02_nlp_with_textblob

September 29, 2021

1 NLP with TextBlob

TextBlob is a python library that provides a simple API for common NLP tasks and builds on the Natural Language Toolkit (nltk) and the Pattern web mining libraries. TextBlob facilitates part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and others.

1.1 Imports & Settings

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
%matplotlib inline
from pathlib import Path

import numpy as np
import pandas as pd

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# spacy, textblob and nltk for language processing
from textblob import TextBlob, Word
import nltk
from nltk.stem.snowball import SnowballStemmer

# sklearn for feature extraction & modeling
from sklearn.feature_extraction.text import CountVectorizer
```

[ ]:
```
# download NLTK resources
nltk.download('punkt')
```

[3]:
```
sns.set_style('white')
np.random.seed(42)
```

1.2 Load BBC Data

To illustrate the use of TextBlob, we sample a BBC sports article with the headline 'Robinson ready for difficult task'. Similar to spaCy and other libraries, the first step is to pass the document through a pipeline represented by the TextBlob object to assign annotations required for various tasks.

[4]:
```
path = Path('..', 'data', 'bbc')
files = sorted(list(path.glob('**/*.txt')))
doc_list = []
for i, file in enumerate(files):
    topic = file.parts[-2]
    article = file.read_text(encoding='latin1').split('\n')
    heading = article[0].strip()
    body = ' '.join([l.strip() for l in article[1:]]).strip()
    doc_list.append([topic, heading, body])
```

[5]:
```
docs = pd.DataFrame(doc_list, columns=['topic', 'heading', 'body'])
docs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2225 entries, 0 to 2224
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   topic    2225 non-null   object
 1   heading  2225 non-null   object
 2   body     2225 non-null   object
dtypes: object(3)
memory usage: 52.3+ KB
```

1.3 Introduction to TextBlob

You should already have downloaded TextBlob, a Python library used to explore common NLP tasks.

1.3.1 Select random article

[6]:
```
article = docs.sample(1).squeeze()
```

[7]:
```
print(f'Topic:\t{article.topic.capitalize()}\n\n{article.heading}\n')
print(article.body.strip())
```

```
Topic:  Business

UK house prices dip in November

UK house prices dipped slightly in November, the Office of the Deputy Prime
Minister (ODPM) has said.  The average house price fell marginally to Â£180,226,
```

from Â£180,444 in October. Recent evidence has suggested that the UK housing market is slowing after interest rate increases, and economists forecast a drop in prices during 2005. But while the monthly figures may hint at a cooling of the market, annual house price inflation is still strong, up 13.8% in the year to November. Economists, however, forecast that ODPM figures are likely to show a weakening in annual house price growth in coming months. "Overall, the housing market activity is slowing down and that is backed up by the mortgage lending and the mortgage approvals data," said Mark Miller, at HBOS Treasury Services. "The ODPM data is a fairly lagging indicator." The figures come after the Bank of England said the number of mortgages approved in the UK has fallen to the lowest level for nearly a decade. The Halifax, meanwhile, said last week that house prices increased by 1.1% in December - the first monthly rise since September. The UK's biggest mortgage lender said prices rose 15.1% over the whole of 2004, but by only 2.8% in the second half of the year. It is predicting a 2% fall in overall prices in 2005 as the market stabilises after large gains in recent years. The ODPM attributed the monthly fall of prices in November to a drop in the value of detached houses and flats. It said annual inflation rose between October and November because prices had fallen by 1.1% in the same period in 2003. The ODPM data showed the average house price was Â£192,713 in England; Â£139,544 in Wales; Â£116,542 in Scotland, and Â£111,314 in Northern Ireland. All areas saw a rise in annual house price inflation in November except for Northern Ireland and the West Midlands, where the rate was unchanged, the ODPM said. The North East showed the highest rate of inflation at 26.2%, followed by Yorkshire and the Humber on 21.7%, and the North West on 21.1%. The East Midlands, the West Midlands and the South West all had an annual inflation rate of more than 15%. In London, the area with the highest average house price at Â£262,825, annual inflation rose only slightly in November to 7.1% from 7% the previous month.

[8]:
```
parsed_body = TextBlob(article.body)
```

1.3.2 Tokenization

[9]:
```
parsed_body.words
```

[9]:
```
WordList(['UK', 'house', 'prices', 'dipped', 'slightly', 'in', 'November',
'the', 'Office', 'of', 'the', 'Deputy', 'Prime', 'Minister', 'ODPM', 'has',
'said', 'The', 'average', 'house', 'price', 'fell', 'marginally', 'to',
'Â£180,226', 'from', 'Â£180,444', 'in', 'October', 'Recent', 'evidence', 'has',
'suggested', 'that', 'the', 'UK', 'housing', 'market', 'is', 'slowing', 'after',
'interest', 'rate', 'increases', 'and', 'economists', 'forecast', 'a', 'drop',
'in', 'prices', 'during', '2005', 'But', 'while', 'the', 'monthly', 'figures',
'may', 'hint', 'at', 'a', 'cooling', 'of', 'the', 'market', 'annual', 'house',
'price', 'inflation', 'is', 'still', 'strong', 'up', '13.8', 'in', 'the',
'year', 'to', 'November', 'Economists', 'however', 'forecast', 'that', 'ODPM',
'figures', 'are', 'likely', 'to', 'show', 'a', 'weakening', 'in', 'annual',
'house', 'price', 'growth', 'in', 'coming', 'months', 'Overall', 'the',
```

