Reg. No. 19MAI0040 Name: Sumit Tile

Digital Assignment

Textacy

Textacy is a Python library for performing a variety of natural language processing (NLP) tasks, built on the high-performance spaCy library. With the fundamentals tokenization, part-of-speech tagging, dependency parsing, etc. Delegated to another library, textacy focuses primarily on the tasks that come before and follow after.

Features

1. Convenient entry points to working with one or many documents processed by spaCy, with functionality added via custom extensions and automatic language identification for applying the right spaCy pipeline
2. Variety of downloadable datasets with both text content and metadata, from Congressional speeches to historical literature to Reddit comments
3. Easy file I/O for streaming data to and from disk
4. Cleaning, normalization, and exploration of raw text — before processing
5. Flexible extraction of words, ngrams, noun chunks, entities, acronyms, key terms, and other elements of interest
6. Tokenization and vectorization of documents, with functionality for training, interpreting, and visualizing topic models
7. String, set, and document similarity comparison by a variety of metrics
8. Calculations for common text statistics, including Flesch-Kincaid Grade Level and multilingual Flesch Reading Ease

---

First step in using textacy is to install it. The commands to install it are:

pip install textacy

conda install -c conda-forge textacy

The textacy package contains all top-level functionalities for which it is made

In [48]:

```
import textacy
#import textacy.datasets
```

In [49]:

```
myfile = open("task.txt")            #...Importing an text file
text = myfile.read()
text
```

Out[49]:

```
'Four score and seven years ago our fathers brought forth on this continent, a new nation,
conceived in Liberty, and dedicated to the proposition that all men are created equal.\nNow we are
engaged in a great civil war, testing whether that nation, or any nation so conceived and so
dedicated, can long endure. \nWe are met on a great battle-field of that war. We have come to dedi
cate a portion of that field, \nas a final resting place for those who here gave their lives that
that nation might live. It is altogether fitting and proper that we should do this.\n\nBut, in a l
arger sense, we can not dedicateâ€”we can not consecrateâ€”we can not hallowâ€”this ground. The br
ave men, living and dead, who struggled here, have consecrated it, \nfar above our poor power to a
dd or detract. \nThe world will little note, nor long remember what we say here, but it can never
forget what they did here. It is for us the living, rather, \nto be dedicated here to the
unfinished work which they who fought here have thus far so nobly advanced. \nIt is rather for us
to be here dedicated to the great task remaining before usâ€”that from these honored dead we take
increased devotion to \nthat cause for which they gave the last full measure of devotionâ€”that we
here highly resolve that these dead shall not have died in vainâ€”that this nation, under God, \ns
hall have a new birth of freedomâ€”and that government of the people, by the people, for the peopl
e, shall not perish from the earth.'
```

To clean messy raw text input, textacy provides an package preprocessing. It cleans the data with some funtions like whitespace, quotation marks, etc.

In [50]:

```
from textacy import preprocessing
preprocessing.normalize_whitespace(preprocessing.remove_punctuation(text))[:]
```

Out[50]:

```
'Four score and seven years ago our fathers brought forth on this continent a new nation conceived
in Liberty and dedicated to the proposition that all men are created equal \nNow we are engaged in
a great civil war testing whether that nation or any nation so conceived and so dedicated can long
endure \nWe are met on a great battle field of that war We have come to dedicate a portion of that
field \nas a final resting place for those who here gave their lives that that nation might live I
t is altogether fitting and proper that we should do this \nBut in a larger sense we can not
dedicateâ€ we can not consecrateâ€ we can not hallowâ€ this ground The brave men living and dead w
ho struggled here have consecrated it \nfar above our poor power to add or detract \nThe world
will little note nor long remember what we say here but it can never forget what they did here It
is for us the living rather \nto be dedicated here to the unfinished work which they who fought he
re have thus far so nobly advanced \nIt is rather for us to be here dedicated to the great task re
maining before usâ€ that from these honored dead we take increased devotion to \nthat cause for wh
ich they gave the last full measure of devotionâ€ that we here highly resolve that these dead
shall not have died in vainâ€ that this nation under God \nshall have a new birth of freedomâ€ and
that government of the people by the people for the people shall not perish from the earth'
```

