Exercise Lab

TDT4310 - Intelligent Text Analytics and Language Understanding

Spring 2020

Duong Quang Huy

Email: quang.huy.duong@ntnu.no

Room: 202 IT-bygget

Norwegian University of Science and Technology

Communication

Email: quang.huy.duong@ntnu.no

Room: 202 IT-bygget

Blackboard: https://ntnu.blackboard.com

Piazza: for public questions/discussions, https://piazza.com/ntnu.no/spring2020/tdt4310/home

Norwegian University of Science and Technology

Outline of Exercise Labs

6/7 labs

- Lab 1 (14-Jan-2020): Corpora
- Lab 2 (28-Jan-2020): Tagging
- Lab 3 (11-Feb-2020): Classification
- Lab 4 (25-Feb-2020): Grammar & Parsing
- Lab 5 (10-Mar-2020): Information Extraction
- Lab 6 (31-Mar-2020): Chat bots & Deep learning

[Option] Lab 7: Reviews & Extra Exercises

- Improve a chat bot: remember information, train on multiple datasets
- Trends, topics at NIPS, ICLR, ICML (2020)
- Reading some papers, implement algorithms, discuss advantages & disadvantages

Requirements

Textbook:

1. Applied Text Analysis with Python
    - by Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda
    - the code repository of the book is at https://github.com/foxbook/
2. Natural Language Processing with Python
    - by Steven Bird, Ewan Klein & Edward Loper
    - nltk.org

Requirements

- Python 3+ (Current ver 3.8.1) Download at http://www.python.org/downloads/
- Attendance is not mandatory
- But you must submit your solution for your assignments

Requirements

- Source code of assignments must be in Python.
- In each lab, we have exercises and your task is to code and submit your solution.
- Deadline for submission is in the assignment (officially, deadline of each lab is the day before the next lab)

How?

- I will upload the assignment & slides to blackboard 1 day before or on the day of the lab
- You can do the assignments alone or in a group up to 2 persons
- Again: no failure (of course, only in our lab) if you do something

How?

- Solution is uploaded to blackboard before the deadline
- Solution = code + report
- Code: is executeable (if have arguments, need intruction of how to run

How?

- If done by a group, need to report about task of each member
- You can use any tools you want for writing your code: e.g anaconda, jupyter, pycharm, pure python IDLE.
- No failure: if you already do and submit your code, I will try to assest it and show where we could do better. No wrong code :d

Structure of a Lab

- In 2 hours
- The first part (period) presents the content of current lab
- Go through the solutions for exercises of the assignment
- Q&A

Reason

Natural Language is *unstructured data.* spoke by human. Human can understand but machine can not.

Need to process and analyze Natural Language to *structured data that machine can understand.*

Why python: a language that has rich scientific, powerful and numeric computing supported libraries.

Some important packages

- *Scikit-Learn: https://scikit-learn.org*
- *NLTK: http://www.nltk.org/*
- *Gensim: https://pypi.org/project/gensim/*
- *NumPy: http://www.numpy.org/*
- *Pandas: https://pandas.pydata.org*
- *Keras: https://keras.io/*

Basic

How to install package in python: use pip with command line

```
Pip install/uninstall -U package_name
```

How to use package in your code: use "import" keyword

```
>>>import package
```

Corpus

Management:

- Database Management
- Disk structure

Accessing the corpus: use *CorpusReader*

Corpus

Some custom-specific purpose readers:

- PlaintextCorpusReader
- TaggedCorpusReader
- BracketParseCorpusReader
- ChunkedCorpusReader

Corpus (cont.)

Some custom-specific purpose readers:

- ChunkedCorpusReader
- TwitterCorpusReader
- WordListCorpusReader
- XMLCorpusReader
- CategorizedCorpusReader
- Or we can write our own custom reader

Accessing Text Corpora and Lexical Resources

Basic corpus functionality:

- **fileids**(options): Get files of the corpus.
- **categories**(options): Get categories of the corpus.
- **raw**(options=[o1,o2]): Get raw content of the corpus.
- **words**(options=[o1,o2]): Get words of the corpus.
- **sents**(options=[o1,o2]): Get sentences of the specified options.

Corpora processing

Breaking Down Documents:

- Paragraphs
- Sentences
- Words

Tokenizers: breaks a stream of characters into individual tokens.

Corpora processing

Different tokenizers:

- TreebankWordTokenizer
- WordPunctTokenize
- PunktWordTokenizer
- RegexpTokenizer

Exercise Corpora

Use *gutenberg* in *nltk.corpus*

```
>>> from nltk.corpus import gutenberg as gb
```

Your tasks:

- List all document in the corpus.
- Count number of characters of the first document.
- Count number of words of the first document
- Count number of sentences of the first document.
- Display the words from position 11th to 40th of the first document.

Exercise Corpora

- List all documents in the corpus

```
>>> gb.fileids()
```

- Count number of characters of the first document

```
>>> len(gb.raw('austen-emma.txt'))
```

- Count number of words of the first document

```
>>> len(gb.words('austen-emma.txt'))
```

- Count number of sentences of the first document

```
>>> len(gb. sents('austen-emma.txt'))
```

- Display the words from position 11th to 40th of the first document

```
>>> ' '.join(gb.words('austen-emma.txt')[11:40])
```

Exercise Genre and Frequencies

Frequency distributions:

- use *Brown* Corpus by using *import brown.*
- compare genres in *Brown* Corpus

'news', 'religion', 'hobbies', 'science_fiction', 'romance'

- with modal words

'can', 'could', 'may', 'might', 'must', 'will'

Exercise Genre and Frequencies

```
import nltk
from  nltk.corpus import brown
cfd = nltk.ConditionalFreqDist((genre, word)
        for genre in brown.categories()
                for word in brown.words(categories=genre))
genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance']
modals = ['can', 'could', 'may', 'might', 'must', 'will']
cfd.tabulate(conditions=genres, samples=modals)
```

Exercises

- We uploaded the exercises and slides of our lab to blackboard. Check it out.

- Exercises in the class:
- Sentence parsing
- Gender testing
- Extracting NTNU Job post.

Exercises

Sentence parsing:

- Pick one text file as your *Corpus* source.

- Write code to extract all sentences of the corpus.

Exercises

Gender testing:

- Analyze gendered words in a corpus.

- Print the percentage of gendered words, sentences in the corpus

Exercises

Extracting NTNU Jobs:

- NTNU posts their “Vacancies and Job Openings" at “https://www.ntnu.edu/vacancies”

- Write a program to do:

1. Print how many jobs/vacations are currently posted.
2. Print all titles of the jobs/vacations
3. Extract the deadline of each post

Build Corpora from Tweets

- Package: tweepy → *pip install tweepy*
- Need: consumer_key, consumer_secret, access_key, access_secret at https://developer.twitter.com/

```
import nltk
Import tweepy
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth)
```

Build Corpora from Tweets

- If authentication is successful

```
import nltk
Tweets = api.user_timeline(screen_name=account,count=count, max_id=idmax)
```

- Use *tweet_mode='extended'* to get full text of tweets

1. retweeted_status.full_text
2. full_text

- Tokenization : hashtags (#), urls, emoji.

Next Lab

- Remember submitting exercise solutions
- Deadline for solutions of Lab 1 is: **27th Jan 2020**
- Next Exercise Lab: Lab 2
  - **28th Jan 2020**
  - Content: **Tagging**
  - Time **14:15 pm - 16:00 pm**
  - Place**: F4 Gamle fysikk**

Q&A