Natural Language Processing

Dr. Karen Mazidi

Part One: Foundations

Topics

- NLTK
- Tokenization
- Text preprocessing

Quizzes

- No quiz on this material

Homework

- Assignment 1: Text Processing

NLTK

- Natural Language Toolkit (https://www.nltk.org)
  - Open-source NLP functions
- Companion book: https://www.nltk.org/book
- API: https://www.ntlk.org/api/nltk.htm

Install:

- First, install with pip/pip3
- Then do this:

```
$python or $python3
>>> import nltk
>>> nltk.download()
```

- Download all if space available, else “book” data

NLTK in Google Colab

- You can import NLTK in colab
- But you have to download resources

```
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
```

Tokenization

Tokenize – break text into smaller units

Tokenize aka word tokenize: break into tokens, which are words, numbers, punctuation

Use nltk word_tokenize

Breaking text into sentences is called sentence tokenization or sentence segmentation

Use nltk sent_tokenize

Tokenize

- See notebooks in GitHub
- End of chapter reference

Reference 3.4.1 Tokenize Text

```
# returns a list of tokens including punctuation
from nltk import word_tokenize
tokens = word_tokenize(text)
```

Reference 3.4.2 Tokenize Sentences

```
# returns a list of sentences
from nltk import sent_tokenize
sentences = sent_tokenize(text)
```

Preprocessing text aka text normalization

- lower case
- remove punctuation
- remove digits or replace with NUM
- remove noise in messy data
- stem words by removing affixes
- lemmatize words by reducing to root form (lemma)
- remove stop words

Normalization

- Makes text more analyzable for applications
- Which steps depend on what you are doing with the text and can depend on the domain and how clean the text is
- Often not done for huge data like all of Wikipedia

- Ex: combining Education, education, educational, educationally into education

Remove punctuation and/or numbers

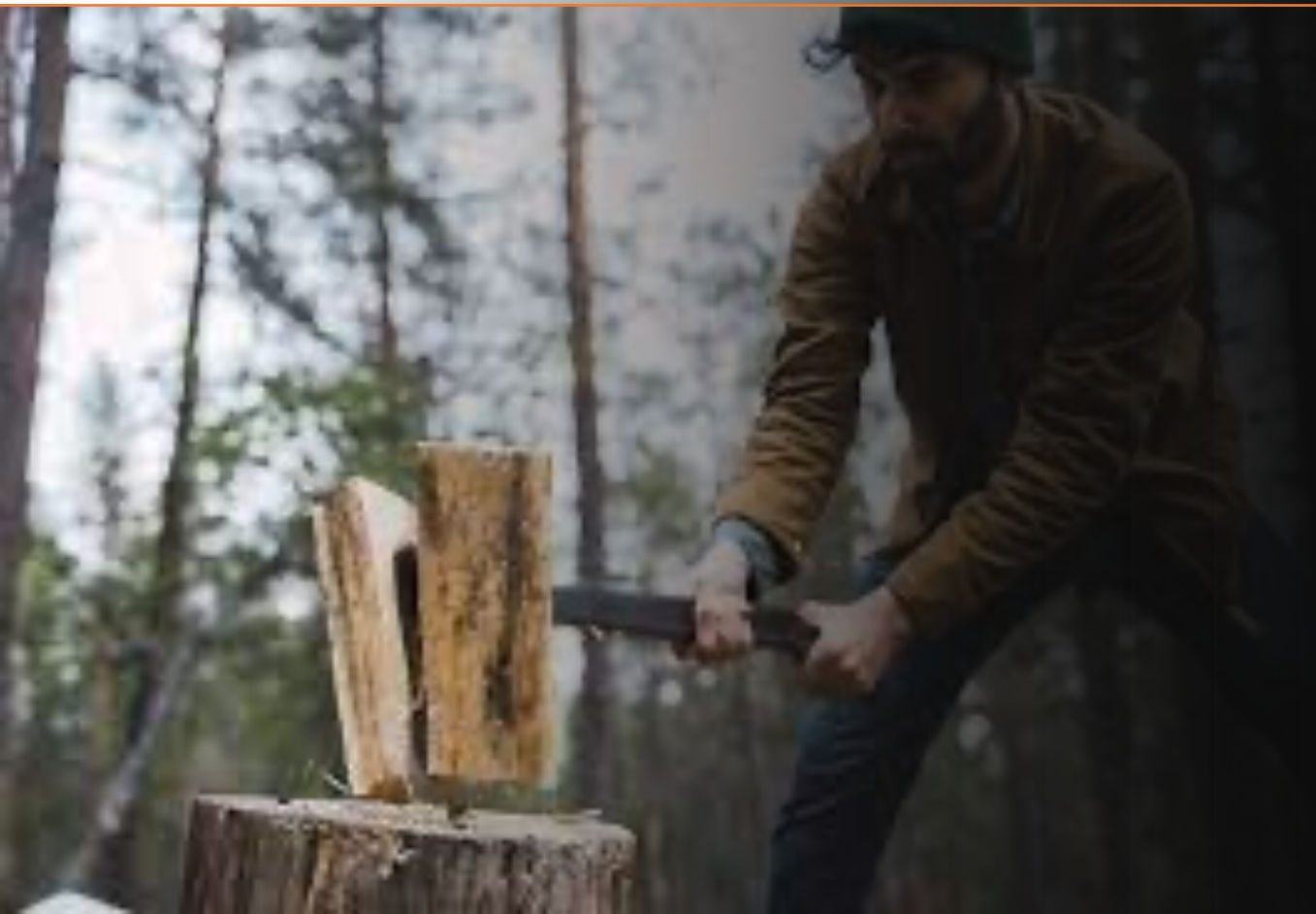
- See notebooks in GitHub
- End of chapter reference