```
'housing', 'market', 'activity', 'is', 'slowing', 'down', 'and', 'that', 'is',
'backed', 'up', 'by', 'the', 'mortgage', 'lending', 'and', 'the', 'mortgage',
'approvals', 'data', 'said', 'Mark', 'Miller', 'at', 'HBOS', 'Treasury',
'Services', 'The', 'ODPM', 'data', 'is', 'a', 'fairly', 'lagging', 'indicator',
'The', 'figures', 'come', 'after', 'the', 'Bank', 'of', 'England', 'said',
'the', 'number', 'of', 'mortgages', 'approved', 'in', 'the', 'UK', 'has',
'fallen', 'to', 'the', 'lowest', 'level', 'for', 'nearly', 'a', 'decade', 'The',
'Halifax', 'meanwhile', 'said', 'last', 'week', 'that', 'house', 'prices',
'increased', 'by', '1.1', 'in', 'December', 'the', 'first', 'monthly', 'rise',
'since', 'September', 'The', 'UK', "'s", 'biggest', 'mortgage', 'lender',
'said', 'prices', 'rose', '15.1', 'over', 'the', 'whole', 'of', '2004', 'but',
'by', 'only', '2.8', 'in', 'the', 'second', 'half', 'of', 'the', 'year', 'It',
'is', 'predicting', 'a', '2', 'fall', 'in', 'overall', 'prices', 'in', '2005',
'as', 'the', 'market', 'stabilises', 'after', 'large', 'gains', 'in', 'recent',
'years', 'The', 'ODPM', 'attributed', 'the', 'monthly', 'fall', 'of', 'prices',
'in', 'November', 'to', 'a', 'drop', 'in', 'the', 'value', 'of', 'detached',
'houses', 'and', 'flats', 'It', 'said', 'annual', 'inflation', 'rose',
'between', 'October', 'and', 'November', 'because', 'prices', 'had', 'fallen',
'by', '1.1', 'in', 'the', 'same', 'period', 'in', '2003', 'The', 'ODPM', 'data',
'showed', 'the', 'average', 'house', 'price', 'was', 'Â£192,713', 'in',
'England', 'Â£139,544', 'in', 'Wales', 'Â£116,542', 'in', 'Scotland', 'and',
'Â£111,314', 'in', 'Northern', 'Ireland', 'All', 'areas', 'saw', 'a', 'rise',
'in', 'annual', 'house', 'price', 'inflation', 'in', 'November', 'except',
'for', 'Northern', 'Ireland', 'and', 'the', 'West', 'Midlands', 'where', 'the',
'rate', 'was', 'unchanged', 'the', 'ODPM', 'said', 'The', 'North', 'East',
'showed', 'the', 'highest', 'rate', 'of', 'inflation', 'at', '26.2', 'followed',
'by', 'Yorkshire', 'and', 'the', 'Humber', 'on', '21.7', 'and', 'the', 'North',
'West', 'on', '21.1', 'The', 'East', 'Midlands', 'the', 'West', 'Midlands',
'and', 'the', 'South', 'West', 'all', 'had', 'an', 'annual', 'inflation',
'rate', 'of', 'more', 'than', '15', 'In', 'London', 'the', 'area', 'with',
'the', 'highest', 'average', 'house', 'price', 'at', 'Â£262,825', 'annual',
'inflation', 'rose', 'only', 'slightly', 'in', 'November', 'to', '7.1', 'from',
'7', 'the', 'previous', 'month'])
```

1.3.3 Sentence boundary detection

[11]:
```
parsed_body.sentences
```

[11]:
```
[Sentence("UK house prices dipped slightly in November, the Office of the Deputy
Prime Minister (ODPM) has said."),
 Sentence("The average house price fell marginally to Â£180,226, from Â£180,444
in October."),
 Sentence("Recent evidence has suggested that the UK housing market is slowing
after interest rate increases, and economists forecast a drop in prices during
2005."),
 Sentence("But while the monthly figures may hint at a cooling of the market,
annual house price inflation is still strong, up 13.8% in the year to
```

