Programming for Social Scientists: Text Mining I

黃瀚萱

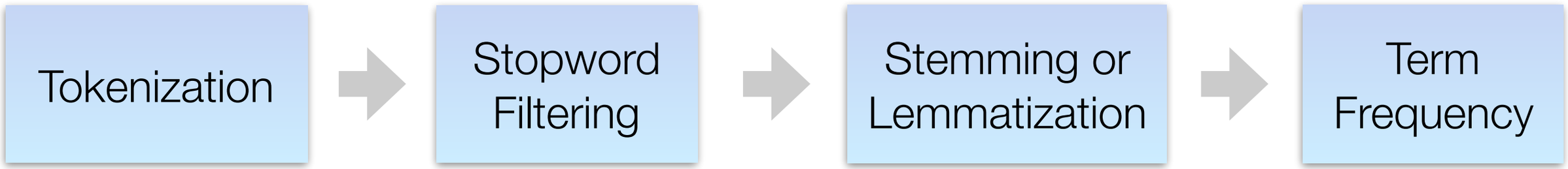
Goal

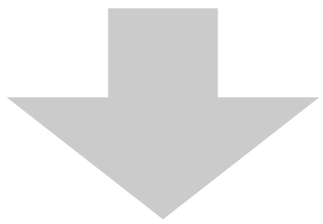
- From unstructured sequence to meaningful symbols.

Hindu refers to any person who regards themselves as culturally, ethnically, or religiously adhering to aspects of Hinduism.[1][2] It has historically been used as a geographical, cultural, and later religious identifier for people indigenous to the Indian subcontinent.[3][4] The historical meaning of the term Hindu has evolved with time. Starting with the Persian and Greek references to the land of the Indus in the 1st millennium BCE through the texts of the medieval era,[5] the term Hindu implied a geographic, ethnic or cultural identifier for people living in the Indian subcontinent around or beyond the Sindhu (Indus) river.[6] .....

Word	Count
Hindu	11
term	6
century	6
Hindus	6
use	4
Indian	4
text	3
subcontinent	3
India	3
cultural	3
world	3
religious	3

Fundamental Text Processing





Tokenization

- Similar to Chinese word segmentation

Hindu refers to any person who regards themselves as culturally, ethnically, or religiously adhering to aspects of Hinduism.[1][2] It has historically been used as a geographical, cultural, and later religious identifier for people indigenous to the Indian subcontinent.[3][4] The historical meaning of the term Hindu has evolved with time. Starting with the Persian and Greek references to the land of the Indus in the 1st millennium BCE through the texts of the medieval era,[5] the term Hindu implied a geographic, ethnic or cultural identifier for people living in the Indian subcontinent around or beyond the Sindhu (Indus) river.[6] .....

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally', ',',
'ethnically', ',', 'or', 'religiously',
'adhering', 'to', 'aspects', 'of', 'Hinduism',
'.', '[', '1', ']', '[', '2', ']', 'It', 'has',
'historically', 'been', 'used', 'as', 'a',
'geographical', ',', 'cultural', ',', 'and',
'later', 'religious', 'identifier', 'for',
'people', 'indigenous', 'to', 'the', 'Indian',
'subcontinent', '.', '[', '3', ']', '[', '4',
']', 'The', 'historical', 'meaning', 'of', 'the',
'term', 'Hindu', 'has', 'evolved', 'with',
'time', '.', 'Starting', 'with', 'the',
'Persian', 'and', 'Greek', 'references', 'to',
'the', 'land', 'of', 'the', 'Indus', 'in', 'the',
'1st', 'millennium', 'BCE', 'through', 'the',
'texts', 'of', 'the', 'medieval', 'era', ',',
'[', '5', ']', 'the', 'term', 'Hindu', 'implied',
'a', 'geographic', ',', 'ethnic', 'or',
'cultural', 'identifier', 'for', 'people',
'living', 'in', 'the', 'Indian', 'subcontinent',
'around', 'or', 'beyond', 'the', 'Sindhu', '(',
'Indus', ')', 'river', '.', '[', '6', ']', 'By',
'the', '16th', 'century', ',', 'the', 'term',
'began', 'to', 'refer', 'to', 'residents', 'of',
'the', 'subcontinent', 'who', 'were', 'not',
'Turkic', 'or', 'Muslims', ...]
```

Tokenization with the split() function

- split(): Return a list of the words in the string, using sep as the delimiter string.

