한국어 임베딩 3장

3.1 데이터 확보

3.1.1 한국어 위키백과

- 한국어 말뭉치로서는 한국어 위키백과만큼 방대한 데이터가 없다.
- https://github.com/ratsgo/embedding/issues/136

3.1.2 KorQuAD

- 한국어 기계 독해를 위한 데이터셋이다.
- 지문(Paragraph)-질문(Question)으로 구성되어 있고 이러한 쌍을 사람들이 직접 만들었다.

3.1.3 네이버 영화 리뷰 말뭉치

- 네이버 영화 페이지의 영화 리뷰들을 평점과 함께 수록한 한국어 말뭉치
- 감성 분석(sentiment analysis) 혹은 문서 분류(document classification) 테스크 수행에 적합하다
- document에 따른 label(긍정, 부정)이 달려 있는 문서들이다.

3.2 지도 학습 기반 형태소 분석

- 한국어 = 조사와 어미가 발달한 교착어(agglutinative language)
- 가다 → 가겠다, 가더라 등 많은 활용형이 존재한다. 이를 처리하는 기준이 필요
- 은전한닢(Mecb) 형태소 분석 결과

  가겠다 → 가, 겠, 다

  가더라 → 가, 더라
- 지도학습 : 정답이 있는 데이터의 패턴을 학습해 모델이 정답을 맞도록 하는 기법
- 태깅 : 모델 입력과 출력 쌍을 만드는 작업

  ex) 입력 : 아버지가방에들어가신다. 출력 : 아버지, 가, 방, 에, 들어가, 신다.

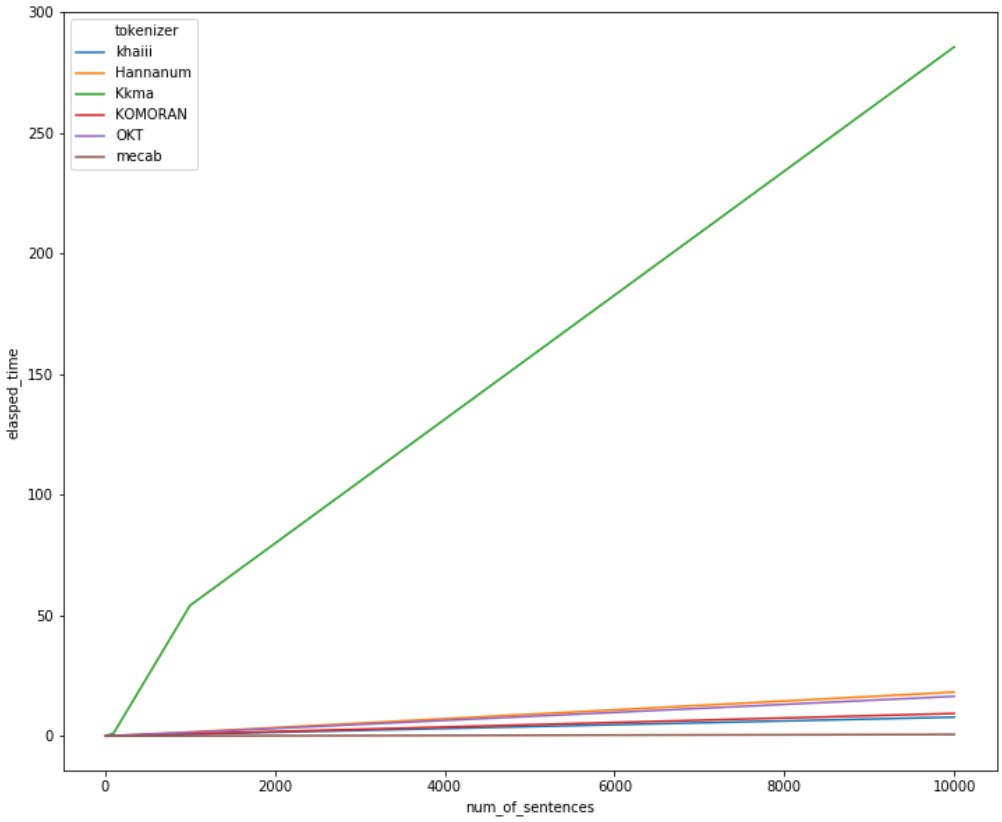
3.2.1 KoNLPy 사용법

- KoNLPy : 은전한닢, 꼬꼬마, 한나눔, Okt, 코모란 등 5개 오픈소스 형태소 분석기를 파이썬 환경에서 사용할 수 있도록 인터페이스를 통일한 한국어 자연어 처리 패키지다.
- 은전한닢(Mecab) 형태소 분석결과

  ['아버지', '가', '방', '에', '들어가', '산다'

  [('아버지', 'NNG'), ~~~ , ('신다', 'EP + EC)] → 이런식으로 품사별 분류까지 가능하다
- 이 외의 다른 꼬꼬마, 한나눔, Okt 역시 비슷하게 형태소를 분석해준다. 하지만 각각의 성능차이가 분명하게 존재한다.

