NLP 스터디 3주차

한국어 임베딩

4.2 Word2Vec

4.3 FastText

이영현

실습 출처: https://wikidocs.net/50739

Word2Vec 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
import urllib.request
from konlpy.tag import Okt
from gensim.models.word2vec import Word2Vec
import pandas as pd
import matplotlib.pyplot as plt
```

```
urllib.request.urlretrieve("https://raw.githubusercontent.com/e9t/nsmc/master/ratings.txt", filename="ratings.txt")
```

```
('ratings.txt', <http.client.HTTPMessage at 0x21011562888>)
```

```
train_data = pd.read_table('ratings.txt')
```

```
train_data[:5]
```

	id	document	label
0	8112052	어릴때보고 지금다시봐도 재밌어요ㅋㅋ	1
1	8132799	디자인을 배우는 학생으로, 외국디자이너와 그들이 일군 전통을 통해 발전해가는 문화산...	1
2	4655635	폴리스스토리 시리즈는 1부터 뉴까지 버릴께 하나도 없음.. 최고.	1
3	9251303	와.. 연기가 진짜 개쩔구나.. 지루할거라고 생각했는데 몰입해서 봤다.. 그래 이런...	1
4	10067386	안개 자욱한 밤하늘에 떠 있는 초승달 같은 영화.	1

```
print(len(train_data))
```

```
200000
```

Word2Vec 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
print(train_data.isnull().values.any())
```

True

```
train_data = train_data.dropna(how = 'any') # Null 값이 존재하는 행 제거
print(train_data.isnull().values.any()) # Null 값이 존재하는지 확인
```

False

```
print(len(train_data)) # 리뷰 개수 출력
```

199992

```
# 정규 표현식을 통한 한글 외 문자 제거
train_data['document'] = train_data['document'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]","")
```

```
train_data[:5] # 상위 5개 출력
```

	id	document	label
0	8112052	어릴때보고 지금다시봐도 재밌어요ㅋㅋ	1
1	8132799	디자인을 배우는 학생으로 외국디자이너와 그들이 일군 전통을 통해 발전해가는 문화산업...	1
2	4655635	폴리스스토리 시리즈는 부터 뉴까지 버릴께 하나도 없음 최고	1
3	9251303	와 연기가 진짜 개쩔구나 지루할거라고 생각했는데 몰입해서 봤다 그래 이런게 진짜 영화지	1
4	10067386	안개 자욱한 밤하늘에 떠 있는 초승달 같은 영화	1

Word2Vec

한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
# 불용어 정의
stopwords = ['의','가','이','은','들','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다']
```

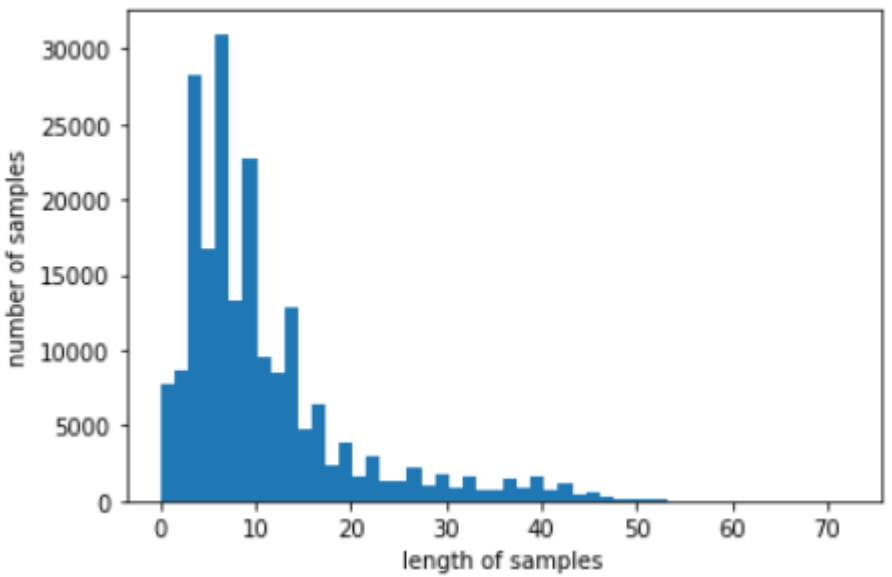
```
# 형태소 분석기 OKT를 사용한 토큰화 작업 (다소 시간 소요)
okt = Okt()
tokenized_data = []
for sentence in train_data['document']:
    temp_X = okt.morphs(sentence, stem=True) # 토큰화
    temp_X = [word for word in temp_X if not word in stopwords] # 불용어 제거
    tokenized_data.append(temp_X)
```

