Text Analysis for Social Sciences in Python

Week 4: Text as Data – Overview, Feasibility, Pre-processing Data

WINTER SEMESTER 21/22

HUYEN NGUYEN

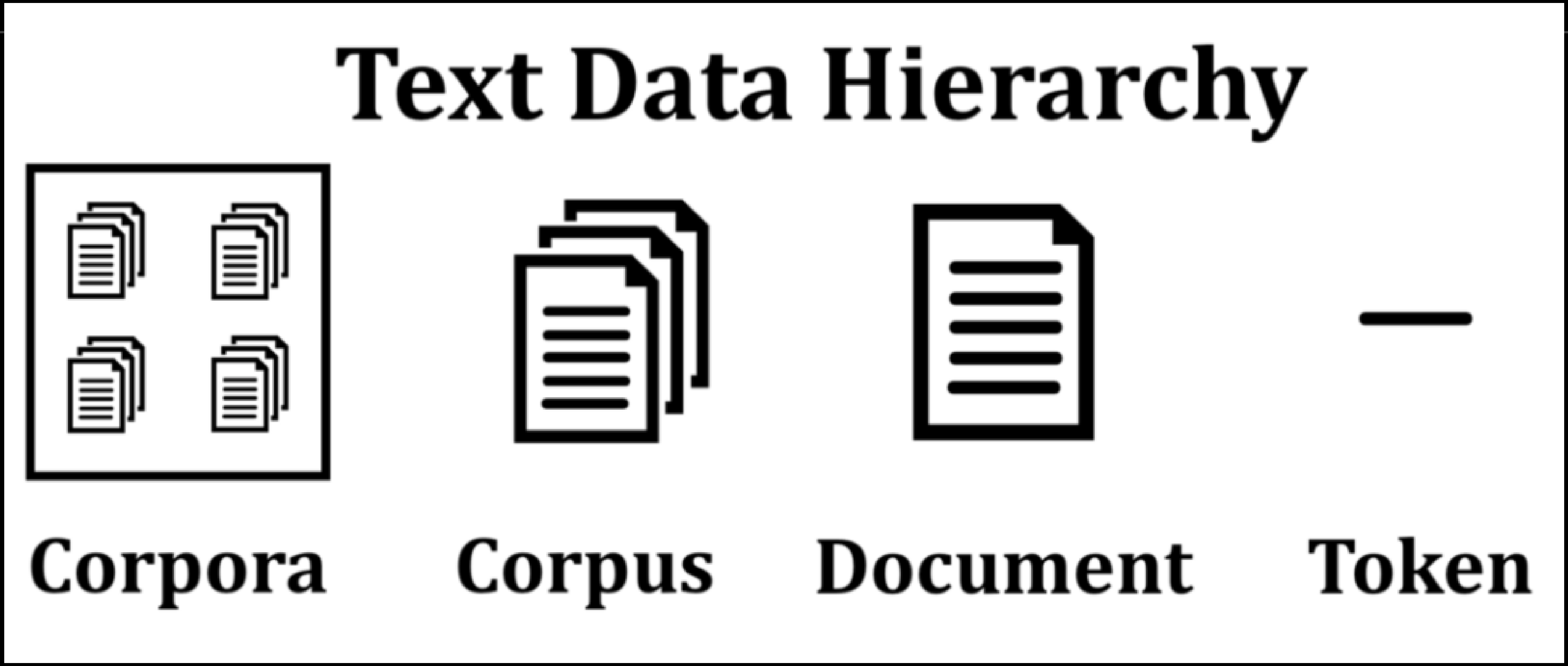
HTTP://HUYENTTNGUYEN.COM

Last week…

Matplotlib
Pandas
Nltk
Scikit-learn

…any questions with the materials? Home exercises?

Text as Data?



https://medium.com/@raufulazam95218/basics-of-natural-language-processing-nlp-e2b75d2e1dfe

Text Documents as Data

- **Documents** $\mathcal{D}$ =Text data as a sequence of characters
- **Corpus** $\mathcal{C}$ = the set of documents
- Token t = words, phrases, or N-grams (i.e. the group of n words together).
- Text data is **unstructured** & **high-dimensional.**

  → The information we want is mixed together with (lots of) information we don't.

  → All text data approaches will throw away some information.

GOAL = retain valuable information

  → Your research question = the key guide.

Text Documents as Data

Suppose we have a sample of documents d, each of which is w words long.

Suppose that each word is drawn from a vocabulary of p possible words.

→ Unique representation of these documents has dimension: p w

E.g: A sample of 30-word Twitter messages, which use 1000 most common words in the English language, has roughly as many dimensions as there are atoms in the universe (!).