```
text = "Hindu refers to any person who regards themselves as culturally, ethnically, or religiously adhering to aspects of Hinduism.[1][2]"

tokens = text.split(" ")

print(tokens)
```

```
['Hindu', 'refers', 'to', 'any', 'person', 'who', 'regards', 'themselves', 'as', 'culturally,', 'ethnically,', 'or', 'religiously', 'adhering', 'to', 'aspects', 'of', 'Hinduism.[1][2]']
```

Tokenization with NLTK

- A better solution
  - Free NLTK book: https://www.nltk.org/book/

```
from nltk.tokenize import word_tokenize
```

```
tokens = word_tokenize(" ")
```

```
print(tokens)
```

```
['Hindu', 'refers', 'to', 'any', 'person', 'who', 'regards',
'themselves', 'as', 'culturally', ',', 'ethnically', ',', 'or',
'religiously', 'adhering', 'to', 'aspects', 'of', 'Hinduism', '.'
'[', '1', ']', '[', '2', ']']
```

Sample (Problematic) Code for Tokenization

```
def tokenize(text):
  tokens = []
  tok = ""
  for ch in text:
    if ch == " ":
      if tok:
        tokens.append(tok)
        tok = ""
    else:
      tok += ch
return tokens
```

Quiz:
Point out 2 problems in this code

Counting

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally', ',',
'ethnically', ',', 'or', 'religiously', 'adhering',
'to', 'aspects', 'of', 'Hinduism', '.', '[', '1',
']', '[', '2', ']', 'It', 'has', 'historically',
'been', 'used', 'as', 'a', 'geographical', ',',
'cultural', ',', 'and', 'later', 'religious',
'identifier', 'for', 'people', 'indigenous', 'to',
'the', 'Indian', 'subcontinent', '.', '[', '3', ']',
'[', '4', ']', 'The', 'historical', 'meaning', 'of',
'the', 'term', 'Hindu', 'has', 'evolved', 'with',
'time', '.', 'Starting', 'with', 'the', 'Persian',
'and', 'Greek', 'references', 'to', 'the', 'land',
'of', 'the', 'Indus', 'in', 'the', '1st',
'millennium', 'BCE', 'through', 'the', 'texts',
'of', 'the', 'medieval', 'era', ',', '[', '5', ']',
'the', 'term', 'Hindu', 'implied', 'a',
'geographic', ',', 'ethnic', 'or', 'cultural',
'identifier', 'for', 'people', 'living', 'in',
'the', 'Indian', 'subcontinent', 'around', 'or',
'beyond', 'the', 'Sindhu', '(', 'Indus', ')',
'river', '.', '[', '6', ']', 'By', 'the', '16th',
'century', ',', 'the', 'term', 'began', 'to',
'refer', 'to', 'residents', 'of', 'the',
'subcontinent', 'who', 'were', 'not', 'Turkic',
'or', 'Muslims', ...]
```

Word	Count
the	35
,	33
[	30
]	30
.	18
and	16
of	14
to	12
in	12
Hindu	11

Counting

```
from collections import Counter

tokens = ['the', 'cat', 'is', 'a', 'nice', 'cat', 'who',
'treats', 'the', 'other', 'cat', 'friendly']
word_count = Counter(tokens)
```

```
Counter({'cat': 3, 'the': 2, 'is': 1, 'friendly': 1, 'who':
1, 'nice': 1, 'other': 1, 'treats': 1, 'a': 1})
```

Punctuation Marks

```
import string

print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

# ASCI only
```

Removal of Punctuation Marks

```
def remove_punctuation_marks(tokens):

    clean_tokens = []

    for tok in tokens:

        if tok not in string.punctuation:

            clean_tokens.append(tok)

    return clean_tokens
```

```
['Hindu', 'refers', 'to', 'any', 'person', 'who', 'regards', 'themselves', 'as',
'culturally', 'ethnically', 'or', 'religiously', 'adhering', 'to', 'aspects', 'of',
'Hinduism', '1', '2', 'It', 'has', 'historically', 'been', 'used', 'as', 'a',
'geographical', 'cultural', 'and', 'later', 'religious', 'identifier', 'for',
'people', 'indigenous', 'to', 'the', 'Indian', 'subcontinent', '3', '4', ...]
```