```
# 리뷰 길이 분포 확인
print('리뷰의 최대 길이 :',max(len(l) for l in tokenized_data))
print('리뷰의 평균 길이 :',sum(map(len, tokenized_data))/len(tokenized_data))
plt.hist([len(s) for s in tokenized_data], bins=50)
plt.xlabel('length of samples')
plt.ylabel('number of samples')
plt.show()
```

```
리뷰의 최대 길이 : 72
리뷰의 평균 길이 : 10.716703668146726
```

형태소 분석기

- Kkma
- Komoran
- Hannanum
- Okt(Twitter)
- Mecab

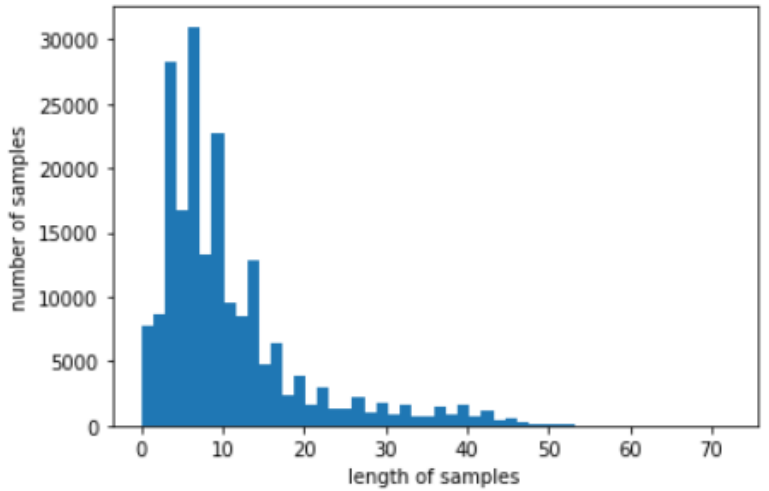


Word2Vec 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
리뷰의 최대 길이 : 72
리뷰의 평균 길이 : 10.716703668146726
```

Tokenized_data

```
['폴리스스토리', '시리즈', '부터', '뉴', '까지', '버리다', '하나', '없다', '최고'],
['오다',
 '연기',
 '진짜',
 '개',
 '쩔다',
 '지루하다',
 '생각',
 '몰입',
 '보다',
 '그렇다',
 '이렇다',
 '진짜',
 '영화',
 '지'],
['안개', '자욱하다', '밤하늘', '뜨다', '있다', '초승달', '같다', '영화'],
['사랑', '을', '해보다', '사람', '라면', '처음', '부터', '끝', '까지', '웃다', '있다', '영화'],
['완전', '감동', '이다', '다시', '보다', '감동'],
['개', '전쟁', '나오다', '나오다', '빠', '로', '보고', '싶다'],
```



Word2Vec 한국어 Word2Vec 만들기(네이버 영화 리뷰)

cbow로 학습

```
from gensim.models import Word2Vec
model = Word2Vec(sentences = tokenized_data, size = 100, window = 5, min_count = 5, workers = 4, sg = 0)
```

```
# 완성된 임베딩 매트릭스의 크기 확인
model.wv.vectors.shape
```

```
(16477, 100)
```

```
print(model.wv.most_similar("최민식"))
```

```
[('한석규', 0.873503565788269), ('안성기', 0.8648371696472168), ('김명민', 0.8608554601669312), ('이민호', 0.8539645671844482), ('최민수', 0.8407468795776367), ('이정재', 0.839943110942840 6), ('박중훈', 0.8386502265930176), ('주진모', 0.8385392427444458), ('송강호', 0.8357796669006348), ('김수현', 0.8341250419616699)]
```

```
print(model.wv.most_similar("히어로"))
```

```
[('슬래셔', 0.8649497032165527), ('무협', 0.8469994068145752), ('호러', 0.8469862937927246), ('느와르', 0.846731960773468), ('물의', 0.8204612731933594), ('정통', 0.8196438550949097), ('블록버스터', 0.8186239004135132), ('무비', 0.8148091435432434), ('물', 0.8108745813369751), ('블랙', 0.7842040061950684)]
```

```
print(model.wv.most_similar("마블"))
```

```
[('어드벤쳐', 0.8466544151306152), ('스타워즈', 0.8439940810203552), ('마스터피스', 0.8426139950752258), ('단언컨대', 0.8371542692184448), ('여고괴담', 0.8286738991737366), ('동방불패', 0.8125902414321899), ('원조', 0.8106462955474854), ('베끼기', 0.8069692850112915), ('괴수영화', 0.8052213191986084), ('지존', 0.8019837141036987)]
```

```
print(model.wv.most_similar("대박"))
```

```
[('안습', 0.7545077204704285), ('짱', 0.7527296543121338), ('꿀잼', 0.7228602170944214), ('노답', 0.6986998319625854), ('우와', 0.6787211894989014), ('헷', 0.6540430188179016), ('ㄷㄷ', 0.649850070476532), ('획오', 0.6456347703933716), ('개판', 0.6380383968353271), ('쩐다', 0.6312457323074341)]
```

```
print(model.wv.most_similar("재미있다"))
```

```
[('재밌다', 0.9223928451538086), ('재다', 0.8279247283935547), ('실망하다', 0.6278233528137207), ('유익하다', 0.5759900808334351), ('괜찮다', 0.5607717037200928), ('흥미진진', 0.5462271571159363), ('재미없다', 0.5452117323875427), ('오래되다', 0.5313775539398193), ('흥미롭다', 0.5304207801818848), ('낫다', 0.527606010437011 7)]
```

