NLP in Python

NLTK

전 용훈

Table of Contents

NLP introduction

- Goal
  - ✓ Speech recognition, Natural-language understanding, Natural-language generation
  - ✓ Extract meaning from text
- Task
  - ✓ Part-of-speech(POS) tagging
  - ✓ Sentiment analysis
  - ✓ Document classification
  - ✓ Topic modeling
  - ✓ Etc.

NLP introduction

- Top-N Libraries
  - ✓ The Conqueror: NLTK
    "most famous, many libraries but slow"
  - ✓ The Prince: TextBlob
    "for processing textual data, fast & easy"
  - ✓ The Mercenary: Stanford CoreNLP
    "POS tagging, entity recognition, pattern learning, parsing, etc., written in Java"
  - ✓ The Usurper: SpaCy
    "new but extremely optimized, with DL frameworks such as TF or Torch"
  - ✓ The Admiral: genism
    "highly optimized for (unsupervised) semantic (topic) modelling."

NLTK tutorials

- Source-code
  https://github.com/yonghoonjhun/NLP/tree/master/NLTK/Tutorials
- Idea
  ① Tokenize words and sentences
  sentence는 문장 단위로, word는 단어 단위로 split된다.
  여러 pre trained 된 Tokenizer가 있는데 적합한 tokenize를 위해서는 own data를 가지고 학습이 필요하다.
  ② Stop words
  NLP에서 쓸모 없는 데이터를 filtering할 수 있다.
  ex) a, how, now ...etc.
  ③ Stemming words(어간 추출)
  단어에 공통적으로 나타나는 부분을 어근으로 처리함.
  ex) ride, riding, rode -> ride
  ④ Part of Speech(POS) Tagging
  문장에서의 단어들을 형태소에 따라 labelling 하는 것.
  형태소 label 정보: https://imgur.com/RXBrbue
  ex) ('PRESIDENT', 'NNP'), ('GEORGE', 'NNP'), ('W.', 'NNP'), ('BUSH', 'NNP')

NLTK tutorials

- Idea (cont'd)
  - ⑤ Chunking
    Regex을 사용하여 의미가 있거나 관련 있는 명사구 등으로 그룹화하는 것.
    ex) (Chunk PRESIDENT/NNP GEORGE/NNP W./NNP BUSH/NNP)
  - ⑥ Chinking
    chunking을 해도 남아있는 단어들을 더 chunking하는 것.
    "the chunk that you remove from your chunk"
  - ⑦ Named Entity recognition(NER)
    미리 정의해 둔 사람, 회사, 장소, 시간, 단위 등에 해당하는 단어(개체명)를 문서에서 인식하여 추출 분류하는 기법.
  - ⑧ Lemmatizing
    stemming과 비슷하지만 다르다.
    stemming은 존재하지 않는 어근을 생성할 수도 있지만, lemmatizing은 실제 존재하는 단어로 생성한다.
  - ⑨ Corpora
    corpus의 복수

NLTK tutorials

- Idea (cont'd)
  - ⑩ Wordnet
    "a lexical database for the English language, created by Princeton"
    단어의 synonyms, antonyms, 단어 간 similarity등을 계산 할 수 있다.
    ex) ship과 boat의 similarity는 약 0.909
  - ⑪ Text Classification
    sentiment analysis나 spam classification등 지도학습 이므로 data에 labelling이 우선 되어야한다.
    준비된 영화 리뷰 데이터를 가지고 긍정/부정 classification을 한다.
  - ⑫ Converting words to Features
    label대로 빈출 단어의 출현유무를 알고자 함.
  - ⑬ Naïve Bayes Classifier
  - ⑭ Saving Classifiers
    pickle 라이브러리 사용해 학습 모델 object를 save하거나 load함.
  - ⑮ Scikit-Learn Sklearn
    다른 classifier 알고리즘을 사용하기 위함.

TODO

- 무엇이 필요한가?
- 어떻게 적용할 것인가?

References

- Top-N Libraries
  https://kleiber.me/blog/2018/02/25/top-10-python-nlp-libraries-2018/
  https://elitedatascience.com/python-nlp-libraries
- Tutorials
  https://pythonprogramming.net/data-analysis-tutorials/

감사합니다.