An overview of text as data analysis procedure

(Grimmer & Stewart 2013, *Political Analysis*)

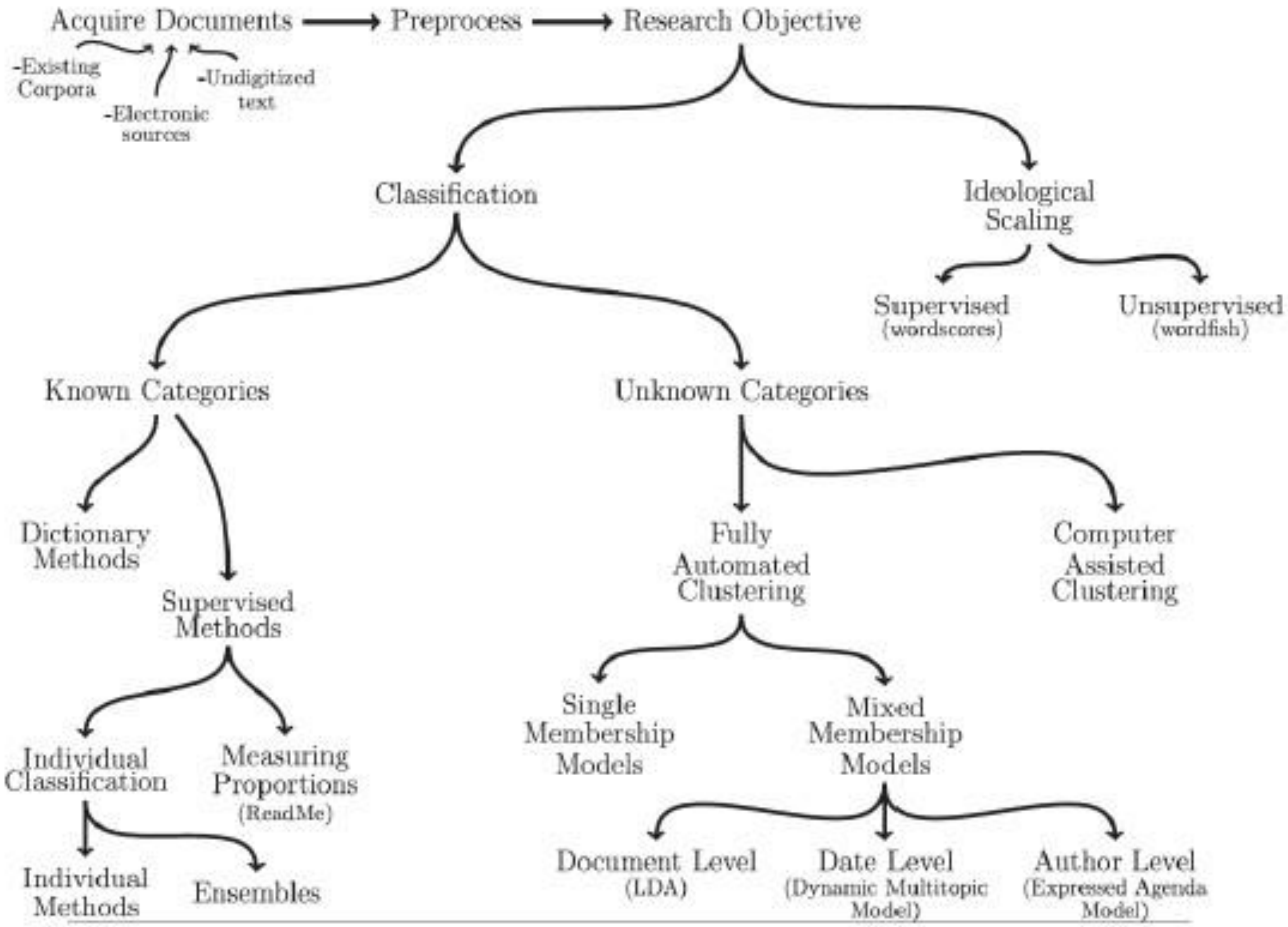


Fig. 1 An overview of text as data methods.

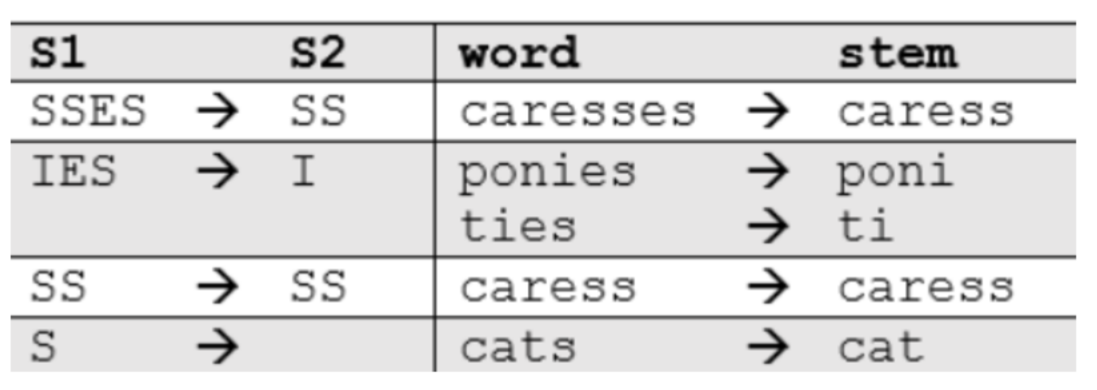
1. From text to numbers (Encoding)

- Represent raw text as a numerical array
- Convert texts to features
  - Words
  - Phrases
  - Syntactic/semantic relations
- Feature selection / dimension reduction to exclude irrelevant information
  - Stop words
  - Stemming
  - Lemmatization

Stemming?

Producing morphological variants of a root/base word.