Removing All Non-alphabet Tokens

```
def remove_punctuation_marks(tokens):

    clean_tokens = []

    for tok in tokens:

        if tok.isalpha():

            clean_tokens.append(tok)

    return clean_tokens
```

```
['Hindu', 'refers', 'to', 'any', 'person', 'who', 'regards', 'themselves', 'as', 'culturally',
'ethnically', 'or', 'religiously', 'adhering', 'to', 'aspects', 'of', 'Hinduism', 'It', 'has',
'historically', 'been', 'used', 'as', 'a', 'geographical', 'cultural', 'and', 'later',
'religious', 'identifier', 'for', 'people', 'indigenous', 'to', 'the', 'Indian',
'subcontinent', 'The', 'historical', 'meaning', 'of', 'the', 'term', 'Hindu', 'has', 'evolved',
'with', 'time', 'Starting', 'with', 'the', 'Persian', 'and', 'Greek', 'references', 'to',
'the', 'land', 'of', 'the', 'Indus', 'in', 'the', '1st', 'millennium', 'BCE', 'through', 'the',
'texts', 'of', 'the', 'medieval', 'era', 'the', 'term', 'Hindu', 'implied', 'a', 'geographic',
'ethnic', ...]
```

Using Python Generator

```
def remove_punctuation_marks(tokens):

        return [tok for tok in tokens if tok.isalpha()]
```

```
['Hindu', 'refers', 'to', 'any', 'person', 'who', 'regards', 'themselves',
'as', 'culturally', 'ethnically', 'or', 'religiously', 'adhering', 'to',
'aspects', 'of', 'Hinduism', 'It', 'has', 'historically', 'been', 'used', 'as',
'a', 'geographical', 'cultural', 'and', 'later', 'religious', 'identifier',
'for', 'people', 'indigenous', 'to', 'the', 'Indian', 'subcontinent', 'The',
'historical', 'meaning', 'of', 'the', 'term', 'Hindu', 'has', 'evolved',
'with', 'time', 'Starting', 'with', 'the', 'Persian', 'and', 'Greek',
'references', 'to', 'the', 'land', 'of', 'the', 'Indus', 'in', 'the', '1st',
'millennium', 'BCE', 'through', 'the', 'texts', 'of', 'the', 'medieval', 'era',
'the', 'term', 'Hindu', 'implied', 'a', 'geographic', 'ethnic', ...]
```

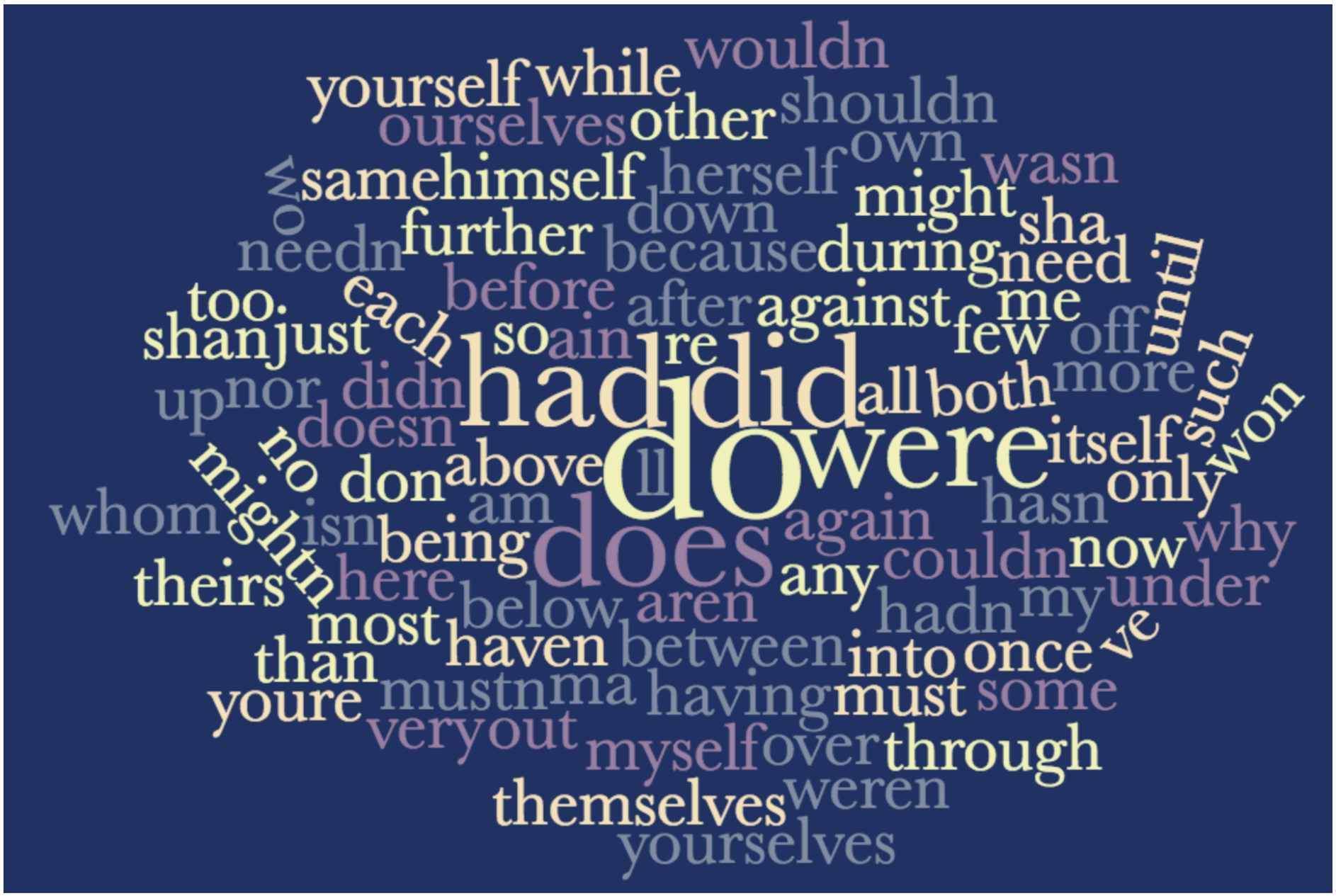
Counting Result with Punctuation Mark Removal