Textacy dataset is package which contains speeches of some top leaders

In [51]:

```
import textacy.datasets  # note the import
ds = textacy.datasets.CapitolWords()
ds.download()
records = ds.records(speaker_name={"Hillary Clinton", "Barack Obama"})
next(records)
```

Out[51]:

```
('I yield myself 15 minutes of the time controlled by the Democrats.',
 {'date': '2001-02-13',
  'congress': 107,
  'speaker_name': 'Hillary Clinton',
  'speaker_party': 'D',
  'title': 'MORNING BUSINESS',
  'chamber': 'Senate'})
```

One of the feature of textacy is getting data into document format. Lets convert text data into an document

In [52]:

```
en = textacy.load_spacy_lang("en_core_web_sm", disable=("parser",))
doc = textacy.make_spacy_doc(text, lang=en)
doc._.preview   #...shows document preview with total number of tokens
```

Out[52]:

```
'Doc(308 tokens: "Four score and seven years ago our fathers brou...")'
```

For analyzing document textacy gives various ways, those are as follows

In [53]:

```
# Trigrams
list(textacy.extract.ngrams( doc, 3, filter_stops=True, filter_punct=True, filter_nums=False))
```

Out[53]:

```
[score and seven,
 seven years ago,
 ago our fathers,
 fathers brought forth,
 conceived in Liberty,
 men are created,
 great civil war,
 nation so conceived,
 come to dedicate,
 dedicate a portion,
 final resting place,
 gave their lives,
 nation might live,
 fitting and proper,
 living and dead,
 power to add,
 add or detract,
 world will little,
 far so nobly,
 great task remaining,
 remaining before usâ€”that,
 measure of devotionâ€”that,
 died in vainâ€”that,
 vainâ€”that this nation,
 birth of freedomâ€”and,
 freedomâ€”and that government,
 shall not perish]
```

In [54]:

```
# Bigrams
list(textacy.extract.ngrams(doc, 2))
```

Out[54]:

```
[seven years,
 years ago,
 fathers brought,
 brought forth,
 new nation,
 created equal,
 great civil,
 civil war,
 long endure,
 great battle,
 final resting,
 resting place,
 altogether fitting,
 larger sense,
 hallowâ€”this ground,
 brave men,
 poor power,
 little note,
 long remember,
 unfinished work,
 nobly advanced,
 great task,
 task remaining,
 honored dead,
 increased devotion,
 highly resolve,
 dead shall,
 new birth]
```

In [55]:

```
# Bigrams having frequency of occurance more than one is showing
```

Out[55]:

```
[]
```

Entity extraction shows entities in the text provided which can be based on pattern matching, linguistics, syntax, semantics, or combination of these.

In [56]:

```
list(textacy.extract.entities(doc, drop_determiners=True))
```

Out[56]:

```
[Four, seven years ago, Liberty]
```

POS tagging

In [57]:

```
# Defining pattern for pos tagging
pattern = textacy.constants.POS_REGEX_PATTERNS["en"]["NP"]
pattern
```

Out[57]:

```
'<DET>? <NUM>* (<ADJ> <PUNCT>? <CONJ>?)* (<NOUN>|<PROPN> <PART>?)+'
```

In [58]:

```
list(textacy.extract.pos_regex_matches(doc, pattern))
```

```
C:\Users\Sumit\Anaconda3\lib\site-packages\textacy\extract.py:332: DeprecationWarning:
`pos_regex_matches()` has been deprecated! for similar but more powerful and performant
functionality, use `textacy.extract.matches()` instead.
  action="once",
```

Out[58]:

```
[Four score,
 seven years,
 our fathers,
 this continent,
 a new nation,
 Liberty,
 the proposition,
 all men,
 a great civil war,
 that nation,
 any nation,
 a great battle,
 field,
 that war,
 a portion,
 that field,
 place,
 their lives,
 that nation,
 a larger sense,
 consecrateâ€”we,
 ground,
 The brave men,
 living,
 our poor power,
```

```
 the unfinished work,
 the great task,
 usâ€”that,
 devotion,
 that cause,
 the last full measure,
 these dead,
 vainâ€”that,
 this nation,
 God,
 a new birth,
 freedomâ€”and,
 that government,
 the people,
 the people,
 the people,
 the earth]
```

Getting key terms from the given text

In [59]:

```
import textacy.ke      #...this package includes many algorithms to find out key terms
textacy.ke.textrank(doc, topn=10)      #...Topn is used to get the number of top words
```