```
November."),
 Sentence("Economists, however, forecast that ODPM figures are likely to show a
weakening in annual house price growth in coming months."),
 Sentence(""Overall, the housing market activity is slowing down and that is
backed up by the mortgage lending and the mortgage approvals data," said Mark
Miller, at HBOS Treasury Services."),
 Sentence(""The ODPM data is a fairly lagging indicator.""),
 Sentence("The figures come after the Bank of England said the number of
mortgages approved in the UK has fallen to the lowest level for nearly a
decade."),
 Sentence("The Halifax, meanwhile, said last week that house prices increased by
1.1% in December - the first monthly rise since September."),
 Sentence("The UK's biggest mortgage lender said prices rose 15.1% over the
whole of 2004, but by only 2.8% in the second half of the year."),
 Sentence("It is predicting a 2% fall in overall prices in 2005 as the market
stabilises after large gains in recent years."),
 Sentence("The ODPM attributed the monthly fall of prices in November to a drop
in the value of detached houses and flats."),
 Sentence("It said annual inflation rose between October and November because
prices had fallen by 1.1% in the same period in 2003."),
 Sentence("The ODPM data showed the average house price was Â£192,713 in
England; Â£139,544 in Wales; Â£116,542 in Scotland, and Â£111,314 in Northern
Ireland."),
 Sentence("All areas saw a rise in annual house price inflation in November
except for Northern Ireland and the West Midlands, where the rate was unchanged,
the ODPM said."),
 Sentence("The North East showed the highest rate of inflation at 26.2%,
followed by Yorkshire and the Humber on 21.7%, and the North West on 21.1%."),
 Sentence("The East Midlands, the West Midlands and the South West all had an
annual inflation rate of more than 15%."),
 Sentence("In London, the area with the highest average house price at
Â£262,825, annual inflation rose only slightly in November to 7.1% from 7% the
previous month.")]
```

1.3.4 Stemming

To perform stemming, we instantiate the SnowballStemmer from the nltk library, call its .stem() method on each token and display tokens that were modified as a result:

[12]:
```
# Initialize stemmer.
stemmer = SnowballStemmer('english')

# Stem each word.
[(word, stemmer.stem(word)) for i, word in enumerate(parsed_body.words)
 if word.lower() != stemmer.stem(parsed_body.words[i])]
```

[12]:
```
[('house', 'hous'),
 ('prices', 'price'),
 ('dipped', 'dip'),
 ('slightly', 'slight'),
 ('November', 'novemb'),
 ('Office', 'offic'),
 ('Deputy', 'deputi'),
 ('Minister', 'minist'),
 ('average', 'averag'),
 ('house', 'hous'),
 ('marginally', 'margin'),
 ('October', 'octob'),
 ('evidence', 'evid'),
 ('suggested', 'suggest'),
 ('housing', 'hous'),
 ('slowing', 'slow'),
 ('increases', 'increas'),
 ('economists', 'economist'),
 ('prices', 'price'),
 ('during', 'dure'),
 ('monthly', 'month'),
 ('figures', 'figur'),
 ('cooling', 'cool'),
 ('house', 'hous'),
 ('inflation', 'inflat'),
 ('November', 'novemb'),
 ('Economists', 'economist'),
 ('however', 'howev'),
 ('figures', 'figur'),
 ('likely', 'like'),
 ('weakening', 'weaken'),
 ('house', 'hous'),
 ('coming', 'come'),
 ('months', 'month'),
 ('Overall', 'overal'),
 ('housing', 'hous'),
 ('activity', 'activ'),
 ('slowing', 'slow'),
 ('backed', 'back'),
 ('mortgage', 'mortgag'),
 ('lending', 'lend'),
 ('mortgage', 'mortgag'),
 ('approvals', 'approv'),
 ('Treasury', 'treasuri'),
 ('Services', 'servic'),
 ('fairly', 'fair'),
 ('lagging', 'lag'),
```