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally',
'ethnically', 'or', 'religiously', 'adhering', 'to',
'aspects', 'of', 'Hinduism', 'It', 'has',
'historically', 'been', 'used', 'as', 'a',
'geographical', 'cultural', 'and', 'later',
'religious', 'identifier', 'for', 'people',
'indigenous', 'to', 'the', 'Indian', 'subcontinent',
'The', 'historical', 'meaning', 'of', 'the', 'term',
'Hindu', 'has', 'evolved', 'with', 'time',
'Starting', 'with', 'the', 'Persian', 'and',
'Greek', 'references', 'to', 'the', 'land', 'of',
'the', 'Indus', 'in', 'the', '1st', 'millennium',
'BCE', 'through', 'the', 'texts', 'of', 'the',
'medieval', 'era', 'the', 'term', 'Hindu',
'implied', 'a', 'geographic', 'ethnic', 'or',
'cultural', 'identifier', 'for', 'people', 'living',
'in', 'the', 'Indian', 'subcontinent', 'around',
'or', 'beyond', 'the', 'Sindhu', 'Indus', 'river',
'By', 'the', '16th', 'century', 'the', 'term',
'began', 'to', 'refer', 'to', 'residents', 'of',
'the', 'subcontinent', 'who', 'were', 'not',
'Turkic', 'or', 'Muslims', 'a', 'b', 'The',
'historical', 'development', 'of', 'Hindu', 'self-
identity', 'within', 'the', 'local', 'South', ...]
```

Word	Count
the	**35**
and	**16**
of	**14**
to	**12**
in	**12**
Hindu	11
or	**6**
Hindus	6
century	6
term	6

Stopwords

- Words without contributions to our task.



Content words vs Function words

Function words ? Content words

Prepositions:
of, at, in, without, between
Pronouns:
he, they, anybody, it, one
Determiners:
the, a, that, my, more, much, either, neither
Auxiliary:
will, have, would, can
Particles:
as

Light Verbs:
Do, make, have, get
Conjunctions:
if, because, but, however, and...
Negatives:
no, not, neither, nor...

Nouns:
land, sea, bank, coach
Proper Nouns:
Taiwan, Pacific, English
Verbs:
write, listen, hold, run
Adjectives:
better, black, sad, fast
Adverbs:
smoothly, significantly, fast, second
Number:
second, ten, 2018

Stopword List

```
import nltk

from nltk.corpus import stopwords

nltk.download('stopwords')

stopword_list = stopwords.words('english')