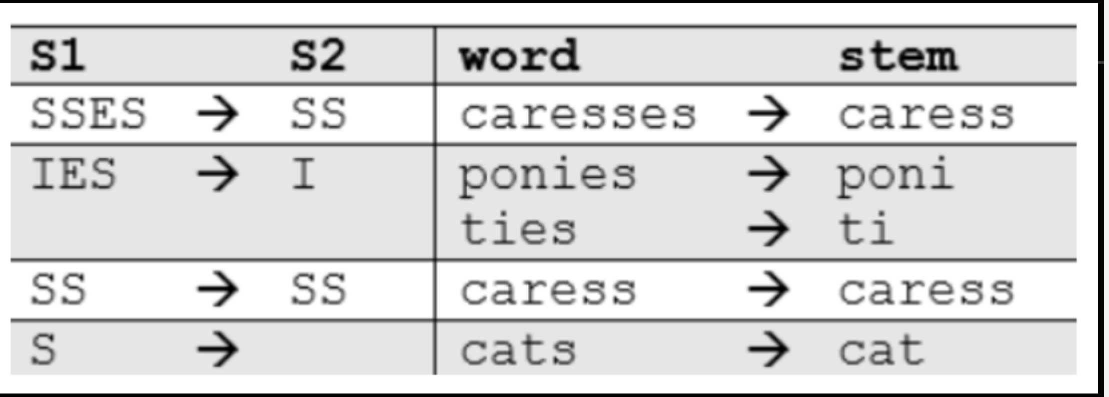
Stemming



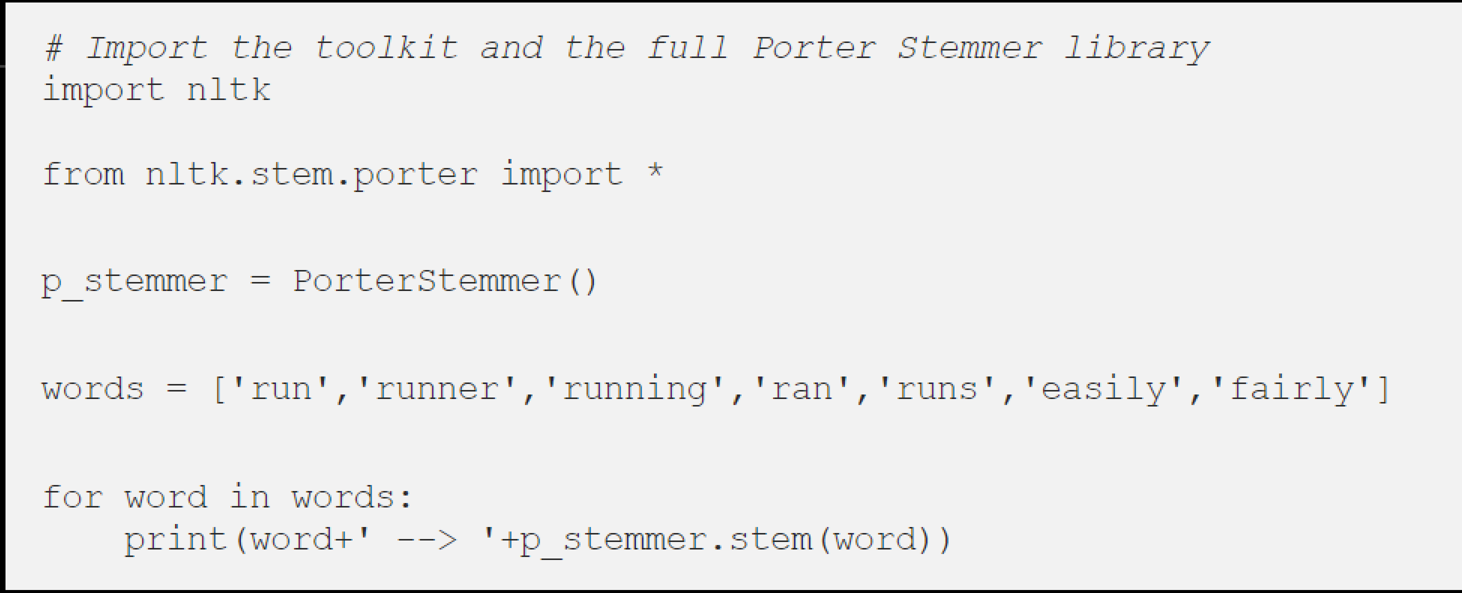
S1		S2	word		stem
SSES	→	SS	caresses	→	caress
IES	→	I	ponies ties	→ →	poni ti
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Example: Porter stemmer

Stemming



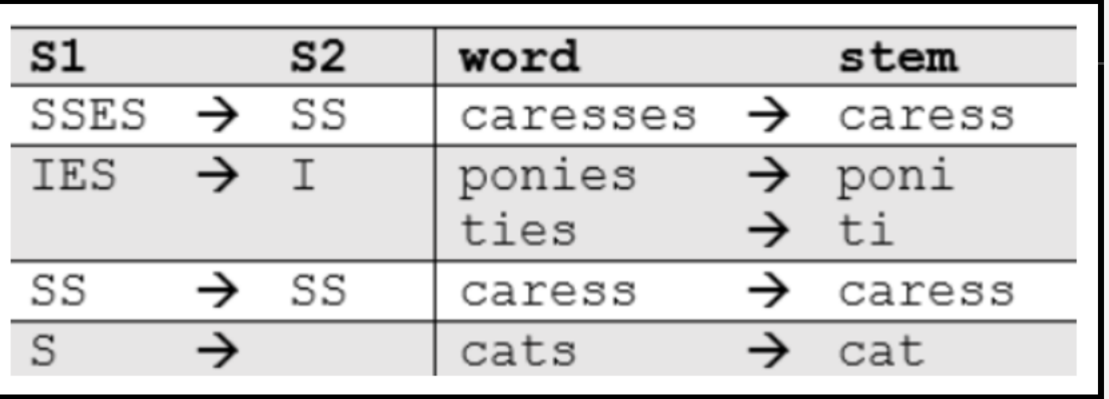
S1		S2	word		stem
SSES	→	SS	caresses	→	caress
IES	→	I	ponies ties	→ →	poni ti
SS	→	SS	caress	→	caress
S	→		cats	→	cat



```
# Import the toolkit and the full Porter Stemmer library
import nltk

from nltk.stem.porter import *


p_stemmer = PorterStemmer()


words = ['run','runner','running','ran','runs','easily','fairly']


for word in words:
    print(word+' --> '+p_stemmer.stem(word))
```