Word2Vec

한국어 Word2Vec 만들기(네이버 영화 리뷰)

Skip- gram으로 학습

```
from gensim.models import Word2Vec
model = Word2Vec(sentences = tokenized_data, size = 100, window = 5, min_count = 5, workers = 4, sg = 1)
```

```
# 완성된 임베딩 매트릭스의 크기 확인
model.wv.vectors.shape
```

```
(16477, 100)
```

```
print(model.wv.most_similar("최민식"))
```

```
[('김명민', 0.8273282051086426), ('서영희', 0.8232049942016602), ('안성기', 0.8180164098739624), ('한석규', 0.8143938779830933), ('설경구', 0.7993581295013428), ('이정재', 0.7936372756958008), ('문소리', 0.7922838926315308), ('정려원', 0.7890018820762634), ('유다인', 0.7885781526565552), ('조재현', 0.7867942452430725)]
```

```
print(model.wv.most_similar("히어로"))
```

```
[('슬래셔', 0.8244971036911011), ('첩보물', 0.8152480721473694), ('서유기', 0.7968639135360718), ('마블', 0.7899734973907471), ('괴수', 0.7850511074066162), ('갱스터', 0.7833945751190186), ('다이하드', 0.7805483341217041), ('로맨스코미디', 0.7781327366828918), ('피조', 0.7750976085662842), ('러시아워', 0.7740487456321716)]
```

```
print(model.wv.most_similar("마블"))
```

```
[('에반게리온', 0.8409952521324158), ('러시아워', 0.8194149732589722), ('캐리비안', 0.817896842956543), ('다이하드', 0.8109111785888672), ('스타워즈', 0.7988831996917725), ('로맨스코미디', 0.7915617227554321), ('트릴로지', 0.7913472652435303), ('히어로', 0.7899736166000366), ('여고괴담', 0.7884902954101562), ('으음', 0.7833701372146606)]
```

```
print(model.wv.most_similar("대박"))
```

```
[('헷', 0.7537832260131836), ('짱', 0.7245103120803833), ('우와', 0.7201640605926514), ('쩔어요', 0.7193781137466431), ('존잼', 0.7190908193588257), ('개꿀잼', 0.7175739407539368), ('꿀잼', 0.7063623666763306), ('쩌러', 0.7047148942947388), ('개잼', 0.7017240524291992), ('핵꿀잼', 0.7011761665344238)]
```

```
print(model.wv.most_similar("재미있다"))
```

```
[('재밌다', 0.9195533990859985), ('재다', 0.8599737882614136), ('재미나', 0.7379024028778076), ('잼남', 0.7157114744186401), ('재밋음', 0.6982831954956055), ('재밋', 0.6961558461189271), ('재밋는', 0.6960072517395020), ('였음', 0.6866178512573242), ('요런', 0.6800719499588013), ('재밋엇음', 0.6796712875366211)]
```

```
print(model.wv.most_similar("초딩"))
```

```
[('초등학생', 0.8425231575965881), ('중딩', 0.8057160973548889), ('중학생', 0.7890909314155579), ('유치원', 0.7768651247024536), ('초등학교', 0.7741670608520508), ('저학년', 0.7631106972694397), ('학예회', 0.7380115985870361), ('고딩', 0.7309287786483765), ('학년', 0.7238266468048096), ('중학교', 0.6996942758560181)]
```

