


Ku - Big

자연어 처리

실전반

kubig 11기 강호석



커리큘럼



1. Text Preprocessing

- Tokenize
- Cleaning and normalization
- encoding

2. Word Representation

2-1 Local Representation

- Bag of words
- Document term Matrix
- TF-IDF
- * Document Similarity

2-2 Continuous Representation

- Word2Vec
- Glove, Elmo

3. Text Classification with RNN

- Naive bayes
- BiLSTM

4. 심화과정

- Tagging
- Encoder & Decoder / Transformer

자연어 처리란?

자연어 처리란?

NLP : Natural Language Processing

자연어 : 일상 생활에서 사용하는 언어

자연어 처리 : 자연어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 하는 작업

텍스트 분류(Classification), 감성분석, 텍스트 요약(Topic modeling), 기계 번역(파파고)

자동응답 시스템(시리, 빅스비, 챗봇)

텍스트 전처리

- 토큰화 -

텍스트 전처리 - 토큰화

- 토큰화

자연어처리 모델을 적용하기전 사전 처리 작업

말 뭉치(Corpus)를 특정 기준에 맞춰 분리하는 작업

어떤 기준으로 토큰을 나누는 지가 중요 : 구두점, 띄워쓰기

ex) Don't => Don't / Don t / Dont / Do n't

Jone's => Jone's / Jone s / Jone / Jones

텍스트 전처리 - 토큰화

- 토큰화 시 주의해야 하는 사항

1. 구두점과 특수문자의 무분별한 배제

ex) 온점(.)은 문장의 경계를 나누는 정보를 포함

단어 자체에 구두점이 포함되는 단어 : Ph.D, AT&T

구두점이 정보를 가지는 경우 : \$45.55, 01/02/06, 123,456,789

텍스트 전처리 - 토큰화

- 토큰화 시 주의해야 하는 사항

2. 줄임말과 단어 내에 띄워쓰기가 있는 경우

ex) 줄임말 : what're => what과 are의 정보 / re 부분을 접어라고 명명

단어 내 띄워쓰기 : New York , rock 'n' roll

텍스트 전처리-토큰화

- 토큰화 시 주의해야 하는 사항

3. 표준 토큰화 예제(Penn Treebank Tokenization의 규칙)

규칙 1 : 하이픈으로 구성된 단어는 하나로 유지한다.

규칙 2 : 아포스트로피로 '접어'가 함께하는 단어는 분리해준다.

"Starting a home-based restaurant may be an ideal. it doesn't have a food chain or restaurant of their own."

['Starting', 'a', 'home-based', 'restaurant', 'may', 'be', 'an', 'ideal.', 'it', 'does', 'n't', 'have', 'a', 'food', 'chain', 'or', 'restaurant', 'of', 'their', 'own', '.']

home-based는 유지, doesn't는 접어와 분리

텍스트 전처리 - 토큰화

- 토큰화 시 주의해야 하는 사항

4. 품사 태깅

같은 단어라도 문장마다 쓰이는 의미 / 품사가 다르다

단어 토큰화 과정에서 각 단어가 어떤 품사로 쓰였는지를 구분

ex) fly : 날다(동사) / 파리(명사)

못 : 동사의 부정적 의미 부여 / 사물을 고정시키는 물체

텍스트 전처리-토큰화

- 토큰화 시 주의해야 하는 사항

5. 문장 토큰화

문장 단위로도 정제되지 않은 자료의 경우, 먼저 문장을 구분 하는 토큰화 작업 진행

단순하게 온점,느낌표,물음표를 기준으로 삼으면 문제 발생

ex1) IP 192.168.56.31 서버에 들어가서 로그 파일 저장해서 ukairia777@gmail.com로 결과 좀 보내줘. 그리고 나서 점심 먹으러 가자.

ex2) Since I'm actively looking for Ph.D. students, I get the same question a dozen times every year.

텍스트 전처리 - 토큰화

- 한국어 토큰화의 어려움

1. 띄워쓰기만으로 구분하기 어렵다.