print(stopword_list)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd",
'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers',
'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which',
'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been',
'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if',
'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',
'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why',
'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not',
'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't",
'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn',
"couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't",
'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't",
'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Removal of Stopwords

```
def remove_stopwords(tokens):

    tokens_clean = []

    for tok in tokens:

        if tok not in stopword_list:

            tokens_clean.append(tok)

    return tokens_clean

print(remove_stopwords(tokens))
```

```
['Hindu', 'refers', 'person', 'regards', 'culturally', 'ethnically', 'religiously', 'adhering', 'aspects',
'Hinduism', 'It', 'historically', 'used', 'geographical', 'cultural', 'later', 'religious', 'identifier',
'people', 'indigenous', 'Indian', 'subcontinent', 'The', 'historical', 'meaning', 'term', 'Hindu', 'evolved',
'time', 'Starting', 'Persian', 'Greek', 'references', 'land', 'Indus', '1st', 'millennium', 'BCE', 'texts',
'medieval', 'era', 'term', 'Hindu', 'implied', 'geographic', 'ethnic', 'cultural', 'identifier', 'people',
'living', 'Indian', 'subcontinent', 'around', 'beyond', 'Sindhu', 'Indus', 'river', 'By', '16th', 'century',
'term', 'began', 'refer', 'residents', 'subcontinent', 'Turkic', 'Muslims', 'b', 'The', 'historical',
'development', 'Hindu', 'self-identity', 'within', 'local', 'South', 'Asian', 'population', 'religious',
'cultural', 'sense', 'unclear', 'Competing', 'theories', 'state', 'Hindu', 'identity', 'developed', 'British',
'colonial', 'era', 'developed', 'post-8th', 'century', 'CE', 'Islamic', 'invasion', 'medieval', 'Hindu-Muslim',
'wars',
```

Capitalization in English

- In NLP and text mining, it is usually to convert all letter to lowercase.

- However, some information would be lost.

- In Python, it is very easy to case conversion.

  - str.lower(): Return a copy of the string with all the cased characters converted to lowercase.

  - str.upper(): Return a copy of the string with all the cased characters converted to uppercase.

Converting All Characters to Lowercase

```
def lowercase(tokens):
    tokens_lower = []
    for tok in tokens:
        tokens_lower.append(tok.lower())
    return tokens_lower
```

```
print(remove_stopwords(lowercase(tokens)))
```

```
['hindu', 'refers', 'person', 'regards', 'culturally', 'ethnically', 'religiously', 'adhering', 'aspects',
'hinduism', 'historically', 'used', 'geographical', 'cultural', 'later', 'religious', 'identifier', 'people',
'indigenous', 'indian', 'subcontinent', 'historical', 'meaning', 'term', 'hindu', 'evolved', 'time',
'starting', 'persian', 'greek', 'references', 'land', 'indus', '1st', 'millennium', 'bce', 'texts', 'medieval',
'era', 'term', 'hindu', 'implied', 'geographic', 'ethnic', 'cultural', 'identifier', 'people', 'living',
'indian', 'subcontinent', 'around', 'beyond', 'sindhu', 'indus', 'river', '16th', 'century', 'term', 'began',
'refer', 'residents', 'subcontinent', 'turkic', 'muslims', 'b', 'historical', 'development', 'hindu', 'self-
identity', 'within', 'local', 'south', 'asian', 'population', 'religious', 'cultural', 'sense', 'unclear',
'competing', 'theories', 'state', 'hindu', 'identity', 'developed', 'british', 'colonial', 'era', 'developed',
'post-8th', 'century', 'ce', 'islamic', 'invasion', 'medieval', 'hindu-muslim', 'wars', ...]
```

Removal of Stopwords with Capitalization Handling

```
def remove_stopwords(tokens):
    tokens_clean = []
    for tok in tokens:
        if tok.lower() not in stopword_list:
            tokens_clean.append(tok)
    return tokens_clean

print(remove_stopwords(tokens))
```

```
['Hindu', 'refers', 'person', 'regards', 'culturally', 'ethnically', 'religiously', 'adhering', 'aspects',
'Hinduism', 'historically', 'used', 'geographical', 'cultural', 'later', 'religious', 'identifier', 'people',
'indigenous', 'Indian', 'subcontinent', 'historical', 'meaning', 'term', 'Hindu', 'evolved', 'time',
'Starting', 'Persian', 'Greek', 'references', 'land', 'Indus', '1st', 'millennium', 'BCE', 'texts', 'medieval',
'era', 'term', 'Hindu', 'implied', 'geographic', 'ethnic', 'cultural', 'identifier', 'people', 'living',
'Indian', 'subcontinent', 'around', 'beyond', 'Sindhu', 'Indus', 'river', '16th', 'century', 'term', 'began',
'refer', 'residents', 'subcontinent', 'Turkic', 'Muslims', 'b', 'historical', 'development', 'Hindu', 'self-
identity', 'within', 'local', 'South', 'Asian', 'population', 'religious', 'cultural', 'sense', 'unclear',
'Competing', 'theories', 'state', 'Hindu', 'identity', 'developed', 'British', 'colonial', 'era', 'developed',
'post-8th', 'century', 'CE', 'Islamic', 'invasion', 'medieval', 'Hindu-Muslim', 'wars', ...]
```

