Основне академске студије
Информациони инжењеринг

Методе и технике науке о подацима

Обрада природног језика и анализа текста у језику Python

(материјали за вежбе)

Библиотека NLTK

Библиотека *NLTK* (*Natural Language Toolkit*)

- посвећена обради садржаја на природном језику и анализи текста
- нуди разноврсне програмске рутине
- нуди корпусе и лексичке ресурсе

Библиотека *NLTK*

- основни подаци
  - покретачи Стивен Берд и Едвард Лоупер
  - настанак 2001. године
  - слободна за употребу
  - актуелна верзија 3.9.2
  - Интернет сајт
    - https://www.nltk.org/

Библиотека NLTK

Припрема за коришћење

- инсталација библиотеке
  - библиотека доступна у оквиру дистрибуције *Anaconda*
  - могућа засебна инсталација
    - употребом *pip*
    - употребом *conda*
- учитавање библиотеке

```
import nltk
```

Библиотека NLTK

Припрема за коришћење

- преузимање додатних садржаја
  - могућност преузимања појединачних пакета или колекција пакета
    - пакети могу обухватити корпусе, граматике, моделе...
  - преузимање помоћу функције **`nltk.download(...)`**
  - примери преузимања
    - преузимање граматика за баскијски језик
      `nltk.download("basque_grammars")`
    - преузимање популарних садржаја
      `nltk.download("popular")`
    - преузимање свих корпуса
      `nltk.download("all_corpora")`
    - преузимање свих садржаја
      `nltk.download("all")`

Обрада природног језика и анализа текста

Библиотека *NLTK*

- корпус – честе основне функције за рад над корпусом
  - преглед доступног садржаја
    - функција **`fileids()`**
  - учитавање текста
    - функција **`paras(...)`**
      учитавање у облику пасуса
    - функција **`sents(...)`**
      учитавање у облику реченица
    - функција **`words(...)`**
      учитавање у облику речи
    - функција **`raw(...)`**
      учитавање у необрађеном облику
    - ...

Библиотека *NLTK*

корпус – пример (1/2)

припрема корпуса *Machado* за коришћење

```
import nltk
nltk.download("machado")
nltk.download("punkt_tab")
from nltk.corpus import machado
```

Библиотека *NLTK*

корпус – пример (2/2)

учитавање текста из корпуса *Machado*

```
pasusi = machado.paras("contos/macn005.txt")
print(pasusi[32])
```

```
[['Hamlet', 'observa', 'a', 'Horácio', 'que', 'há',
'mais', 'coisas', 'no', 'céu', 'e', 'na', 'terra', 'do',
'que', 'sonha', 'a', 'nossa', 'filosofia', '.'], ['Era',
'a', 'mesma', 'explicação', 'que', 'dava', 'a', 'bela',
'Rita', 'ao', 'moço', 'Camilo', ',', 'numa', 'sexta',
'-', 'feira', 'de', 'novembro', 'de', '1869', ',',
'quando', 'este', 'ria', 'dela', ',', 'por', 'ter',
'ido', 'na', 'véspera', 'consultar', 'uma', 'cartomante',
';', 'a', 'diferença', 'é', 'que', 'o', 'fazia', 'por',
'outras', 'palavras', '.']]
```

Библиотека *NLTK*

- стоп-речи (енгл. *stopwords*)
  - речи које су честе у текстовима и обично немају битан утицај на анализу садржаја, те бивају уклоњене из обрађиваног текста

Библиотека *NLTK*

стоп-речи – пример (1/3)

припрема корпуса стоп-речи

```
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
```

Библиотека *NLTK*

стоп-речи – пример (2/3)

стоп-речи за португалски језик

```
stop_reči = stopwords.words("portuguese")
print(stop_reči[::5])
```

```
['a', 'aquelas', 'às', 'das', 'deles', 'é', 'em', 'essa',
'está', 'estava', 'estejam', 'estivemos', 'estiverem',
'estou', 'fora', 'fosse', 'haja', 'haver', 'houvera',
'houverei', 'houveríamos', 'isso', 'mais', 'meus', 'não',
'nós', 'num', 'para', 'por', 'são', 'sem', 'seremos',
'seus', 'suas', 'temos', 'terá', 'teriam', 'tinha',
'tiver', 'tivermos', 'tua', 'vocês']
```

Библиотека *NLTK*

стоп-речи – пример (3/3)

стоп-речи за енглески језик

```
stop_reči = stopwords.words("english")
print(len(stop_reči))
198
print(stop_reči[50:100:2])
["hadn't", 'hasn', 'have', "haven't", 'he', "he'll",
'here', 'herself', 'him', 'his', 'i', 'if', "i'm",
'into', 'isn', 'it', "it'll", 'its', "i've", 'll', 'ma',
'mightn', 'more', 'mustn', 'my']
```

Библиотека *NLTK*

- токенизација
  - токенизатор
    - растављач текста на токене (језичке јединице)
- основни начини растављања
  - растављање на речи и знакове интерпункције
    - функција **`word_tokenize(...)`**
  - растављање на реченице
    - функција **`sent_tokenize(...)`**
  - растављање на основу регуларног израза
    - класа **`RegexpTokenizer`**
  - ...