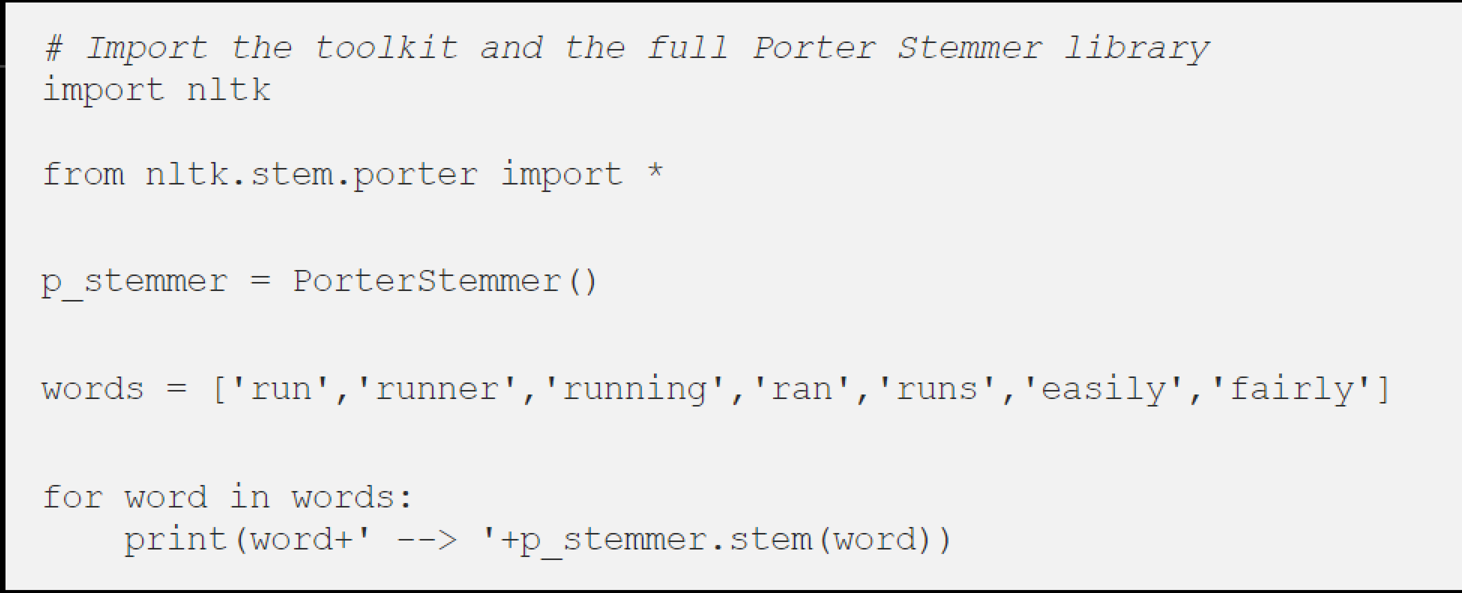
Example: Porter stemmer

Stemming in practice

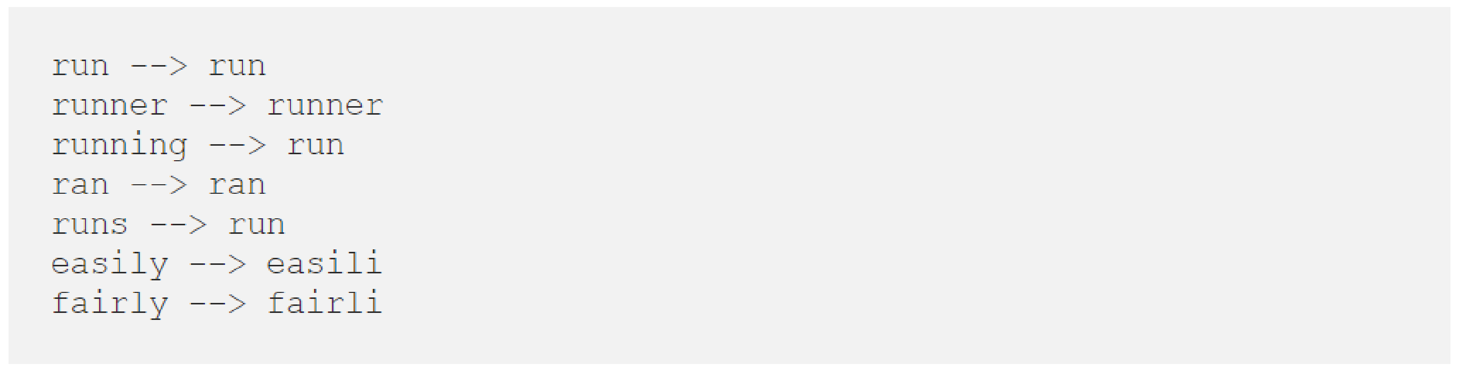


S1		S2	word		stem
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
			ties	→	ti
SS	→	SS	caress	→	caress
S	→		cats	→	cat