Counting Result with Stopword Removal

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally',
'ethnically', 'or', 'religiously', 'adhering', 'to',
'aspects', 'of', 'Hinduism', 'It', 'has',
'historically', 'been', 'used', 'as', 'a',
'geographical', 'cultural', 'and', 'later',
'religious', 'identifier', 'for', 'people',
'indigenous', 'to', 'the', 'Indian', 'subcontinent',
'The', 'historical', 'meaning', 'of', 'the', 'term',
'Hindu', 'has', 'evolved', 'with', 'time',
'Starting', 'with', 'the', 'Persian', 'and',
'Greek', 'references', 'to', 'the', 'land', 'of',
'the', 'Indus', 'in', 'the', '1st', 'millennium',
'BCE', 'through', 'the', 'texts', 'of', 'the',
'medieval', 'era', 'the', 'term', 'Hindu',
'implied', 'a', 'geographic', 'ethnic', 'or',
'cultural', 'identifier', 'for', 'people', 'living',
'in', 'the', 'Indian', 'subcontinent', 'around',
'or', 'beyond', 'the', 'Sindhu', 'Indus', 'river',
'By', 'the', '16th', 'century', 'the', 'term',
'began', 'to', 'refer', 'to', 'residents', 'of',
'the', 'subcontinent', 'who', 'were', 'not',
'Turkic', 'or', 'Muslims', 'a', 'b', 'The',
'historical', 'development', 'of', 'Hindu', 'self-
identity', 'within', 'the', 'local', 'South', ...]
```

Word	Count
Hindu	11
Hindus	6
century	6
term	6
Indian	4
used	3
India	3
cultural	3
texts	3
religious	3

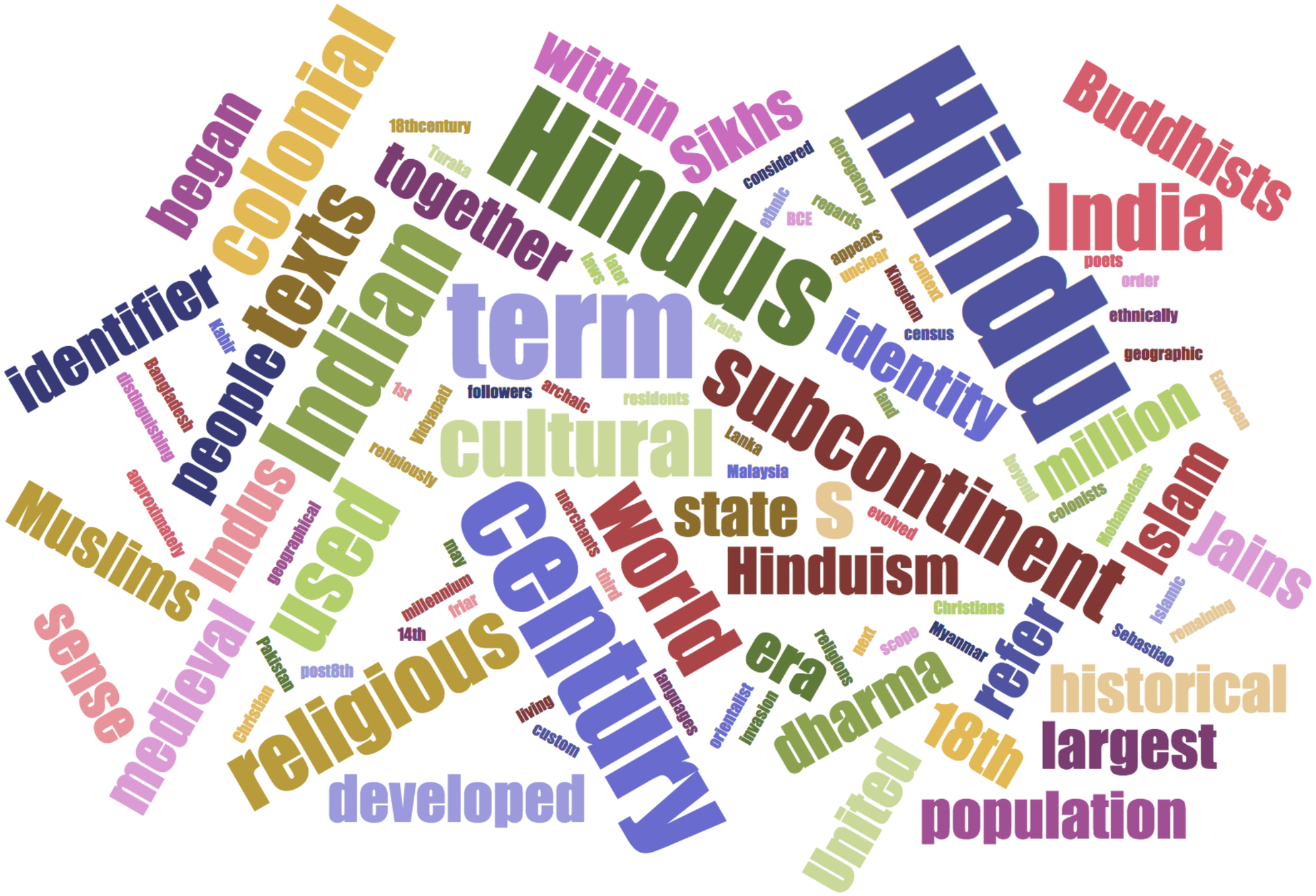
Unicase Result with Stopword Removal

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally',
'ethnically', 'or', 'religiously', 'adhering', 'to',
'aspects', 'of', 'Hinduism', 'It', 'has',
'historically', 'been', 'used', 'as', 'a',
'geographical', 'cultural', 'and', 'later',
'religious', 'identifier', 'for', 'people',
'indigenous', 'to', 'the', 'Indian', 'subcontinent',
'The', 'historical', 'meaning', 'of', 'the', 'term',
'Hindu', 'has', 'evolved', 'with', 'time',
'Starting', 'with', 'the', 'Persian', 'and',
'Greek', 'references', 'to', 'the', 'land', 'of',
'the', 'Indus', 'in', 'the', '1st', 'millennium',
'BCE', 'through', 'the', 'texts', 'of', 'the',
'medieval', 'era', 'the', 'term', 'Hindu',
'implied', 'a', 'geographic', 'ethnic', 'or',
'cultural', 'identifier', 'for', 'people', 'living',
'in', 'the', 'Indian', 'subcontinent', 'around',
'or', 'beyond', 'the', 'Sindhu', 'Indus', 'river',
'By', 'the', '16th', 'century', 'the', 'term',
'began', 'to', 'refer', 'to', 'residents', 'of',
'the', 'subcontinent', 'who', 'were', 'not',
'Turkic', 'or', 'Muslims', 'a', 'b', 'The',
'historical', 'development', 'of', 'Hindu', 'self-
identity', 'within', 'the', 'local', 'South', ...]
```