Reference 3.4.3 Remove Punctuation and Numbers

```
# returns a modified string; also lower cased
import re
text = re.sub(r'[.?!,:;()\-\n\d]',' ', raw_text.lower())
```

nce 3.4.4 Stem Tokens

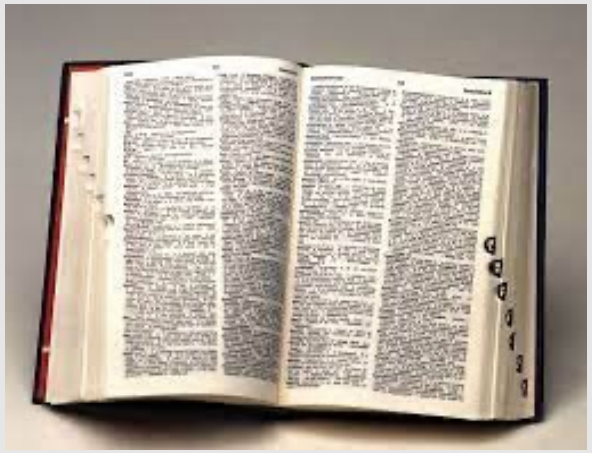
```
urns a list of stemmed tokens (includes pun
nltk.stem.porter import *
er = PorterStemmer()
ed = [stemmer.stem(t) for t in tokens]
```



Stemming: remove affixes

- Can be too aggressive, ex: ‘university’ → ‘univers’
  - ‘Texas’ and ‘Dallas’ become ‘Texa’ and ‘Dalla’

Lemmatizing: reduce to lemma



Reference 3.4.5 Lemmatize Text

```
# returns a list of lemmas (includes punctuation)
from nltk.stem import WordNetLemmatizer()
wnl = WordNetLemmatizer()
lemmas = [wnl.lemmatize(t) for t in tokens]
```