<u>Example</u>: Porter stemmer
=> Check
SnowballStemmer



```
# Import the toolkit and the full Porter Stemmer library
import nltk

from nltk.stem.porter import *


p_stemmer = PorterStemmer()


words = ['run','runner','running','ran','runs','easily','fairly']


for word in words:
    print(word+' --> '+p_stemmer.stem(word))
```



```
run --> run
runner --> runner
running --> run
ran --> ran
runs --> run
easily --> easili
fairly --> fairli
```

Lemmatization

- Lemmatization <u>looks beyond </u>word reduction and considers a language’s full vocabulary to apply a morphological analysis to words.

  E.g: The lemma of ‘was’ = ‘be’ , ‘mice’ = ‘mouse’.

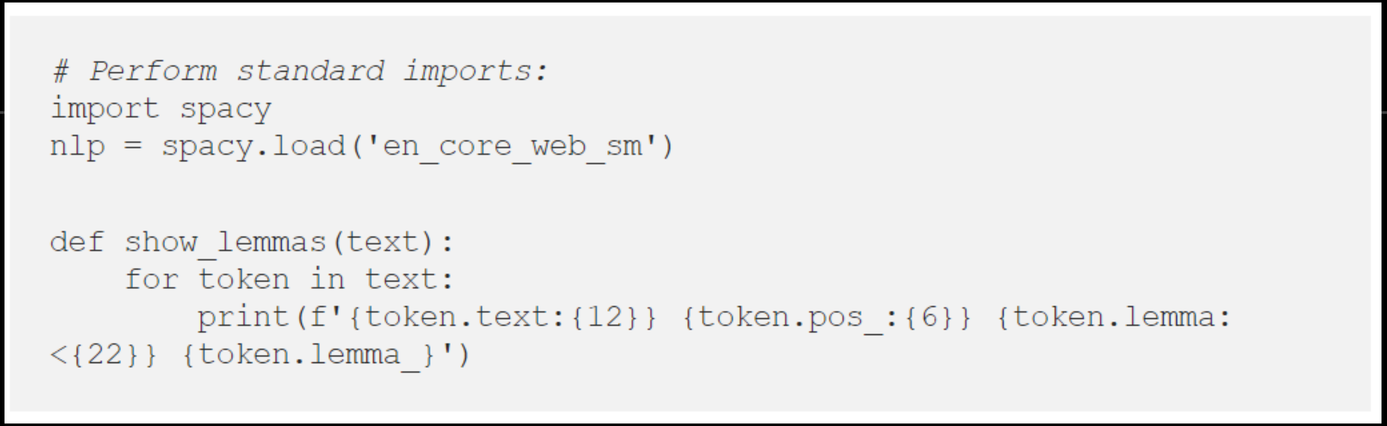
Lemmatization

- Lemmatization looks beyond word reduction and considers a language’s full vocabulary to apply a morphological analysis to words.
  - E.g: The lemma of ‘was’ = ‘be’ , ‘mice’ = ‘mouse’.
- Lemmatization looks at surrounding text to determine a given word’s part of speech, it does not categorize phrases.

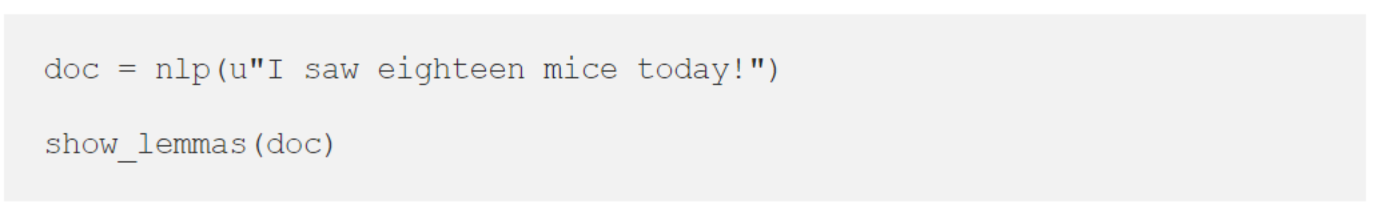
Lemmatization

- Lemmatization looks beyond word reduction and considers a language's full vocabulary to apply a morphological analysis to words.
  - E.g: The lemma of 'was' = 'be' , 'mice' = 'mouse'.
- Lemmatization looks at surrounding text to determine a given word's part of speech, it does not categorize phrases.
- Lemmatization is typically more informative than simple stemming => Spacy opts to have only Lemmatization, instead of stemming.