Word	Count
hindu	11
century	6
term	6
hindus	6
indian	4
used	3
cultural	3
texts	3
religious	3
colonial	3

What is the difference in the two results?

Extracted Terms



https://www.jasondavies.com/wordcloud/

Stemming（詞幹提取）

- Not to distinguish the morphological affixes from words
- Lookup table
  - There is a dictionary for looking up the stem for a given word.
  - Out-of-vocabulary issue.
  - Precise.
- Rule-base
  - If the word ends in 'ed', remove the 'ed'
  - If the word ends in 'ing', remove the 'ing'
  - If the word ends in 'ly', remove the 'ly'
  - Prone to exceptional cases

Stemming with NLTK

```
from nltk.stem.snowball import SnowballStemmer

snowball_stemmer = SnowballStemmer("english")

print(snowball_stemmer.stem('opened'))
```

open

Stemming Results

Input	Stemmed
open	open
opens	open
opened	open
opening	open
unopened	unopen
talk	talk
talks	talk
talked	talk
talking	talk
decompose	decompos
decomposes	decompos
decomposed	decompos
decomposing	decompos

Input	Stemmed
do	do
does	doe
did	did
wrote	wrote
written	written
ran	ran
gave	gave
held	held
went	went
gone	gone
lied	lie
lies	lie
lay	lay
lain	lain
lying	lie

More Stemming Results

Input	Stemmed
cats	cat
people	peopl
feet	feet
smoothly	smooth
firstly	first
secondly	second
install	instal
installed	instal
uninstall	uninstal

Input	Stemmed
internalization	intern
internationalization	internation
decontextualization	decontextu
decontextualized	decontextu
decentralization	decentr
decentralized	decentr

Stemming All Tokens

```
from nltk.stem.snowball import SnowballStemmer

snowball_stemmer = SnowballStemmer("english")
```

```
stemmed_tokens = []

for tok in remove_stopwords(tokens):

    stemmed_tokens.append(stemmer.stem(tok))

word_count = Counter(stemmed_tokens)
```