```
('indicator', 'indic'),
('figures', 'figur'),
('mortgages', 'mortgag'),
('approved', 'approv'),
('nearly', 'near'),
('decade', 'decad'),
('meanwhile', 'meanwhil'),
('house', 'hous'),
('prices', 'price'),
('increased', 'increas'),
('December', 'decemb'),
('monthly', 'month'),
('since', 'sinc'),
('September', 'septemb'),
('mortgage', 'mortgag'),
('prices', 'price'),
('only', 'onli'),
('predicting', 'predict'),
('overall', 'overal'),
('prices', 'price'),
('stabilises', 'stabilis'),
('large', 'larg'),
('gains', 'gain'),
('years', 'year'),
('attributed', 'attribut'),
('monthly', 'month'),
('prices', 'price'),
('November', 'novemb'),
('value', 'valu'),
('detached', 'detach'),
('houses', 'hous'),
('flats', 'flat'),
('inflation', 'inflat'),
('October', 'octob'),
('November', 'novemb'),
('because', 'becaus'),
('prices', 'price'),
('showed', 'show'),
('average', 'averag'),
('house', 'hous'),
('Wales', 'wale'),
('areas', 'area'),
('house', 'hous'),
('inflation', 'inflat'),
('November', 'novemb'),
('Midlands', 'midland'),
('unchanged', 'unchang'),
```

```
 ('showed', 'show'),
 ('inflation', 'inflat'),
 ('followed', 'follow'),
 ('Yorkshire', 'yorkshir'),
 ('Midlands', 'midland'),
 ('Midlands', 'midland'),
 ('inflation', 'inflat'),
 ('average', 'averag'),
 ('house', 'hous'),
 ('inflation', 'inflat'),
 ('only', 'onli'),
 ('slightly', 'slight'),
 ('November', 'novemb')]
```

1.3.5 Lemmatization

[13]:
```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /home/stefan/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[13]:
```
True
```

[14]:
```
[(word, word.lemmatize()) for i, word in enumerate(parsed_body.words)
 if word != parsed_body.words[i].lemmatize()]
```

[14]:
```
[('prices', 'price'),
 ('has', 'ha'),
 ('has', 'ha'),
 ('increases', 'increase'),
 ('economists', 'economist'),
 ('prices', 'price'),
 ('figures', 'figure'),
 ('figures', 'figure'),
 ('months', 'month'),
 ('approvals', 'approval'),
 ('figures', 'figure'),
 ('mortgages', 'mortgage'),
 ('has', 'ha'),
 ('prices', 'price'),
 ('prices', 'price'),
 ('prices', 'price'),
 ('as', 'a'),
 ('gains', 'gain'),
 ('years', 'year'),
 ('prices', 'price'),
```

```
 ('houses', 'house'),
 ('flats', 'flat'),
 ('prices', 'price'),
 ('was', 'wa'),
 ('areas', 'area'),
 ('was', 'wa')]
```

Lemmatization relies on parts-of-speech (POS) tagging; `spaCy` performs POS tagging, here we make assumptions, e.g. that each token is verb.