Библиотека *NLTK*

токенизација – пример (1/2)

припрема токенизатора типа *Punkt*

```
import nltk
nltk.download("punkt")
from nltk.tokenize import word_tokenize
```

Библиотека *NLTK*

токенизација – пример (2/2)

примена токенизатора типа *Punkt*

```
tekst = "Sometimes the hills were so steep that, despite
our driver’s haste, the horses could only go slowly."
tokeni = word_tokenize(tekst)
print(tokeni)
```

```
['Sometimes', 'the', 'hills', 'were', 'so', 'steep',
'that', ',', 'despite', 'our', 'driver', '’', 's',
'haste', ',', 'the', 'horses', 'could', 'only', 'go',
'slowly', '.']
```

текст из дела *Дракула* од Брема Стокера

Библиотека *NLTK*

- стемовање
  - стемер
    - уклањач афикса
  - примери стемера
    - *Porter Stemmer* (Портеров стемер)
    - *Snowball Stemmer* (стемер *Snowball*)
- поступак стемовања
  - издвајање основне морфеме из речи
    - функција **`stem(...)`**

Библиотека *NLTK*

стемовање – пример (1/2)

припрема и примена Портеровог стемера

```
import nltk
from nltk.stem import PorterStemmer

reči = ['sometimes', 'hills', 'so', 'steep', 'despite',
'driver', 'haste', 'horses', 'only', 'go', 'slowly']
stemer = PorterStemmer()
for reč in reči:
    stem = stemer.stem(reč)
    print("{} -> {}".format(reč, stem))
```

Библиотека *NLTK*

стемовање – пример (2/2)

резултати стемовања Портеровим стемером

```
sometimes -> sometim
hills -> hill
so -> so
steep -> steep
despite -> despit
driver -> driver
haste -> hast
horses -> hors
only -> onli
go -> go
slowly -> slowli
```

Библиотека *NLTK*

- лематизација
  - лематизатор
    - уклањач афикса који за резултат даје реч из речника
- поступак лематизације
  - одређивање леме за реч
    - функција **`lemmatize(...)`**

Библиотека *NLTK*

лематизација – пример (1/2)

припрема лематизатора типа *WordNet*

```
import nltk
nltk.download("wordnet")
from nltk.stem import WordNetLemmatizer
```

Библиотека *NLTK*

лематизација – пример (2/2)

примена лематизатора типа *WordNet*

```
tekst = "sometimes the hills were so steep that despite
our driver’s haste the horses could only go slowly"
delovi = tekst.split()
lematizator = WordNetLemmatizer()
leme = [lematizator.lemmatize(deo) for deo in delovi]
print(leme)
['sometimes', 'the', 'hill', 'were', 'so', 'steep',
'that', 'despite', 'our', 'driver’s', 'haste', 'the',
'horse', 'could', 'only', 'go', 'slowly']
```

Библиотека *NLTK*

- расподела учесталости
  - учесталости појављивања за речи
- одређивање расподеле учесталости
  - прихват листе речи за коју треба одредити расподелу учесталости
    - класа **`FreqDist`**
    - функција **`__init__(...)`**
  - очитавање броја појављивања за најзаступљеније речи
    - функција **`most_common(...)`**

Библиотека *NLTK*

расподела учесталости – пример (1/2)

припрема текста и израчунавање учесталости

```
import nltk
nltk.download("punkt")
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

tekst = "sometimes the hills were so steep that despite
our driver’s haste the horses could only go slowly"
tokeni = word_tokenize(tekst)
raspodela = FreqDist(tokeni)
```

Библиотека *NLTK*

расподела учесталости – пример (2/2)

учесталости

```
print(raspodela.most_common())
```

```
[('the', 2), ('sometimes', 1), ('hills', 1), ('were', 1),
('so', 1), ('steep', 1), ('that', 1), ('despite', 1),
('our', 1), ('driver', 1), ('’', 1), ('s', 1), ('haste',
1), ('horses', 1), ('could', 1), ('only', 1), ('go', 1),
('slowly', 1)]
```

Представљање речи у бројчаном облику

често коришћене представе

јединична представа (енгл. *one-hot encoding*)

- реч представљена n-димензионалним вектором при чему је n број речи у коришћеном речнику
- појединачним речима одговарају различите појединачне компоненте вектора
- ако посматраној речи одговара i-та компонента вектора, онда у вектору који представља ту реч i-та компонента има вредност 1 а све остале вредност 0

уградна представа (енгл. *word embedding*)

- реч представљена вектором при чему блиским речима треба да одговарају блиски вектори
- димензија вектора у случају уградне представе очекивано битно мања него она у случају јединичне представе
- одређивање векторских представа за коришћене речи може бити изведено помоћу неуронске мреже
- током обучавања неуронске мреже долази до формирања погодних представа