Lemmatization in practice

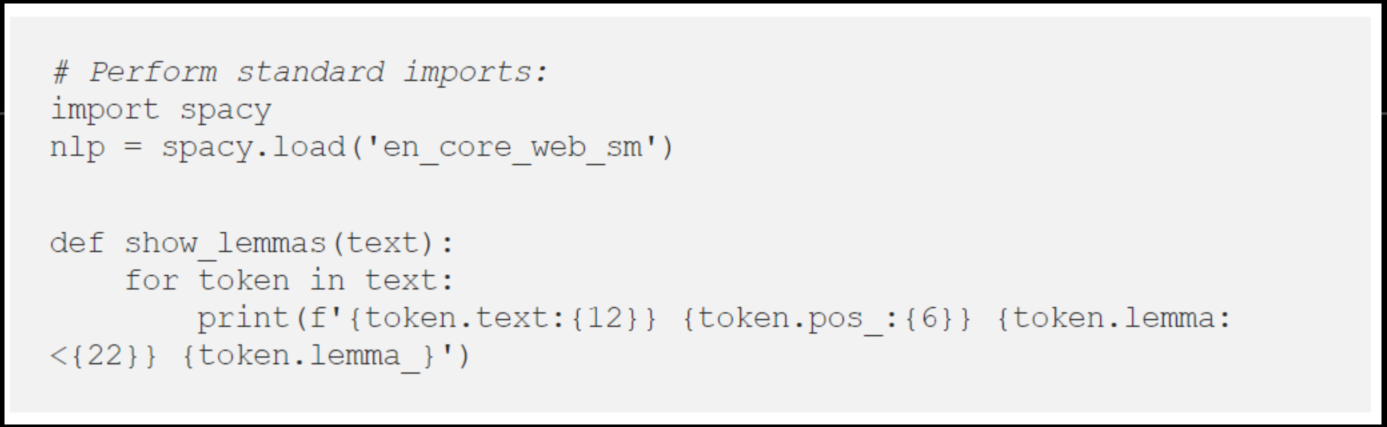


```
# Perform standard imports:
import spacy
nlp = spacy.load('en_core_web_sm')


def show_lemmas(text):
    for token in text:
        print(f'{token.text:{12}} {token.pos_:{6}} {token.lemma:
<{22}} {token.lemma_}')
```

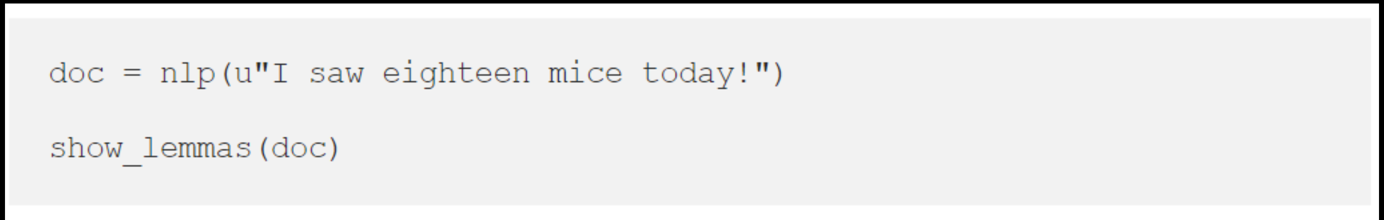


```
doc = nlp(u"I saw eighteen mice today!")

show_lemmas(doc)
```

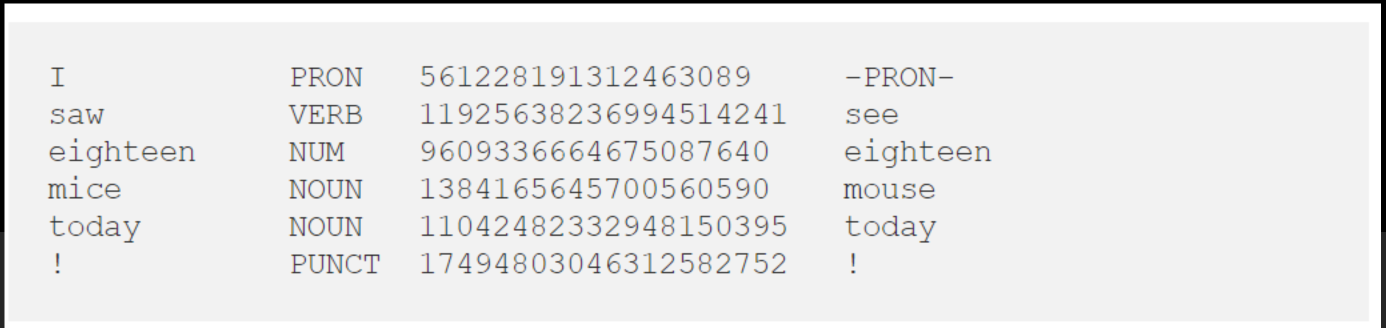
Lemmatization in practice



```
# Perform standard imports:
import spacy
nlp = spacy.load('en_core_web_sm')


def show_lemmas(text):
    for token in text:
        print(f'{token.text:{12}} {token.pos_:{6}} {token.lemma:
<{22}} {token.lemma_}')
```



```
doc = nlp(u"I saw eighteen mice today!")

show_lemmas(doc)
```



```
I            PRON   561228191312463089     -PRON-
saw          VERB   11925638236994514241   see
eighteen     NUM    9609336664675087640    eighteen
mice         NOUN   1384165645700560590    mouse
today        NOUN   11042482332948150395   today
!            PUNCT  17494803046312582752   !
```

2. Corpora Interpretation: Supervised Methods

- Dictionary methods for targeted studies (classification with known categories)
  - E.g. spam filters, sentiment analysis