Remove stopwords

to remove or not is a critical choice

```
> stopwords("english")
  [1] "i"          "me"         "my"         "myself"     "we"
  [6] "our"        "ours"       "ourselves"  "you"        "your"
 [11] "yours"      "yourself"   "yourselves" "he"         "him"
 [16] "his"        "himself"    "she"        "her"        "hers"
 [21] "herself"    "it"         "its"        "itself"     "they"
 [26] "them"       "their"      "theirs"     "themselves" "what"
 [31] "which"      "who"        "whom"       "this"       "that"
 [36] "these"      "those"      "am"         "is"         "are"
 [41] "was"        "were"       "be"         "been"       "being"
 [46] "have"       "has"        "had"        "having"     "do"
```

Reference 3.4.6 Remove Stopwords

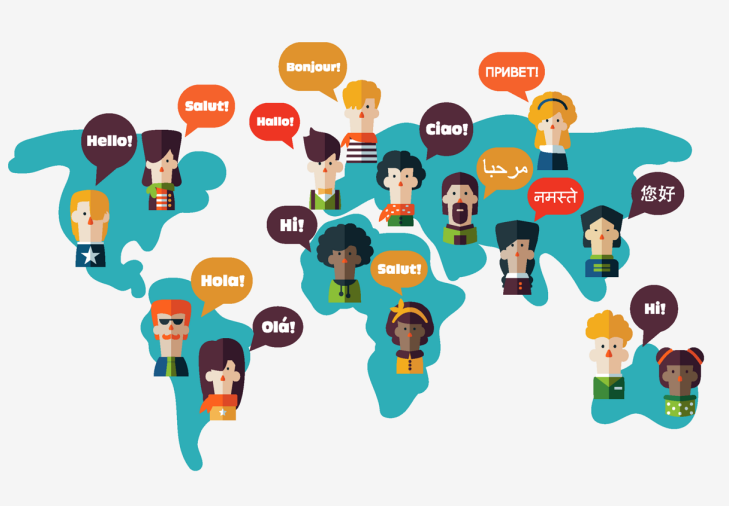
```
# returns a list of tokens that are not stopwords
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
tokens = [t for t in tokens if not t in stop_words]
```

NLTK in other languages

Varies by function

Code 3.3.1 — NLTK. Other Languages.

```
import nltk.data
tokenizer = nltk.data.load('tokenizers/punkt/spanish.pickle')
>>> tokenizer.tokenize('Hola mi amor. Como estas?')
['Hola mi amor.', 'Como estas?']
```



