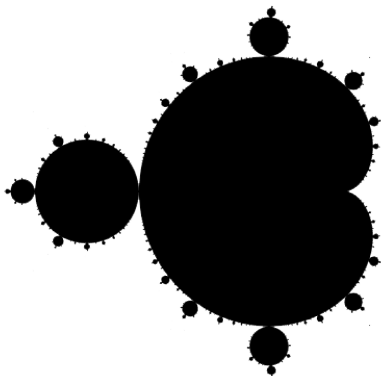


Natural Language Processing




TextBlob

Quick text analysis with TextBlob

- ▶ Noun phrases
 - ▶ "A small group of words standing together as a conceptual unit, typically forming a component of a clause."
- ▶ Sentiment analysis
 - ▶ Whether a text is positive, neutral or negative

TextBlob



The screenshot shows the GitHub repository page for TextBlob. The header includes the repository name 'TextBlob: Simplified Text Processing' and the release version 'v0.15.2'. A '5,962' star count is visible. The README text describes TextBlob as a Python library for NLP tasks. A code block shows how to import TextBlob and process a sample text about 'The Blob'.

TextBlob: Simplified Text Processing

Release v0.15.2. ([Changelog](#))

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

from textblob import TextBlob

```
text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating" flesh en contact.
Seize comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the gray goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''

blob = TextBlob(text)
blob.tags      # (('The', 'DT'), ('Blob', 'NN'), ('...', 'DT'), ...)
blob.noun_phrases # WordList(['titular threat', 'blob', 'ultimate movie monster', 'amoeba-like mass', ...])
```

→ Documentation

Noun phrases with TextBlob

```
1  from textblob import TextBlob
2
3  blob = TextBlob(text.lower())
4  print(blob.tags)
5  print(blob.noun_phrases)
6  print(blob.sentiment)
```

Excource: Regular Expression

- ▶ Regular expression are patterns to search for groups of characters
- ▶ Used in all programming languages, operating systems, search engines, etc.
- ▶ RegEx is f***ing fast

Excource: Regular Expression, cont.

- ▶ Standard characters match literally
 - ▶ letters: a to z, A to Z
 - ▶ digits: 0 to 9
 - ▶ symbols: !, % \$
- ▶ Combine them in groups with square or round brackets

Excource: Regular Expression, cont.

- ▶ Standard characters match literally
 - ▶ letters: a to z, A to Z
 - ▶ digits: 0 to 9
 - ▶ symbols: !,%\$
- ▶ Combine them in groups with square or round brackets
- ▶ Q: What matches `f` and `k` at the same time but not `S`?

Excource: Regular Expression, cont.

- ▶ Standard characters match literally
 - ▶ letters: a to z, A to Z
 - ▶ digits: 0 to 9
 - ▶ symbols: !, % \$
- ▶ Combine them in groups with square or round brackets
- ▶ Q: What matches f and k at the same time but not S?
- ▶ A: [a-z]

Excource: Regular Expression, cont.

- ▶ Some patterns indicate number of occurrences
 - ▶ 0 or more times: $*$
 - ▶ 1 or more times: $+$
 - ▶ exactly n times: n
 - ▶ exactly n till m times: n, m

Excource: Regular Expression, cont.

- ▶ Some patterns indicate number of occurrences
 - ▶ 0 or more times: `*`
 - ▶ 1 or more times: `+`
 - ▶ exactly `n` times: `n`
 - ▶ exactly `n` till `m` times: `n, m`
- ▶ Q: What matches `aaa` and `aa` but not `a`?

Excource: Regular Expression, cont.

- ▶ Some patterns indicate number of occurrences
 - ▶ 0 or more times: `*`
 - ▶ 1 or more times: `+`
 - ▶ exactly `n` times: `n`
 - ▶ exactly `n` till `m` times: `n, m`
- ▶ Q: What matches `aaa` and `aa` but not `a`?
- ▶ A: `[a]{2,3}`

Excource: Regular Expression, cont.

- ▶ There are groups of characters too:
 - ▶ All digits: `\d`
 - ▶ All non-digits: `\D`
 - ▶ Any character: `.`

Excercise: Regular Expression, cont.

- ▶ There are groups of characters too:
 - ▶ All digits: `\d`
 - ▶ All non-digits: `\D`
 - ▶ Any character: `.`
- ▶ Q: What matches `L0V3` and `L34D` at the same time, but not `light`?

Excource: Regular Expression, cont.

- ▶ There are groups of characters too:
 - ▶ All digits: `\d`
 - ▶ All non-digits: `\D`
 - ▶ Any character: `.`
- ▶ Q: What matches `L0V3` and `L34D` at the same time, but not `light`?
- ▶ A: `[LVD\d]*`

Excource: Regular Expression, cont.

- ▶ There are patterns for positions
 - ▶ Beginning of line: ^
 - ▶ End of line: \$

Excercise: Regular Expression, cont.

- ▶ There are patterns for positions
 - ▶ Beginning of line: ^
 - ▶ End of line: \$
- ▶ Q: How to match `Mission` in `Mission successful` but not in `Last Mission unsuccessful` and also not in `Next Mission python`?

Excource: Regular Expression, cont.

- ▶ There are patterns for positions
 - ▶ Beginning of line: ^
 - ▶ End of line: \$
- ▶ Q: How to match `Mission` in `Mission successful` but not in `Last Mission unsuccessful` and also not in `Next Mission python`?
- ▶ A: `^Mission` or similar

Excuse: Regular Expression, cont.

- ▶ Finally exclude characters with ^ (at the beginning of [])

Excercise: Regular Expression, cont.

- ▶ Finally exclude characters with `^` (at the beginning of `[]`)
- ▶ Q: What matches `hot` and `dog` at the same time but not `bog`?

Excuse: Regular Expression, cont.

- ▶ Finally exclude characters with `^` (at the beginning of `[]`)
- ▶ Q: What matches `hot` and `dog` at the same time but not `bog`?
- ▶ A: `[^b]og`