영어는 띄워쓰기가 없으면 문장을 이해하기 어렵지만 한국어는 쉬운 편

역설적으로 영어는 띄워쓰기만 잘 파악해도 토큰화 하기 쉽다.

ex1) 제가이렇게띄어쓰기를전혀하지않고글을썼다고하더라도글을이해할수있습니다.

ex2) Tobeornottobethatisthequestion

텍스트 전처리 - 토큰화

- 한국어 토큰화의 어려움

2. 한국어는 교착어

조사(은,는,이,가,에게....)의 영향으로 띄워쓰기만 고려해서는 토큰화가 어렵다

ex) he / 그는, 그가, 그에게 : 영어에 비해 다양한 형태가 존재

따라서 한국어는 형태소 토큰화 작업이 필요하다.

형태소 : 의미를 가지는 가장 작은 낱말 단위

한국어 토큰화 : Okt(Open Korea Text), 메cab(Mecab), 코모란(Komoran), 한나눔(Hannanum),꼬꼬마(Kkma)

텍스트 전처리

- 정제 및 정규화 -

텍스트 전처리 - 정제 및 정규화

- 정제(Cleaning) / 정규화(Normalization)

정제(cleaning) : 갖고 있는 코퍼스로부터 노이즈 데이터를 제거한다.

정규화(normalization) : 표현 방법이 다른 단어들을 통합시켜서 같은 단어로 만들어준다.

- 토큰화 전후로 목적에 맞게 활용
- 완벽한 정제 및 정규화는 없다 / 목적에 따라 합의점을 찾음

텍스트 전처리 - 정제 및 정규화

- 규칙에 기반한 표기가 다른 단어들의 통합

다른 단어이지만 같은 의미를 가지는 단어들을 묶는 작업

ex) US / USA, uh-huh / uhhuh

텍스트 전처리 - 정제 및 정규화

- 대소문자 통합

영어의 경우 같은 단어임에도 대소문자로 인해 다른 단어로 취급될 가능성

ex) Automobile / automobile => automobile로 통합

단, 고유명사 및 대소문자로 구분되는 경우도 고려

ex) US(미국) / us(우리) , bush(풀숲) / Bush(사람 이름)

텍스트 전처리 - 정제 및 정규화

- 불필요한 단어 제거

1. 등장 빈도가 작은 단어

스팸 메일 분류기 학습시 10만개의 데이터 중에서 5번 등장한 단어

학습시 영향력이 매우 낮음 : 제거

텍스트 전처리 - 정제 및 정규화

- 불필요한 단어 제거

2. 길이가 짧은 단어

한국어와 달리 영어는 글자가 가지는 함축적 의미가 거의 없어 길이가 긴 편

ex) 학교 : 배울 학(學)과 학교 교(校) / school : s,c,h,o,o,l

영어에서 짧은 단어를 제거 시 많은 불필요한 단어 제거 가능

ex) 1글자 : a / I , 2글자 : as / to / by / it 등 전치사 및 지시대명사

3글자 이상부터는 고민이 필요 : out / this / that vs car / dog / bus

텍스트 전처리 - 정제 및 정규화

- 불용어(Stopwords)

불용어 : 분석에 큰 의미가 없는 단어들

ex) the, a, an, is, I, my, 이것, 저것

불용어 데이터에 따라 사용자가 정의를 내리는 경우가 많아 csv 형태로 저장 후 활용

자주 쓰이는 불용어 리스트 <https://www.ranks.nl/stopwords/korean>

텍스트 전처리 - 정제 및 규제화

- 표제어 추출(Lemmatization)과 어간 추출(Stemming)
 - 두 방법 모두 코퍼스의 복잡도를 줄이는 정규화 과정
 - 다른 단어지만 의미상으로 같은 단어를 통합시켜 단어사전(Bag of words)를 최소화시키고자 하는 목적

텍스트 전처리 - 정제 및 규제화

- 표제어 추출

다양한 단어의 뿌리 단어를 찾아가는 과정 : am/is/are => be

형태소 파싱(parsing) 이후 어간 / 접사를 분리하는 방법 : cat => cat / -s

어간 추출에 비해 원본 유지력이 뛰어남 + 단어의 품사정보를 보존

품사정보가 없을 시 의미를 알 수 없는 단어를 반환

dies ,watched,has =>	품사정보 입력 전	품사정보 입력 후
	dy	die
	watched	watch
	ha	have

텍스트 전처리 - 정제 및 규제화

- 어간 추출

정해진 규칙만을 적용하여 단어의 어미를 자르는 작업

	ALIZE → AL	formalize → formal
ex)	ANCE → 제거	allowance → allow
	ICAL → IC	electrical → electric