Counting Result with Stemming

```
['Hindu', 'refers', 'to', 'any', 'person', 'who',
'regards', 'themselves', 'as', 'culturally',
'ethnically', 'or', 'religiously', 'adhering', 'to',
'aspects', 'of', 'Hinduism', 'It', 'has',
'historically', 'been', 'used', 'as', 'a',
'geographical', 'cultural', 'and', 'later',
'religious', 'identifier', 'for', 'people',
'indigenous', 'to', 'the', 'Indian', 'subcontinent',
'The', 'historical', 'meaning', 'of', 'the', 'term',
'Hindu', 'has', 'evolved', 'with', 'time',
'Starting', 'with', 'the', 'Persian', 'and',
'Greek', 'references', 'to', 'the', 'land', 'of',
'the', 'Indus', 'in', 'the', '1st', 'millennium',
'BCE', 'through', 'the', 'texts', 'of', 'the',
'medieval', 'era', 'the', 'term', 'Hindu',
'implied', 'a', 'geographic', 'ethnic', 'or',
'cultural', 'identifier', 'for', 'people', 'living',
'in', 'the', 'Indian', 'subcontinent', 'around',
'or', 'beyond', 'the', 'Sindhu', 'Indus', 'river',
'By', 'the', '16th', 'century', 'the', 'term',
'began', 'to', 'refer', 'to', 'residents', 'of',
'the', 'subcontinent', 'who', 'were', 'not',
'Turkic', 'or', 'Muslims', 'a', 'b', 'The',
'historical', 'development', 'of', 'Hindu', 'self-
identity', 'within', 'the', 'local', 'South', ...]
```

Word	Count
hindu	12
term	6
hindus	6
centuri	6
religi	4
indian	4
use	4
refer	4
cultur	4
text	3
coloni	3
world	3
state	3
popul	3

Lemmatization（字形還原）

- Grouping together the inflected forms of a word as a single lemma.
- A process more complex than stemming.
- Most based on dictionary.
- The outcome is more readable.

Lemmatization with NLTK

```
from nltk.stem import WordNetLemmatizer

wordnet_lemmatizer = WordNetLemmatizer()

print(wordnet_lemmatizer.lemmatize('opened', pos = 'v'))
```

open

Lemmatization with NLTK Regardless of POS

```
def lemmatize(token):

    # ADJ (a), ADJ_SAT (s), ADV (r), NOUN (n) or VERB (v)

    for p in ['v', 'n', 'a', 'r', 's']:

        l = wordnet_lemmatizer.lemmatize(token, pos=p)

        if l != token:

            return l

return token
```

Stemming vs Lemmatization

Input	Stemmed	Lemmatized
unopened	unopen	unopened
decompose	decompos	decompose
decomposes	decompos	decompose
decomposed	decompos	decompose
decomposing	decompos	decompose
does	doe	do
did	did	do
wrote	wrote	write
written	written	write
ran	ran	run
gave	gave	give
held	held	hold
went	went	go
gone	gone	go
lain	lain	lie

Stemming vs Lemmatization

Input	Stemmed	Lemmatized
people	peopl	people
feet	feet	foot
women	women	woman
smoothly	smooth	smoothly
firstly	first	firstly
secondly	second	secondly
install	instal	install
uninstall	uninstal	uninstall
internalization	intern	internalization
internationalization	internation	internationalization
decontextualization	decontextu	decontextualization
decontextualized	decontextu	decontextualized
decentralization	decentr	decentralization
decentralized	decentr	decentralize

Stemming/Lemmatization is Sensitive to Final Task

- Important information can be lost during stemming/ lemmatization.
    - Tense
- To do or not to do
    - It depends on your final task.

Results

Hindu refers to any person who regards themselves as culturally, ethnically, or religiously adhering to aspects of Hinduism.[1][2] It has historically been used as a geographical, cultural, and later religious identifier for people indigenous to the Indian subcontinent.[3][4] The historical meaning of the term Hindu has evolved with time. Starting with the Persian and Greek references to the land of the Indus in the 1st millennium BCE through the texts of the medieval era,[5] the term Hindu implied a geographic, ethnic or cultural identifier for people living in the Indian subcontinent around or beyond the Sindhu (Indus) river.[6] .....

Word	Count
Hindu	11
term	6
century	6
Hindus	6
use	4
Indian	4
text	3
subcontinent	3
India	3
cultural	3
world	3
religious	3
colonial	3
refer	3
population	3

Visualization

Assignment (Due: 2018/10/31 23:59)

- Answer the Quiz in Slide #7
  - Point out 2 problems in the code.
  - Provide a revised version (bonus)
- Generate a top word list and a word cloud with the data in corpus.txt.
  - Feel free to modify my procedure for a better result.