DATA 641 - Lab 2

Homayoon Fotros

Exercise 1

Translation / Back-translation

[1]:
```python
from BackTranslation import BackTranslation as BkTrans

trans = BkTrans()

## To Spanish
result = trans.translate('coffee', src='en', tmp='es')
result.tran_text
```

[1]:
```
'café'
```

[4]:
```python
## To French
result2 = trans.translate('coffee', src='en', tmp='fr')
result2.tran_text
```

[4]:
```
'café'
```

[5]:
```python
## To German
result3 = trans.translate('coffee', src='en', tmp='de')
result3.tran_text
```

[5]:
```
'Kaffee'
```

Exercise 2

HTML Parsing and Cleanup

[6]:
```python
from bs4 import BeautifulSoup
from urllib.request import urlopen
```

[9]:
```python
## Part (a) - Parsing the question on Stack Overflow
myurl = "https://stackoverflow.com/questions/415511/
 ↪how-to-get-the-current-time-in-python"

html = urlopen(myurl).read()
```

```
soupified = BeautifulSoup(html, 'html.parser')

question = soupified.find("div", {"class": "question"})

questiontext = question.find("div", {"class": "s-prose js-post-body"})

print('Page Title:' , soupified.title, '\n')
print("Question: \n", questiontext.get_text().strip())
```

```
Page Title: <title>datetime - How to get the current time in Python - Stack
Overflow</title>

Question:
 What is the module/method used to get the current time?
```

[10]:

```
## Part (b)  - Downloading COVID data set
import urllib.request
```

[19]:

```
zois_url = "https://zoisboukouvalas.github.io/Code.html"
zois_html = urlopen(zois_url).read()
zois_soup = BeautifulSoup(zois_html, 'html.parser')

dt_cls = zois_soup.find_all("a") ## parsing all <a> placeholders

lnk = []
for link in dt_cls: ## parsing all the links
    if(link.get('href')!=None):
        lnk.append(link.get('href'))

urls_zois = [item for item in lnk if item.find('http')==0] ## filtering out␣
 ↪links not starting with http

xls_zois = [item for item in urls_zois if item.find('xls')!=-1] ## locating the␣
 ↪download link with .xls suffix

urllib.request.urlretrieve(xls_zois[0], "zois_dataset.xlsx") ## Downloading the␣
 ↪file
```

[19]:

```
('zois_dataset.xlsx', <http.client.HTTPMessage at 0x24308df8460>)
```

Exercise 3

Extracting Text from PDF Files

Difficulties of Extracting from PDF files

- Read/Copy Protection
- Characters and Text out of the page (Off-page Text)

- Invisible or hardly-visible Text
- Kerning-related issues (Extra spaces within and between words)
- Spaces removed after extraction
- Embedded fonts (subfonts and code maps)
- Word and Paragraph detection (also ordering)
- Detecting images and other layers

PDF Extraction / Working On the Corpus

[2]:
```
from PyPDF2 import PdfFileReader
```

[3]:
```
file_var = open('sample.pdf','rb')

my_file = PdfFileReader(file_var)
pg1 = my_file.getPage(0)
txt = pg1.extractText()

print(txt)
```

```
 A Simple PDF File  This is a small demonstration .pdf file -  just for use in
the Virtual Mechanics tutorials. More text. And more  text. And more text. And
more text. And more text.  And more text. And more text. And more text. And more
text. And more  text. And more text. Boring, zzzzz. And more text. And more
text. And  more text. And more text. And more text. And more text. And more
text.  And more text. And more text.  And more text. And more text. And more
text. And more text. And more  text. And more text. And more text. Even more.
Continued on page 2 ...
```

[4]:
```
file_var.close()
```

Exercise 4

Text Pre-processing

[11]:
```
import re
import string
```

[12]:
```
corpus = "Need to finalize the demo corpus which will be used for this notebook␣
 ↪& should be done soon !!. It should be done by the ending of this month. But␣
 ↪will it? This notebook has been run 4 times !!"
```

[13]:
```
## Lower-case corpus
corpus_lower = corpus.lower()
corpus_lower
```

[13]:

```
'need to finalize the demo corpus which will be used for this notebook & should
be done soon !!. it should be done by the ending of this month. but will it?
this notebook has been run 4 times !!'
```

[15]:

```
## Removing digits, punctuations, and trimming white-spaces
corpus_no_dig = re.sub('\d+[ ]', '', corpus_lower) ## remove digits
corpus_rem_punc = (' '.join(word.strip(string.punctuation) for word in␣
 ↪corpus_no_dig.split())) ## remove punctuations
corpus_clean = re.sub('\s+', ' ',corpus_rem_punc).strip() ## remove extra␣
 ↪white-spaces

corpus_clean
```

[15]:

```
'need to finalize the demo corpus which will be used for this notebook should be
done soon it should be done by the ending of this month but will it this
notebook has been run times'
```

[16]:

```
## Tokenizing Corpus and Removing Stopwords
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

corpus_token = word_tokenize(corpus_clean)

stopw = stopwords.words('english')
corp_no_stop = ' '.join(word for word in corpus_token if word not in stopw)

corp_no_stop
```

[16]:

```
'need finalize demo corpus used notebook done soon done ending month notebook
run times'
```

[20]:

```
## Stemming
from nltk.stem import PorterStemmer
from nltk.stem import LancasterStemmer

ps = PorterStemmer() ## Porter Stemmer

corpus_stemmed = [ps.stem(word) for word in word_tokenize(corp_no_stop)]
' '.join(corpus_stemmed)
```

[20]:

```
'need final demo corpu use notebook done soon done end month notebook run time'
```

[21]:

```
## Lemmentizing
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
```

```
words = [lemmatizer.lemmatize(word,pos='a') for word in␣
 ↪word_tokenize(corp_no_stop)]
' '.join(words)
```

```
[21]: 'need finalize demo corpus used notebook done soon done ending month notebook
      run times'
```