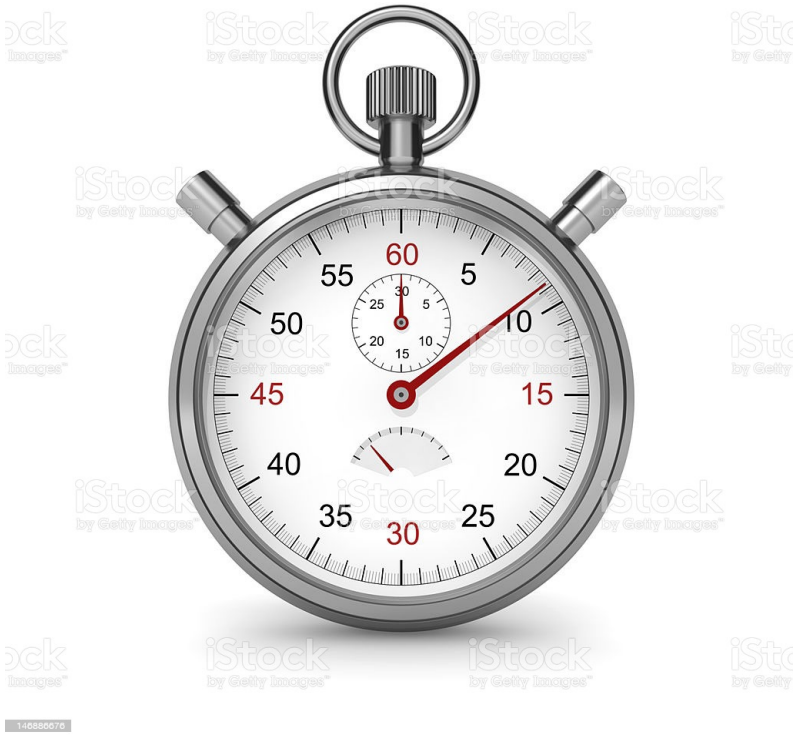
Timing code

- Making code more efficient

Reference 3.4.7 Timing Code

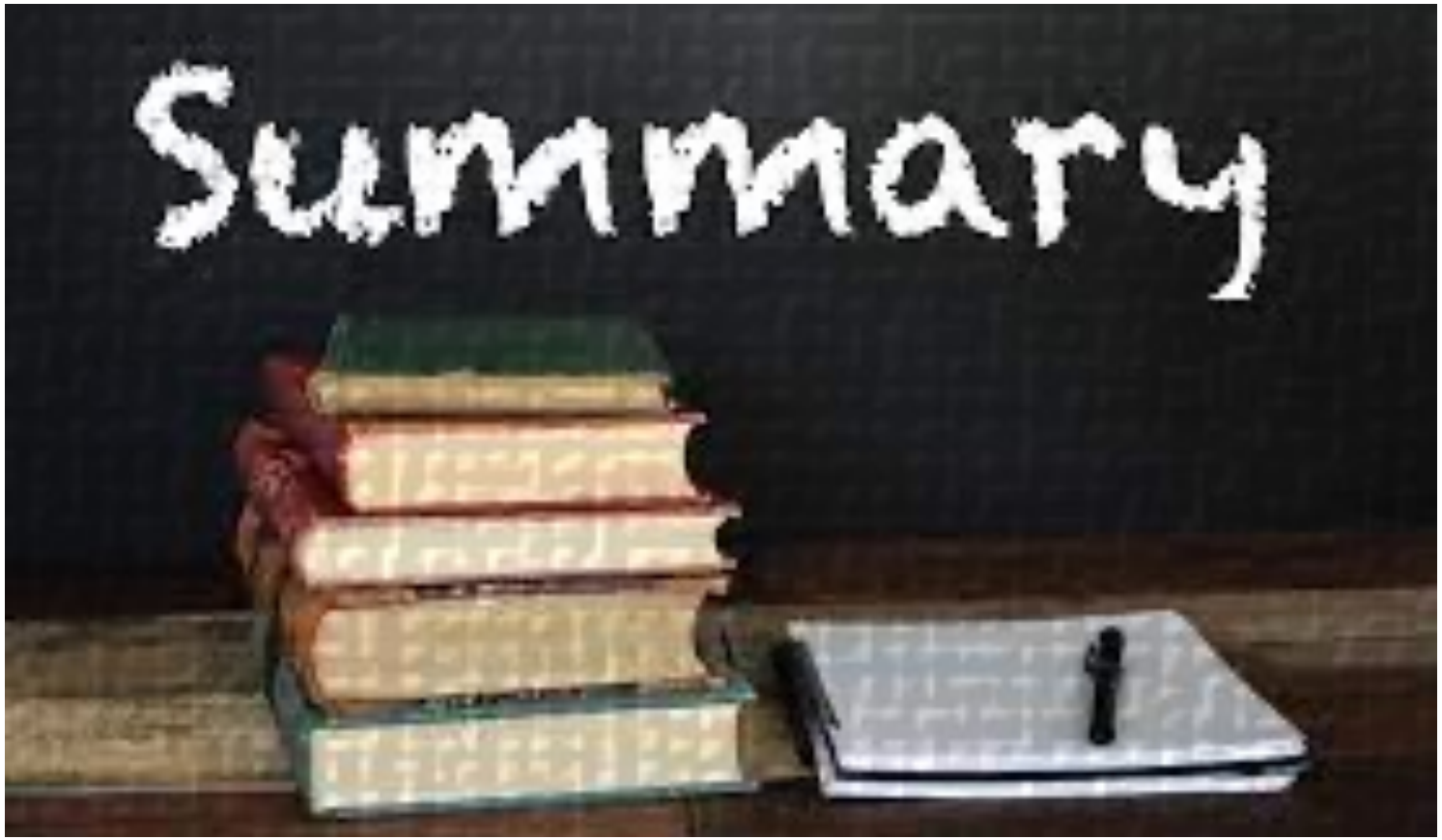
```
import timeit

start_time = timeit.default_timer()
# do something
stop_time = timeit.default_timer()
print('Time:', stop_time - start_time)
```