[15]:
```
[(word, word.lemmatize(pos='v')) for i, word in enumerate(parsed_body.words)
 if word != parsed_body.words[i].lemmatize(pos='v')]
```

[15]:
```
[('prices', 'price'),
 ('dipped', 'dip'),
 ('has', 'have'),
 ('said', 'say'),
 ('has', 'have'),
 ('suggested', 'suggest'),
 ('housing', 'house'),
 ('is', 'be'),
 ('slowing', 'slow'),
 ('increases', 'increase'),
 ('prices', 'price'),
 ('figures', 'figure'),
 ('cooling', 'cool'),
 ('is', 'be'),
 ('figures', 'figure'),
 ('are', 'be'),
 ('weakening', 'weaken'),
 ('coming', 'come'),
 ('housing', 'house'),
 ('is', 'be'),
 ('slowing', 'slow'),
 ('is', 'be'),
 ('backed', 'back'),
 ('lending', 'lend'),
 ('said', 'say'),
 ('is', 'be'),
 ('lagging', 'lag'),
 ('figures', 'figure'),
 ('said', 'say'),
 ('mortgages', 'mortgage'),
 ('approved', 'approve'),
 ('has', 'have'),
 ('fallen', 'fall'),
 ('said', 'say'),
```

```
 ('prices', 'price'),
 ('increased', 'increase'),
 ('said', 'say'),
 ('prices', 'price'),
 ('rose', 'rise'),
 ('is', 'be'),
 ('predicting', 'predict'),
 ('prices', 'price'),
 ('stabilises', 'stabilise'),
 ('gains', 'gain'),
 ('attributed', 'attribute'),
 ('prices', 'price'),
 ('detached', 'detach'),
 ('houses', 'house'),
 ('said', 'say'),
 ('rose', 'rise'),
 ('prices', 'price'),
 ('had', 'have'),
 ('fallen', 'fall'),
 ('showed', 'show'),
 ('was', 'be'),
 ('was', 'be'),
 ('said', 'say'),
 ('showed', 'show'),
 ('followed', 'follow'),
 ('had', 'have'),
 ('rose', 'rise')]
```

1.3.6 Sentiment & Polarity

TextBlob provides polarity and subjectivity estimates for parsed documents using dictionaries provided by the Pattern library. These dictionaries lexicon map adjectives frequently found in product reviews to sentiment polarity scores, ranging from -1 to +1 (negative positive) and a similar subjectivity score (objective subjective).

The .sentiment attribute provides the average for each over the relevant tokens, whereas the .sentiment_assessments attribute lists the underlying values for each token

[16]:
```
parsed_body.sentiment
```

[16]:
```
Sentiment(polarity=0.10447845804988663, subjectivity=0.44258786848072557)
```

[17]:
```
parsed_body.sentiment_assessments
```

[17]:
```
Sentiment(polarity=0.10447845804988663, subjectivity=0.44258786848072557,
assessments=[(['slightly'], -0.16666666666666666, 0.16666666666666666, None),
(['average'], -0.15, 0.39999999999999997, None), (['recent'], 0.0, 0.25, None),
(['strong'], 0.4333333333333333, 0.7333333333333333, None), (['likely'], 0.0,
```

```
1.0, None), (['overall'], 0.0, 0.0, None), (['down'], -0.15555555555555559,
0.2888888888888889, None), (['fairly'], 0.7, 0.9, None), (['nearly'], 0.1, 0.4,
None), (['last'], 0.0, 0.06666666666666667, None), (['first'], 0.25,
0.3333333333333333, None), (['rose'], 0.6, 0.95, None), (['whole'], 0.2, 0.4,
None), (['only'], 0.0, 1.0, None), (['second'], 0.0, 0.0, None), (['half'],
-0.16666666666666666, 0.16666666666666666, None), (['overall'], 0.0, 0.0, None),
(['large'], 0.21428571428571427, 0.42857142857142855, None), (['recent'], 0.0,
0.25, None), (['rose'], 0.6, 0.95, None), (['same'], 0.0, 0.125, None),
(['average'], -0.15, 0.39999999999999997, None), (['more'], 0.5, 0.5, None),
(['average'], -0.15, 0.39999999999999997, None), (['rose'], 0.6, 0.95, None),
(['only'], 0.0, 1.0, None), (['slightly'], -0.16666666666666666,
0.16666666666666666, None), (['previous'], -0.16666666666666666,
0.16666666666666666, None)])
```

1.3.7 Combine Textblob Lemmatization with `CountVectorizer`

[18]:
```
def lemmatizer(text):
    words = TextBlob(text.lower()).words
    return [word.lemmatize() for word in words]
```

[19]:
```
vectorizer = CountVectorizer(analyzer=lemmatizer, decode_error='replace')
```