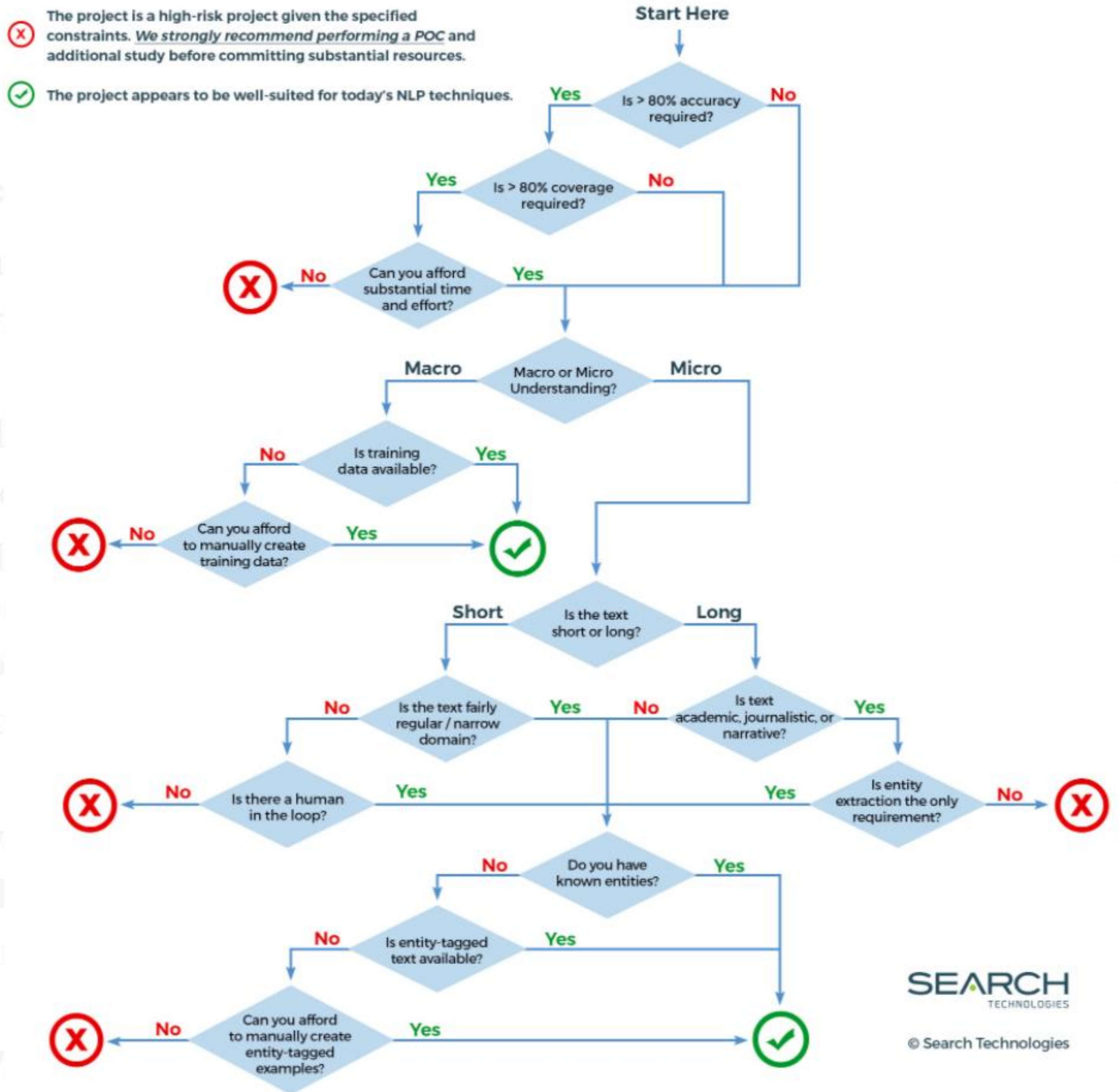
- Applying regressors and classifiers to text features

3. Corpora Interpretation: Unsupervised Methods

- Topic models (topic clustering/categorization)
  - E.g: Central bank communication and stock market reaction
- Word/Document embeddings
  - **E.g: Word embedding for isolating dimensions of language to analyze** values, attitudes, and ideologies
- Text similarity, word clouds, and automatic text summarization

→ Your homework: compare the pros and cons of these methods, along with the list of applications these methods are used.