3.2.2 KoNLPy 내 분석기별 성능 차이 분석

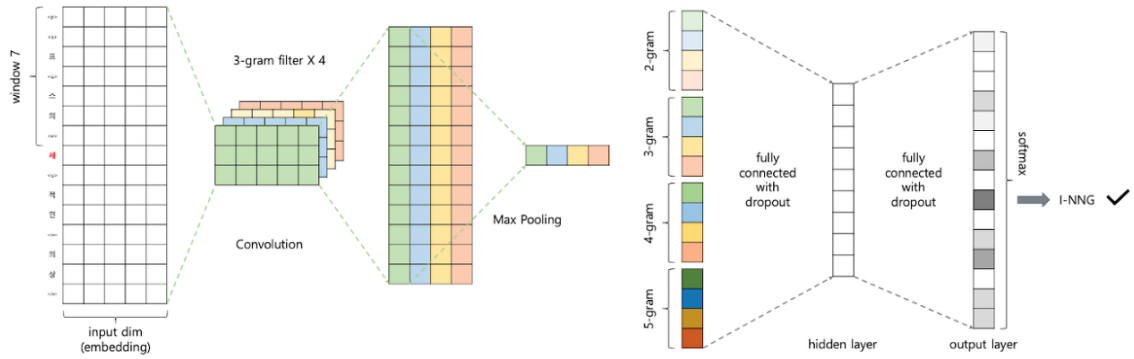


- 위의 표에서 카카오에서 공개한 khaii라는 형태소 분석기를 포함한 분석결과를 보여준다.
- 그 결과, 6개의 형태소 분석기 모두 문자개수가 많아질 때 실행시간이 기하급수적으로 증가함을 알 수 있다.

- 또한, 은전한닢(mecab)이 다른 분석기 대비 속도가 빠른 것을 확인할 수 있다.

3.2.3 Khaiii 사용법

- Khaiii : 카카오가 2018년말 공개한 오픈소스 한국어 형태소 분석기이다.
  1. 입력 문장을 문자 단위로 읽어 들인 뒤 컨볼루션 필터(convoultion filter)가 이 문자들을 슬라이딩해 가면서 정보를 추출한다.
  2. 이러한 정보를 종합해 형태소의 경계와 품사 태그를 예측한다.
- Khaiii 아키텍처



- Khaiii 품사 정보 확인

  ['아버지/NNG', '가/JKS', '방에/NNG'~~ 'ㄴ다/EC']

3.2.4 은전한닢에 사용자 사전 추가하기

- "가우스 전자 텔레비전 정말 좋네요"

  → ['가우스', '전자', '텔레비전', '정말', '좋', '네요']

  위와 같이 형태소 분석이 된경우 관심 단어인 가우스전자가 두 개의 토큰으로 분석된 것을 확인 할 수 있다. 이를 하나의 토큰으로 분석될 수 있도록 강제해야 한다.

  방법은 간단하다. → mecb-user-dic.csv에 붙여 썻으면 하는 단어를 추가해 주면 된다

  ex) 가우스전자,,,,NNP,*,F,가우스전나,*,*,*,*,*

  이 후 코드 실행시 잘 돌아가는 것을 확인 할 수 있다.

3.3 비지도 학습 기반 형태소 분석

- 비지도 학습 :데이터의 패턴을 모델 스스로 학습하게 함으로써 형태소를 분석하는 방법, 즉 데이터에 자주 등장하는 단어들을 형태소로 인식하는 방법

3.3.1 soynlp 형태소 분석기

https://github.com/lovit/soynlp/blob/master/tutorials/wordextractor_lecture.ipynb

- soynlp : 형태소 분석, 품사 판별 등을 지원하는 파이썬 한국어 자연어 처리 패키지
- 응집 확률(Cohesion)

