese-stopwords.txt', 'r')
    stopwords = f.readlines()
    stop_words = [s.replace("\n", '') for s in stopwords]
    doc_words = []

    for word in strings:
        if word not in stop_words:
            doc_words.append(word)

    doc_str = ' '.join(doc_words).strip()
    return doc_str
    #### END YOUR CODE #####
```

Câu hỏi 2.5: Xây dựng hàm tiền xử lý

Gợi ý: Gọi lần lượt các hàm clean_str, text_lowercase, tokenize, remove_stopwords, rồi trả ra kết quả cho hàm.

```
def text_preprocessing(strings):
    #### YOUR CODE HERE ####

    temp = clean_str(strings)
    temp = text_lowercase(temp)
    temp = tokenize(temp)
    temp = remove_stopwords(temp)
    return temp

    #### END YOUR CODE #####
```

Câu hỏi 3: Thực hiện tiền xử lý

Công việc giờ chúng ta sẽ đọc corpus từ file đã làm ở câu hỏi 1. Sau đó ta sẽ gọi hàm tiền xử lý cho từng doc trong corpus trên.

Gợi ý: Gọi hàm text_preprocessing() với tham số đầu vào là doc_content, lưu vào biến temp1

```
#### YOUR CODE HERE ####
doc_content_list = []

# Đọc corpus từ file rồi ghi vào doc_content_list
f = open(dataset_name + '.txt', 'br')
for line in f.readlines():
    doc_content_list.append(line.strip().decode('utf-8'))
f.close()
print("\nlength of docs = ", len(doc_content_list))
print('docs[0]:\n' + doc_content_list[0])

clean_docs = []
# Gọi hàm tiền xử lý cho từng doc
for doc_content in doc_content_list:
    temp1 = text_preprocessing(doc_content)
    clean_docs.append(temp1)
#### END YOUR CODE #####
print("\nlength of clean_docs = ", len(clean_docs))
print('clean_docs[0]:\n' + clean_docs[0])
```

```
length of docs =  3916
docs[0]:
Giải thưởng sách khoa học uy tín 2004 đã có chủ Với ấn phẩm A Short History Of Nearly Everything,

length of clean_docs =  3916
clean_docs[0]:
giải_thưởng sách khoa_học uy_tín 2004 chủ ấn_phẩm a short history of nearly everything , nhà_văn
```

Câu 4: Ghi dữ liệu đã tiền xử lý

Gợi ý: Ghi dữ liệu đã tiền xử lý vào file có tên dataset_name + '.clean.txt', trong đó mỗi doc ghi trên một dòng.

```
#### YOUR CODE HERE ####
clean_corpus_str = '\n'.join(clean_docs)
f = open(dataset_name + '.clean.txt', 'w')
f.write(clean_corpus_str)
f.close()
#### YOUR CODE HERE ####
```