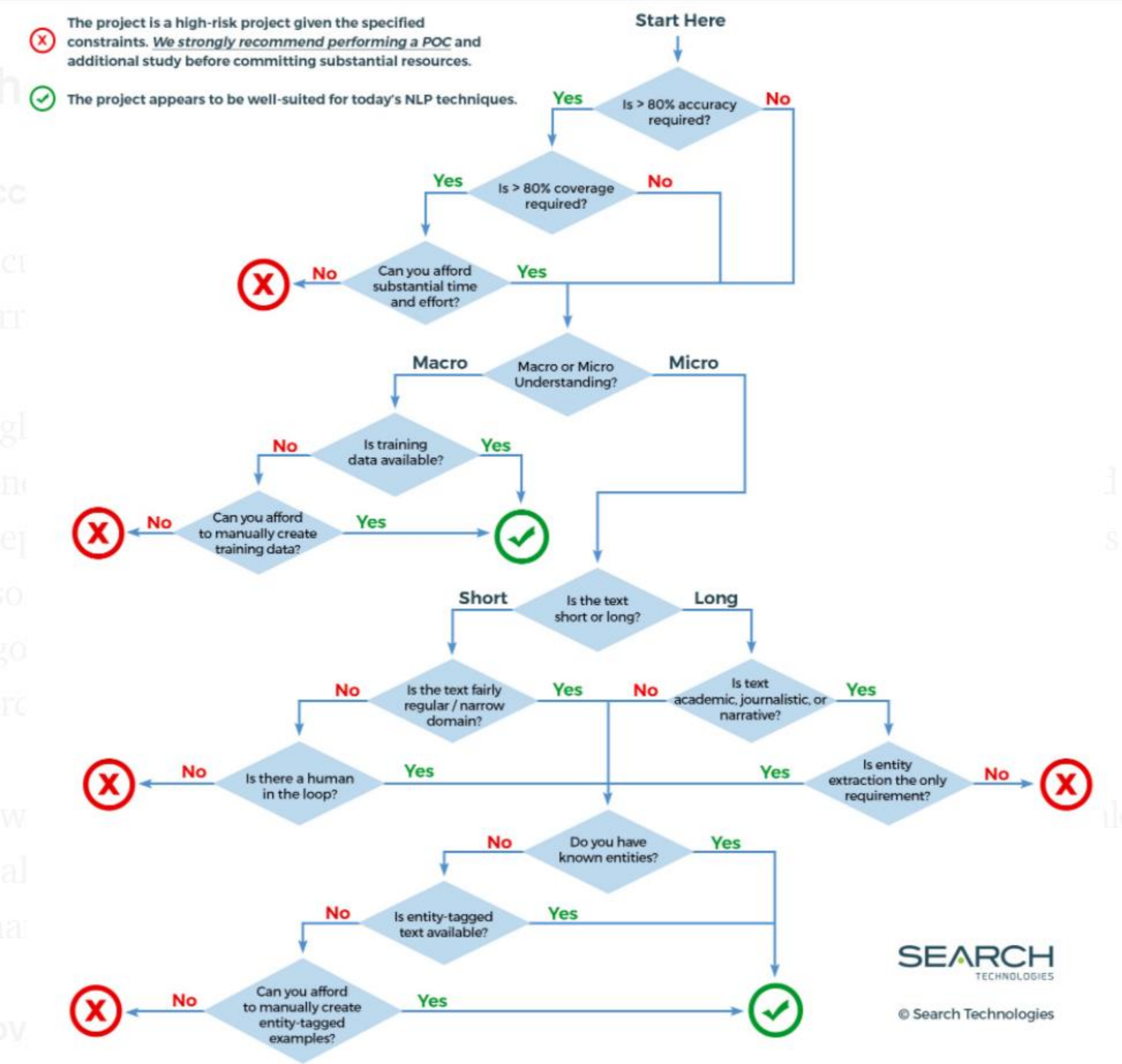
Is your NLP project feasible?



The project is a high-risk project given the specified constraints. *We strongly recommend performing a POC* and additional study before committing substantial resources.

The project appears to be well-suited for today's NLP techniques.

Start Here
Is > 80% accuracy required?
Yes
No
Is > 80% coverage required?
Yes
No
Can you afford substantial time and effort?
No
Yes
Macro or Micro Understanding?
Macro
Micro
Is training data available?
No
Yes
Can you afford to manually create training data?
No
Yes
Is the text short or long?
Short
Long
Is the text fairly regular / narrow domain?
No
Yes
Is text academic, journalistic, or narrative?
No
Yes
Is there a human in the loop?
No
Yes
Is entity extraction the only requirement?
Yes
No
Do you have known entities?
No
Yes
Is entity-tagged text available?
No
Yes
Can you afford to manually create entity-tagged examples?
No
Yes
SEARCH TECHNOLOGIES
© Search Technologies



ⓧ The project is a high-risk project given the specified constraints. *We strongly recommend performing a POC* and additional study before committing substantial resources.

✓ The project appears to be well-suited for today's NLP techniques.

Start Here
Is > 80% accuracy required?
Yes
No
Is > 80% coverage required?
Yes
No
Can you afford substantial time and effort?
No
Yes
Macro or Micro Understanding?
Macro
Micro
Is training data available?
No
Yes
Can you afford to manually create training data?
No
Yes
Is the text short or long?
Short
Long
Is the text fairly regular / narrow domain?
No
Yes
Is text academic, journalistic, or narrative?
No
Yes
Is there a human in the loop?
No
Yes
Is entity extraction the only requirement?
Yes
No
Do you have known entities?
No
Yes
Is entity-tagged text available?
No
Yes
Can you afford to manually create entity-tagged examples?
No
Yes
SEARCH TECHNOLOGIES

© Search Technologies

Is your NLP project feasible?

Your group homework:

1) What is your research idea?
2) Do you think it is feasible? Why? Why not?

Practice time ☺

Open your Google Colab/ Jupyter Notebook

Practice time ☺

https://github.com/httn21uhh/Text-Analysis-for-Social-Sciences-in-Python

There is also the .ipynb file and the data set for your reference

- Download it
- Run it on your own laptop
- This is NOT meant for passive scrolling!

This week…

Open your Google Colab/ Jupyter Notebook

Pre-processing data

This week…

Open your Google Colab/ Jupyter Notebook

Pre-processing data

- Follow closely the illustrated examples and replicate yourself as the session proceeds.
- Raise your hands to ask questions at any point, including when you think things go too fast/slow for you.