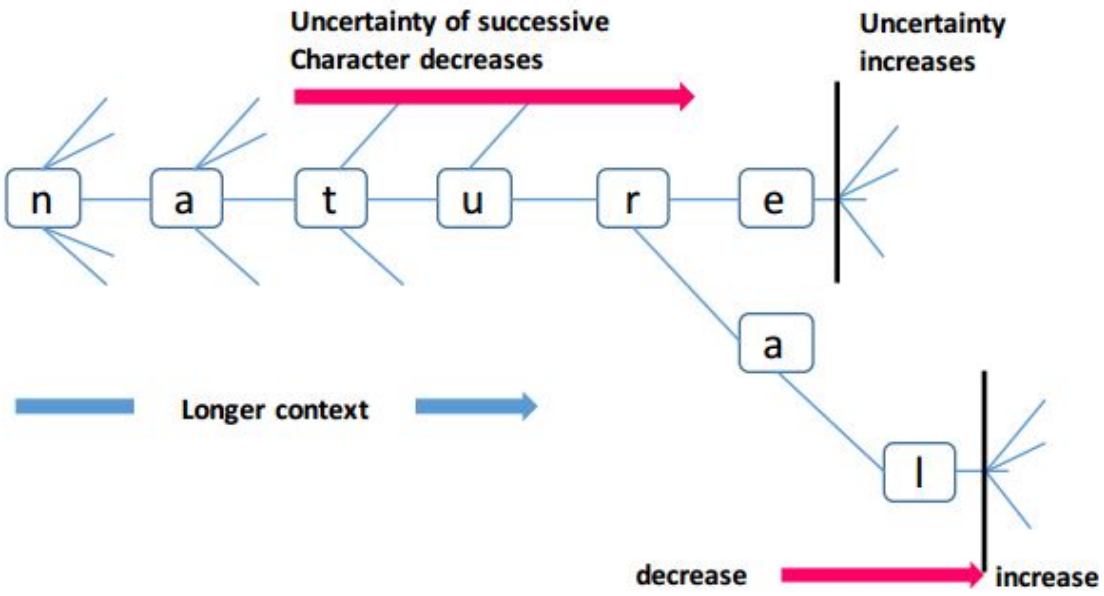
  문자열을 글자단위로 분리하여 부분문자열(substring)을 만들 때 왼쪽부터 문맥을 증가시키면서 각 문맥이 주어졌을 때 그 다음 글자가 나올 확률을 계산하여 누적곱을 한 값이다.

$$cohesion(c_1, c_2, \ldots, c_n) = \sqrt[n-1]{\prod_{i=1}^{n-1} P(c_1, \ldots, c_{i+1} | c_1, \ldots c_i)}$$

$$P(c_1 c_2 | c_1) = \frac{\#c_1 c_2}{\#c_1}$$

- 브랜칭 엔트로피

$$H(X\,|X_n) = \sum_{x \in X} P(x\,|x_n) * \log(\,P(x\,|x_n)\,)$$



3.3.2. 구글 센텐스피스(sentencepiece)

- 구글에서 공개한 비지도 학습 기반 형태소 분석 패키지
- 바이트 페어 인코딩(BPE, Byte Pair Encoding)등을 지원한다.
- BPE : 말뭉치에서 가장 많이 등장한 문자열을 병합해 문자열을 압축하는 것
- 단어는 의미를 가진 더 작은 서브워드들의 조합으로 이루어진다. 빈도수에 따라 문자를 병합하여 subword를 구성한다.
- 예를 들면 "conference" -> 'con', 'f', 'er', 'ence' 로 분절 할 수 있다.
- 한국어, 일본어, 영어 등 언어 형식에 구애받지 않고, 별도의 문법 입력 없이 조사 구분이 가능하다.
- 어휘에서 자유롭게 학습이 가능하고, Rare word를 위한 back-off model이 필요 없다. 그리고 성능 향상 효과도 있다.
- 서브워드 단위로 쪼개면 차원과 sparsiry도 줄일 수 있다. 그리고 처음본 단어(unknown word) 처리에 효과적이다.
- 또한 BPE의 성능개선 방법으로 BPE-DROPOUT이 있다.

3.3.3. 띄어쓰기 교정

- soynlp 형태소 분석이나 BPE 방식의 토크나이즈 기법은 띄어쓰기에 따라 분석 결과가 크게 달라짐 → 모델을 학습하기 전 띄어쓰기 교정을 적용하면 분석 품질을 개선할수 있음
- soynlp의 CountSpace 모듈을 주로 사용함

3.3.4 형태소 분석 완료된 데이터 다운로드

- 특정 데이터셋들을 이용하여 형태소 분석된 데이터셋을 다운로드 할 수 있음.

3.4 이 장의 요약

1. 임베딩 학습용 말뭉치는 라인 하나가 문서면 좋다.
2. 지도 학습 기반의 형태소 분석 모델은 문자열이 주어졌을 때 사람이 알려준 정답 패턴에 최대한 가깝게 토크나이즈한다. ex) KoNLPy, Khaiii
3. 비지도 학습 기반의 형태소 분석 모델은 데이터의 패턴을 스스로 학습하게 함으로써 형태소를 나누는 기법이다. 자주 나오는 단어의 빈도수를 파악해 이를 형태소로 인식한다. ex) soynlp, google sentencepiece