Python Code Examples

- Multi-file examples in GitHub:
  - Part 1 Foundations -> Chapter 02 -> Python Sample Code
- Preprocessing code:
  - Part 1 Foundations -> Chapter 03 -> Preprocessing.py

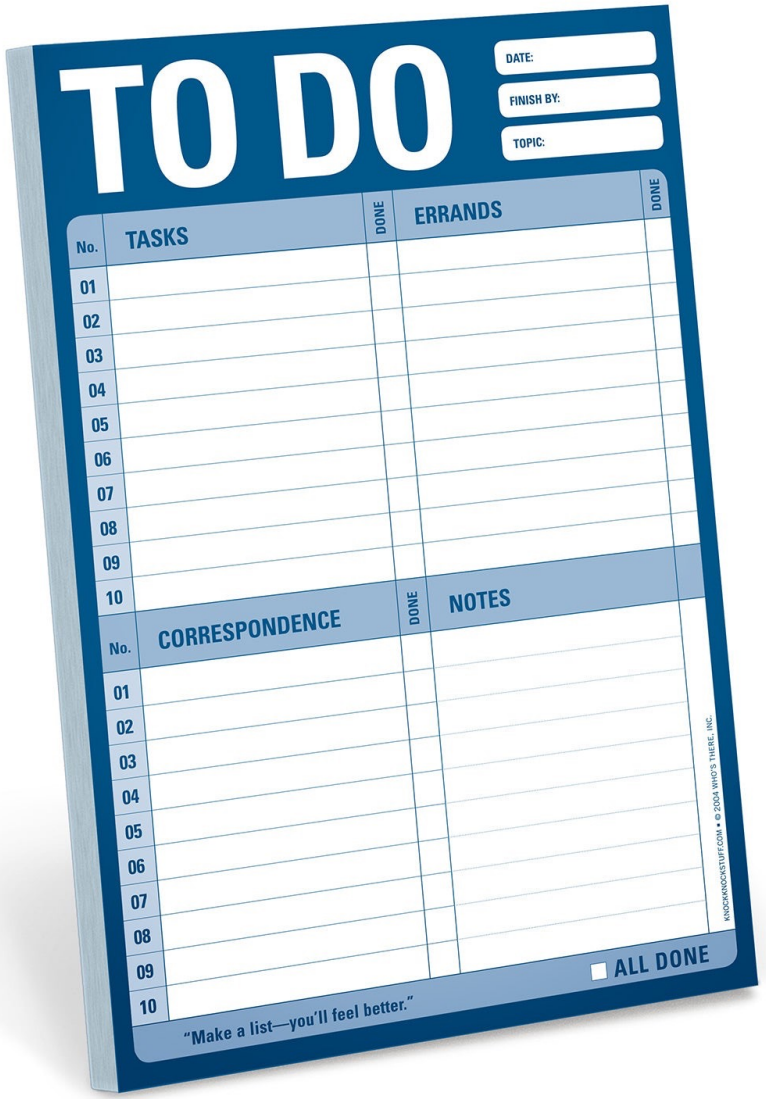


Essential points to note

- How to use NLTK
- How to preprocess text using regex

To Do

- Quiz Chapter 2 Python
- Portfolio 0: Setup
- Portfolio 1: Text Processing



Next topic

Linguistics 101