FastText 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
from gensim.models import FastText
ft_model = FastText(sentences = tokenized_data, size=100, window=5, min_count=5, workers=4, sg=1)
```

```
# 완성된 임베딩 매트릭스의 크기 확인
ft_model.wv.vectors.shape
```

```
(16477, 100)
```

```
print(ft_model.wv.most_similar("최민식"))
```

```
[('김명민', 0.839626669883728), ('한석규', 0.8175941705703735), ('안성기', 0.8159623742103577), ('최민수', 0.8121828436851501), ('김창완', 0.8100709915161133), ('서영희', 0.7955340147018433), ('조재현', 0.793571412563324), ('임원희', 0.7900139093399048), ('이정재', 0.7898099422454834), ('윤제문', 0.788894534111023)]
```

```
print(ft_model.wv.most_similar("히어로"))
```

```
[('슈퍼히어로', 0.821395754814148), ('첩보물', 0.799573540687561), ('어로', 0.7904710173606873), ('마블', 0.779258668422699), ('슬래셔', 0.7784781455993652), ('다이하드', 0.7767077088356018), ('피조', 0.7598381042480469), ('오컬트', 0.758699893951416), ('첩보', 0.7574349641799927), ('첩혈쌍웅', 0.7564233541488647)]
```

```
print(ft_model.wv.most_similar("마블"))
```

```
[('명탐정', 0.7980530261993408), ('에반게리온', 0.7883397340774536), ('명탐정코난', 0.7871848344802856), ('여고괴담', 0.7823944091796875), ('슈퍼히어로', 0.7802602648735046), ('히어로', 0.7792586088180542), ('스타워즈', 0.7742405533790588), ('나니아', 0.7728855609893799), ('엑소시즘', 0.7650706171989441), ('캐리비안', 0.7646437883377075)]
```

```
print(ft_model.wv.most_similar("로다주"))
```

```
[('윤상현', 0.9263811111450195), ('괘', 0.9153809547424316), ('박형식', 0.9144102334976196), ('채정안', 0.9130532145500183), ('고천락', 0.9114694595336914), ('이홍기', 0.9113540053367615), ('김정태', 0.9110448360443115), ('고경표', 0.9089635610580444), ('정민', 0.9069292545318604), ('현준', 0.9055861234664917)]
```

FastText VS Word2Vec 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
print(ft_model.wv.most_similar("초딩영화"))
```

```
[('유치원', 0.5538469552993774), ('초딩', 0.5463035702705383), ('후레쉬맨', 0.5253959894180298), ('파워레인저', 0.5243077278137207),
('중딩', 0.513674259185791), ('어렷을때', 0.5085972547531128), ('옛날', 0.5052133798599243), ('초때', 0.503542423248291), ('학예회', 0.
4986870430374146), ('강시선생', 0.4914855360984802)]
```

```
print(model.wv.most_similar("초딩영화"))
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-67-6f3557e1dbfe> in <module>
----> 1 print(model.wv.most_similar("초딩영화"))

~₩Anaconda3₩lib₩site-packages₩gensim₩models₩keyedvectors.py in most_similar(self, positive, negative, topn, restrict_voca
b, indexer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_norm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~₩Anaconda3₩lib₩site-packages₩gensim₩models₩keyedvectors.py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word)
    469
    470     def get_vector(self, word):

KeyError: "word '초딩영화' not in vocabulary"
```

Word2Vec은 학습하지 않은 단어에 대해서 유사한 단어를 찾아내지 못 했지만, 패스트텍스트는 유사한 단어를 계산해서 출력하고 있음을 볼 수 있다.

FastText VS Word2Vec

한국어 Word2Vec 만들기(네이버 영화 리뷰)

Word2Vec-cbow

```
print(model.wv.most_similar("대박"))
```

```
[('안습', 0.7545077204704285), ('짱', 0.7527296543121338), ('꿀잼', 0.7228602170944214), ('노답', 0.6986998319625854), ('우와', 0.67872
11894989014), ('헷', 0.6540430188179016), ('ㄷㄷ', 0.649850070476532), ('획오', 0.6456347703933716), ('개판', 0.6380383968353271), ('쩐
다', 0.6312457323074341)]
```

Word2Vec-skip gram

```
print(model.wv.most_similar("대박"))
```

```
[('헷', 0.7537832260131836), ('짱', 0.7245103120803833), ('우와', 0.7201640605926514), ('쩔어요', 0.7193781137466431), ('존잼', 0.71909
08193588257), ('개꿀잼', 0.7175739407539368), ('꿀잼', 0.7063623666763306), ('쩌러', 0.7047148942947388), ('개잼', 0.7017240524291992),
('핵꿀잼', 0.7011761665344238)]
```