Модел типа *Word2Vec*

- познати модел за уградну представу речи
  - речи бивају уграђене у векторски простор
    - за речи које су блиске у односу на контекст очекивано је да представе тих речи буду међусобно близу у векторском простору
  - заснива се на употреби неуронске мреже с једним скривеним слојем
    - представа речи се подудара с одређеним тежинским коефицијентима
- основне варијанте модела типа *Word2Vec*
  - континуални модел скок-грама (енгл. *continuous skip-gram model*)
  - континуални модел мултискупа речи (енгл. *continuous bag-of-words model*)

Библиотека Gensim

Библиотека *Gensim* (*Generate Similar*)

- подржава програмску обраду неструктурираног текста
- ослања се на примену ненадгледаног машинског учења

Библиотека *Gensim*

- основни подаци
  - покретач Радим Рехуржек
  - настанак 2008. године
  - слободна за употребу
  - актуелна верзија 4.4.0
  - Интернет сајт
    - https://radimrehurek.com/gensim/

Библиотека *Gensim*

- модел типа *Word2Vec*
  - формирање модела
    - класа **`Word2Vec`**
  - употреба уградних представа
    - приступ уградним представама у оквиру модела
      - члан **`wv`** (инстанца класе **`KeyedVectors`**)
        - пресликавање речи на уградне представе
    - одређивање најсличнијих речи у односу на назначене речи у оквиру пресликавања
      - функција **`most_similar(...)`**
    - одређивање косинусне сличности између две речи у оквиру пресликавања
      - функција **`similarity(...)`**

Библиотека *Gensim*

- модел типа *Word2Vec* – генерички пример
  - припрема модела за одређени улаз и добављање уградне представе за одређену реч

```
from gensim.models import Word2Vec

model = Word2Vec(ulaz, vector_size=50, epochs=20)
vektor = model.wv[reč]
```

Задаци

Задатак 1.

Из корпуса који је посвећен Општој декларацији о људским правима учитати верзију декларације на једном од доступних језика. Приказати текст декларације у основном облику, као и у облику у којем је текст растављен на реченице. Издвојити и приказати један члан из декларације.

Задаци

Задатак 2.

Из корпуса *Gutenberg* засебно за Џејн Остин и Вилијама Шекспира учитати у обједињеном облику сва њихова расположива дела. Оба учитана садржаја раставити по речима и избацити стоп-речи. За оба садржаја одредити 20 најучесталијих лема али узимајући у обзир само леме састављене од барем 10 знакова.

Задаци

Задатак 3.

Из корпуса *Gutenberg* обједињено учитати сва расположива дела Џејн Остин и Гилберта Кита Честертона. Учитани садржај уредити и над њим формирати модел типа *Word2Vec*. Применом формираног модела, за одабрану реч одредити најсличније речи а за одабрану групу речи одредити међусобне сличности.

Литература

Основна литература

NLTK Project. NLTK :: Natural Language Toolkit. Internet: https://www.nltk.org/

NLTK Project. NLTK :: nltk package. Internet: https://www.nltk.org/api/nltk.html

NLTK Project. NLTK :: Installing NLTK. Internet: https://www.nltk.org/install.html

NLTK Project. NLTK :: Installing NLTK data. Internet: https://www.nltk.org/data.html

GitHub. FAQ · nltk/nltk Wiki · GitHub. Internet: https://github.com/nltk/nltk/wiki/FAQ

Bird S, Klein E, Loper E. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit. 2019. Internet: https://www.nltk.org/book/

Литература

Основна литература

Jurafsky D, Martin JH. Speech and Language Processing. 3rd edition draft. 2026. Internet: https://web.stanford.edu/~jurafsky/slp3/

Géron A. Mašinsko učenje: Scikit-Learn, Keras i TensorFlow: koncepti, alati i tehnike za izgradnju inteligentnih sistema. Prevod 2. izdanja. O'Reilly (Sebastopol, CA, USA), Mikro knjiga (Beograd, Srbija); 2021.

TensorFlow. Word embeddings | Text | TensorFlow. Internet: https://www.tensorflow.org/text/tutorials/word_embeddings

TensorFlow. word2vec | Text | TensorFlow. Internet: https://www.tensorflow.org/text/tutorials/word2vec

Rehurek R, Sojka P. Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. University of Malta (Valletta, Malta); 2010. p. 46–50.

Литература

Основна литература

Radim Rehurek. Gensim: Topic modelling for humans. Internet: https://radimrehurek.com/gensim/

Radim Rehurek. API reference — gensim. Internet: https://radimrehurek.com/gensim/apiref.html

Radim Rehurek. Word2Vec model — gensim. Internet: https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html

GitHub. GitHub - piskvorky/gensim: Topic modelling for humans · GitHub. Internet: https://github.com/piskvorky/gensim/

Project Gutenberg. Dracula by Bram Stoker. Internet: https://gutenberg.org/ebooks/345