Out[59]:

```
[('great civil war', 0.021696862470056172),
 ('new nation', 0.018172169732000733),
 ('great task', 0.015419500236084066),
 ('great battle', 0.014956691963640079),
 ('brave man', 0.011442764212560144),
 ('new birth', 0.010170824703844948),
 ('dead', 0.010052778364395897),
 ('unfinished work', 0.009197000477015455),
 ('large sense', 0.008733454058238664),
 ('poor power', 0.00863586069445221)]
```

In [60]:

```
# To count the unique words in text
ts = textacy.TextStats(doc)
ts.n_unique_words
```

Out[60]:

```
141
```

In [61]:

```
# It gives some information about text such as count
ts.basic_counts
```

Out[61]:

```
{'n_sents': None,
 'n_words': 265,
 'n_chars': 1170,
 'n_syllables': 351,
 'n_unique_words': 141,
 'n_long_words': 40,
 'n_monosyllable_words': 204,
 'n_polysyllable_words': 19}
```

In [62]:

```
#..Bot gives bagofterm function which consists too many operations which can be used in various wa
```

```
#..Here we are taking frequency words in text
sorted(bot.items(), key=lambda x: x[1], reverse=True)[:15]
```

Out[62]:

```
[('nation', 5),
 ('dedicate', 4),
 ('great', 3),
 ('dead', 3),
 ('shall', 3),
 ('people', 3),
 ('new', 2),
 ('conceive', 2),
 ('man', 2),
 ('war', 2),
 ('long', 2),
 ('field', 2),
 ('give', 2),
 ('living', 2),
 ('far', 2)]
```

Now usnig spacy and textacy together we can take advantage of both

In [63]:

```
import spacy # Importing spacy packages
nlp = spacy.load('en_core_web_sm')
```

In [64]:

```
docx_spacy = nlp(text) #Converting raw text into document
```

In [65]:

```
docx_spacy
```

Out[65]:

```
Four score and seven years ago our fathers brought forth on this continent, a new nation,
conceived in Liberty, and dedicated to the proposition that all men are created equal.
Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived a
nd so dedicated, can long endure.
We are met on a great battle-field of that war. We have come to dedicate a portion of that field,
as a final resting place for those who here gave their lives that that nation might live. It is
altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicateâ€”we can not consecrateâ€”we can not hallowâ€”this gro
und. The brave men, living and dead, who struggled here, have consecrated it,
far above our poor power to add or detract.
The world will little note, nor long remember what we say here, but it can never forget what they
did here. It is for us the living, rather,
to be dedicated here to the unfinished work which they who fought here have thus far so nobly adva
nced.
It is rather for us to be here dedicated to the great task remaining before usâ€”that from these h
onored dead we take increased devotion to
that cause for which they gave the last full measure of devotionâ€”that we here highly resolve tha
t these dead shall not have died in vainâ€”that this nation, under God,
shall have a new birth of freedomâ€”and that government of the people, by the people, for the peop
le, shall not perish from the earth.
```

In [66]:

```
type(docx_spacy)  # checking type to confirm type of document
```

Out[66]:

```
spacy.tokens.doc.Doc
```

In [67]:

```
# Using SpaCy Named Entities Recognition
[ (entity.text,entity.label_) for entity in docx_spacy.ents ]
```

Out[67]:

```
[('Four', 'CARDINAL'), ('seven years ago', 'DATE'), ('Liberty', 'GPE')]
```

In [68]:

```
# ngrams with textacy from the document created by using spacy

list(textacy.extract.ngrams(docx_spacy,3))
```

Out[68]:

```
[score and seven,
 seven years ago,
 ago our fathers,
 fathers brought forth,
 conceived in Liberty,
 men are created,
 great civil war,
 nation so conceived,
 come to dedicate,
 dedicate a portion,
 final resting place,
 gave their lives,
 nation might live,
 fitting and proper,
 living and dead,
 power to add,
 add or detract,
 world will little,
 far so nobly,
 great task remaining,
 remaining before usâ€”that,
 measure of devotionâ€”that,
 died in vainâ€”that,
 vainâ€”that this nation,
 birth of freedomâ€”and,
 freedomâ€”and that government,
 shall not perish]
```