FastText-skip gram

```
print(ft_model.wv.most_similar("대박"))
```

```
[('핵꿀잼', 0.7239328622817993), ('우와', 0.7202774286270142), ('개꿀잼', 0.7202334403991699), ('헷', 0.6935589909553528), ('짱', 0.691
777765750885), ('개꿀', 0.6907384395599365), ('존잼', 0.6900402307510376), ('와우', 0.677210807800293), ('차태현', 0.6761030554771423),
('뱀검', 0.6711919903755188)]
```

FastText 모델의 강점

조사나 어미가 발달한 한국어에 종은 성능을 보임

용언(동사, 형용사)의 활용이나 그와 관계된 어미들이 벡터 공간상 가깝게 임베딩 됨

하였다(타깃 단어(t)), 진행(문맥 단어의 포지티브 샘플(c))

〈하였, 하였다, 였다〉 벡터(z)들 각각이 진행에 해당하는 벡터(v_c)와의 유사도가 높아짐

하였다 벡터와 하(다), 했(다), (하)였으며 등에 해당하는 벡터간 유사도가 높을 것
-〉 문맥이 서로 비슷하므로

FastText 모델의 강점

오타나 미등록 단어에도 로버스트함

-〉 각 단어의 임베딩을 문자 단위 n-gram 벡터의 합으로 표현하므로

서울특별시

서울 등이 문자 단위 n-gram에 포함

울특, 특벌 등이 모두 미등록 단어라 할지라도 서울특별시에 대한 임베딩 추정 가능

FastText 한국어 Word2Vec 만들기(네이버 영화 리뷰)

```
print(ft_model.wv.most_similar("하였다"))
```

```
[('봤', 0.3136533498764038), ('방금', 0.3124797344207763 7), ('보고오다', 0.30907753109931946), ('낚였', 0.29798755049705505), ('낫', 0.28990170359611 51), ('인터넷', 0.2876492142677307), ('첨으로', 0.28617802262306213), ('봤는데', 0.2835083305835724), ('낚였다', 0.28280770778656006), ('내생', 0.279556006193161)]
```

```
print(ft_model.wv.most_similar("재밌다"))
```

```
[('재미있다', 0.9148136377334595), ('재다', 0.8783955574035645), ('재밋는듯', 0.8098613023757935), ('재밋긴', 0.8026115894317627), ('재밋는', 0.7969444394111633), ('재밋슴', 0.7921749353408813), ('재밋기', 0.7798007130622864), ('재미나다', 0.7761648893356323), ('재밋게봄', 0.7740651369094849), ('재밋더', 0.7734447717666626)]
```

```
print(ft_model.wv.most_similar("재미있다"))
```

```
[('재밌다', 0.9148136377334595), ('재다', 0.8236192464828491), ('재미나다', 0.7759655714035034), ('재밋는듯', 0.7615798711776733), ('재밋긴', 0.7533753514289856), ('재밋는', 0.7462509870529175), ('재밋슴', 0.7458091378211975), ('재밋게봄', 0.7334438562393188), ('재밋엇음', 0.7273903489112854), ('재밋더', 0.7270290851593018)]
```

```
print(ft_model.wv.most_similar("자미있다"))
```

```
[('재미있다', 0.6404945850372314), ('재밌다', 0.5660013556480408), ('재다', 0.5070028305053711), ('재미나', 0.5031919479370117), ('재미나다', 0.4983797073364258), ('슈렉', 0.487751245498657 2), ('재밋긴', 0.4862810969352722), ('재밋게봄', 0.483900755643844 6), ('재밋는듯', 0.4728877544403076), ('평타', 0.46678829193115234)]
```

프로젝트 주제

- (딥러닝 기반 댓글 필터링) http://www.datamarket.kr/xe/index.php?mid=board_pdzw77&page=2&document_srl=44636
- (올 한해 핫 이슈 요약 및 네트워크도를 통한 시각화) http://www.datamarket.kr/xe/index.php?mid=board_pdzw77&page=2&document_srl=37770
- (DrQA, NER, 유사도 기반 지식백과 QA봇) http://www.datamarket.kr/xe/board_pdzw77/50315
- (딥러닝을 이용한 수능 영어영역 풀이) http://www.datamarket.kr/xe/board_pdzw77/50240