표제화 추출에 비해 반환된 단어의 형태가 사전에 없을 가능성이 높다

'This', 'was', 'not' ⇒ 'thi', 'wa', 'not'

어간 추출에 비해 속도가 빠르며 영어에 적용할 때 적절한 방법

패키지마다 규칙이 다르기 때문에 결과 값이 다르다

텍스트 전처리

- 인코딩 -

텍스트 전처리-인코딩

- 정수 인코딩

정제 / 정규화 / 어간 추출 / 토큰화 이후 축소화된 텍스트를 숫자로 변환하는 작업

각 단어를 고유한 정수에 매핑하는 방법 : book \Rightarrow 147, car \Rightarrow 368

일반적으로 빈도순으로 정렬 후 정수를 매핑

```
{'barber': 8, 'person': 3, 'good': 1, 'huge': 5, 'knew': 1, 'secret': 6, 'kept': 4, 'word': 2, 'keeping': 2, 'driving': 1, 'crazy': 1, 'went': 1, 'mountain': 1}
```

텍스트 전처리-인코딩

- 패딩

문장마다 길이가 다르기 때문에, 길이가 짧은 문장을 임의적으로 늘려주는 작업

```
[[1, 5], [1, 8, 5], [1, 3, 5], [9, 2], [2, 4, 3, 2], [3, 2], [1, 4, 6], [1, 4, 6], [1, 4, 2], [7, 7, 3, 2, 10, 1, 11], [1, 12, 3, 13]]
```



```
array([[ 1,  5,  0,  0,  0,  0,  0],
       [ 1,  8,  5,  0,  0,  0,  0],
       [ 1,  3,  5,  0,  0,  0,  0],
       [ 9,  2,  0,  0,  0,  0,  0],
       [ 2,  4,  3,  2,  0,  0,  0],
       [ 3,  2,  0,  0,  0,  0,  0],
       [ 1,  4,  6,  0,  0,  0,  0],
       [ 1,  4,  6,  0,  0,  0,  0],
       [ 1,  4,  2,  0,  0,  0,  0],
       [ 7,  7,  3,  2, 10,  1, 11],
       [ 1, 12,  3, 13,  0,  0,  0]])
```

텍스트 전처리-인코딩

- 원핫 인코딩

단어 집합 / 사전을 만든 뒤, 각 단어에 인덱스를 부여

단어 별로 해당 단어가 있는 열에 1 나머지는 0을 부여하는 방식

Pet	Cat	Dog	Turtle	Fish
Cat	1	0	0	0
Dog	0	1	0	0
Turtle	0	0	1	0
Fish	0	0	0	1
Cat	1	0	0	0

텍스트 전처리-인코딩


- 원핫 인코딩

단어 사전이 커질수록 벡터 저장 공간이 늘어남 / 매우 비효율적인 방법

단어간의 유사성을 보여줄 수 없음


$$\begin{pmatrix} 1.0 & 0 & 5.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.0 & 0 & 0 & 0 & 0 & 11.0 & 0 \\ 0 & 0 & 0 & 0 & 9.0 & 0 & 0 & 0 \\ 0 & 0 & 6.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.0 & 0 & 0 & 0 & 0 \\ 2.0 & 0 & 0 & 0 & 0 & 10.0 & 0 & 0 \\ 0 & 0 & 0 & 8.0 & 0 & 0 & 0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 12.0 \end{pmatrix} \Rightarrow \text{단어를 다차원 공간에 벡터화하여 해결}$$

LSA, HAL, Word2Vec, Glove 등



제목을 입력하십시오

부제목을 입력하십시오



제목을 입력하십시오

제목을 입력하십시오

내용을 입력하십시오

제목을 입력하십시오

내용을 입력하십시오

내용을 입력하십시오

xx%

내용을 입력하십시오

제목을 입력하십시오

제목을 입력하십시오

내용을 입력하십시오

제목을 입력하십시오

제목을 입력하십시오

부제목을 입력하십시오

내용을 입력하십시오

내용을 입력하십시오

xx%

내용을 입력하십시오