In [70]:

```
# Tokenizing the given document
mylemma = [(token.lemma_) for token in docx_spacy]
```

In [71]:

```
mylemma
```

Out[71]:

```
['four',
 'score',
 'and',
 'seven',
 'year',
 'ago',
 '-PRON-',
 'father',
 'bring',
 'forth',
 'on',
 'this',
```

```
'a',
'new',
'nation',
',',
'conceive',
'in',
'Liberty',
',',
'and',
'dedicate',
'to',
'the',
'proposition',
'that',
'all',
'man',
'be',
'create',
'equal',
'.',
'\n',
'now',
'-PRON-',
'be',
'engage',
'in',
'a',
'great',
'civil',
'war',
',',
'test',
'whether',
'that',
'nation',
',',
'or',
'any',
'nation',
'so',
'conceive',
'and',
'so',
'dedicated',
',',
'can',
'long',
'endure',
'.',
'\n',
'-PRON-',
'be',
'meet',
'on',
'a',
'great',
'battle',
'-',
'field',
'of',
'that',
'war',
'.',
'-PRON-',
'have',
'come',
'to',
'dedicate',
'a',
'portion',
'of',
'that',
'field',
',',
'\n',
```

```
'final',
'rest',
'place',
'for',
'those',
'who',
'here',
'give',
'-PRON-',
'life',
'that',
'that',
'nation',
'may',
'live',
'.',
'-PRON-',
'be',
'altogether',
'fitting',
'and',
'proper',
'that',
'-PRON-',
'should',
'do',
'this',
'.',
'\n\n',
'but',
',',
'in',
'a',
'large',
'sense',
',',
'-PRON-',
'can',
'not',
'dedicateâ€”we',
'can',
'not',
'consecrateâ€”we',
'can',
'not',
'hallowâ€”this',
'ground',
'.',
'the',
'brave',
'man',
',',
'living',
'and',
'dead',
',',
'who',
'struggle',
'here',
',',
'have',
'consecrate',
'-PRON-',
',',
'\n',
'far',
'above',
'-PRON-',
'poor',
'power',
'to',
'add',
'or',
'detract',
'.',
```

```
'world',
'will',
'little',
'note',
',',
'nor',
'long',
'remember',
'what',
'-PRON-',
'say',
'here',
',',
'but',
'-PRON-',
'can',
'never',
'forget',
'what',
'-PRON-',
'do',
'here',
'.',
'-PRON-',
'be',
'for',
'-PRON-',
'the',
'living',
',',
'rather',
',',
'\n',
'to',
'be',
'dedicate',
'here',
'to',
'the',
'unfinished',
'work',
'which',
'-PRON-',
'who',
'fight',
'here',
'have',
'thus',
'far',
'so',
'nobly',
'advanced',
'.',
'\n',
'-PRON-',
'be',
'rather',
'for',
'-PRON-',
'to',
'be',
'here',
'dedicate',
'to',
'the',
'great',
'task',
'remain',
'before',
'usâ€”that',
'from',
'these',
'honor',
'dead',
'-PRON-',
```

```
'devotion',
'to',
'\n',
'that',
'cause',
'for',
'which',
'-PRON-',
'give',
'the',
'last',
'full',
'measure',
'of',
'devotionâ€”that',
'-PRON-',
'here',
'highly',
'resolve',
'that',
'these',
'dead',
'shall',
'not',
'have',
'die',
'in',
'vainâ€”that',
'this',
'nation',
',',
'under',
'God',
',',
'\n',
'shall',
'have',
'a',
'new',
'birth',
'of',
'freedomâ€”and',
'that',
'government',
'of',
'the',
'people',
',',
'by',
'the',
'people',
',',
'for',
'the',
'people',
',',
'shall',
'not',
'perish',
'from',
'the',
'earth',
'.']
```

References

[1] DeWilde, Burton. "textacy Documentation."(2017)

[2] https://pypi.org/project/textacy/0.3.1/

[3] https://jcharistech.wordpress.com/2018/11/28/natural-language-processing-with-textacy spacy/#:~:text=With%20the%20basics%20%E2%80%94%20tokenization%2C%20part,quotation%20attribution%2C%20and%20more%

Note: The research paper on Textacy is not available either on IEEE, Springer or Google Scholar.

In [ ]: