



Natural Language Processing
& Artificial Intelligence

Bilingual Word Embedding



INDEX

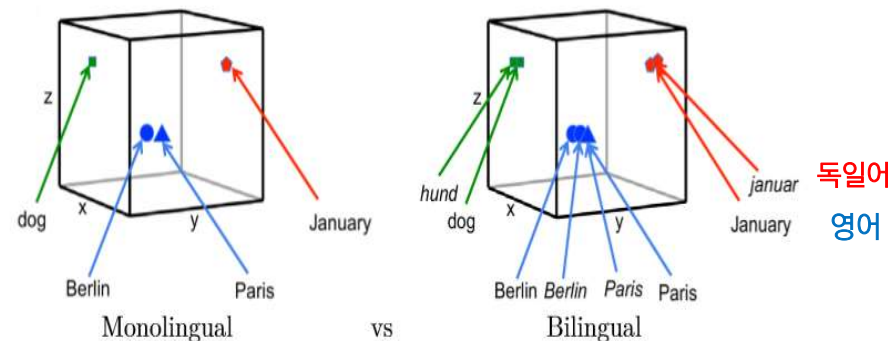
- ◆ Overview of Bilingual Word Embedding
- ◆ Overview of implementation
- ◆ 실습
- ◆ Visualization



Overview of Bilingual Word Embedding

Overview of Bilingual Word Embedding

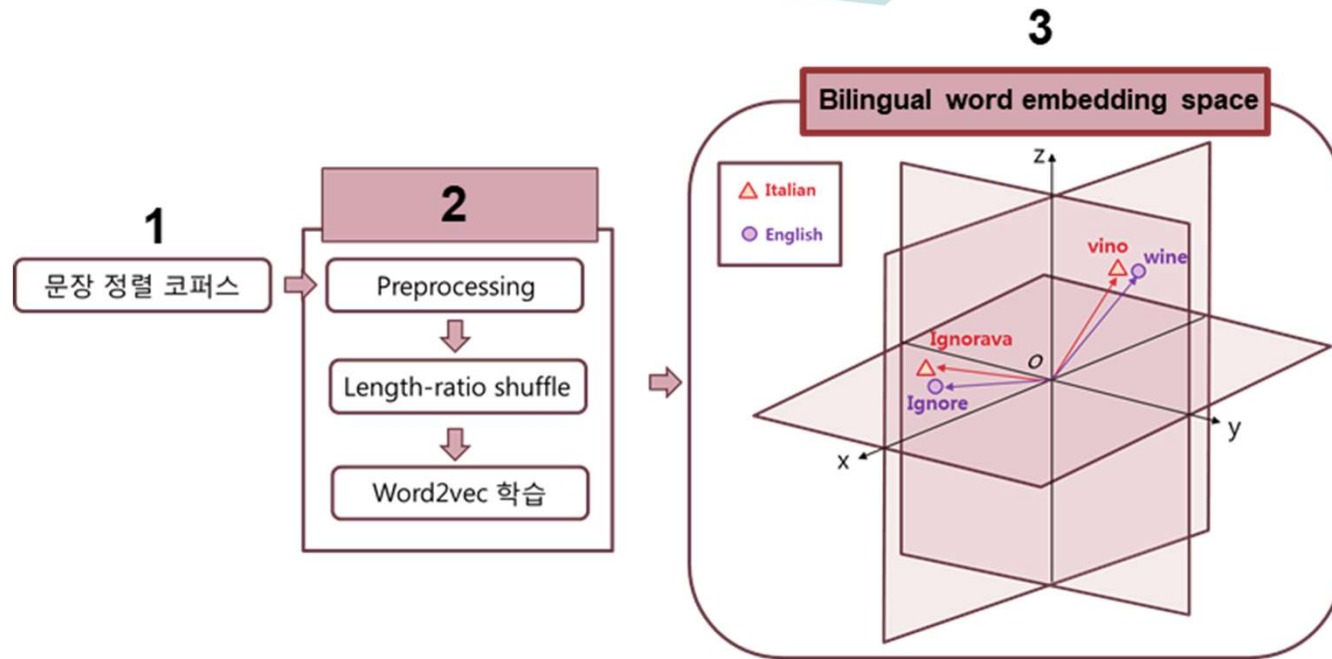
- ◆ 단일 언어(한국어)에 대한 word embedding을 통해 단어 간의 의미를 파악하였다면, 서로 다른 두 언어를 embedding 하는 것은 어떨까?
- ◆ Bilingual Word Embedding : 두 개의 다른 언어(bilingual)에서 유사한 의미의 단어는 유사한 공간에 있도록 하나의 벡터공간으로 단어를 임베딩





Overview of implementation

Overview of implementation





실습

Dataset 및 코드 구성 개요

- ◆ 실험 환경

- ◆ Google colab 

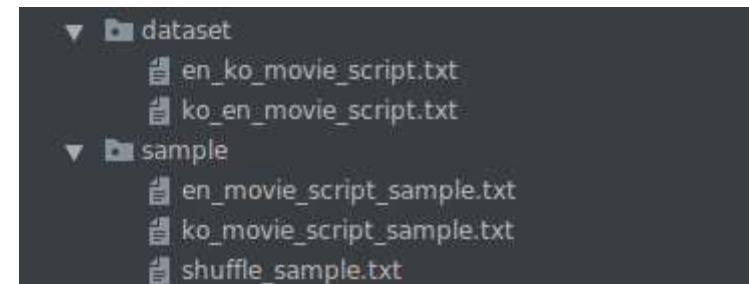
- ◆ Dataset

- ◆ Sample dataset (data snippet)
 - ◆ Full dataset

- ◆ 코드

- ◆ Preprocessing_skeleton.ipynb
 - ◆ Train_skeleton.ipynb
 - ◆ Gensim_vis.ipynb

[Dataset]



Dataset 다운로드

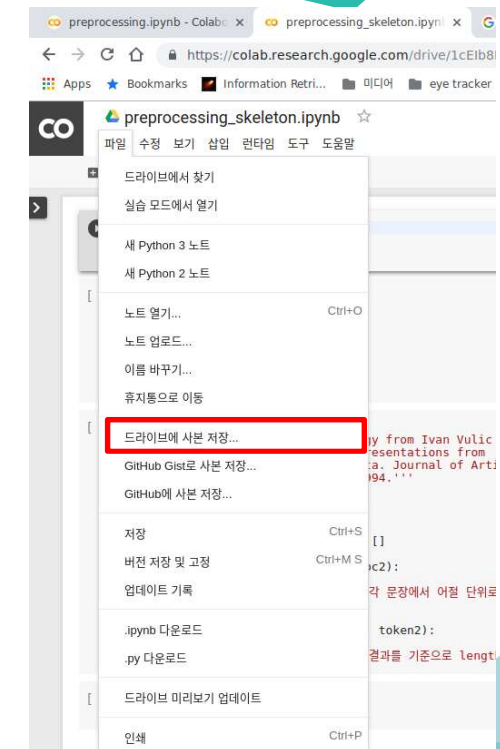
- ◆ Dataset 다운로드
- ◆ Full dataset

[Dataset]

```
▼ dataset
  en_ko_movie_script.txt
  ko_en_movie_script.txt
▼ sample
  en_movie_script_sample.txt
  ko_movie_script_sample.txt
  shuffle_sample.txt
```

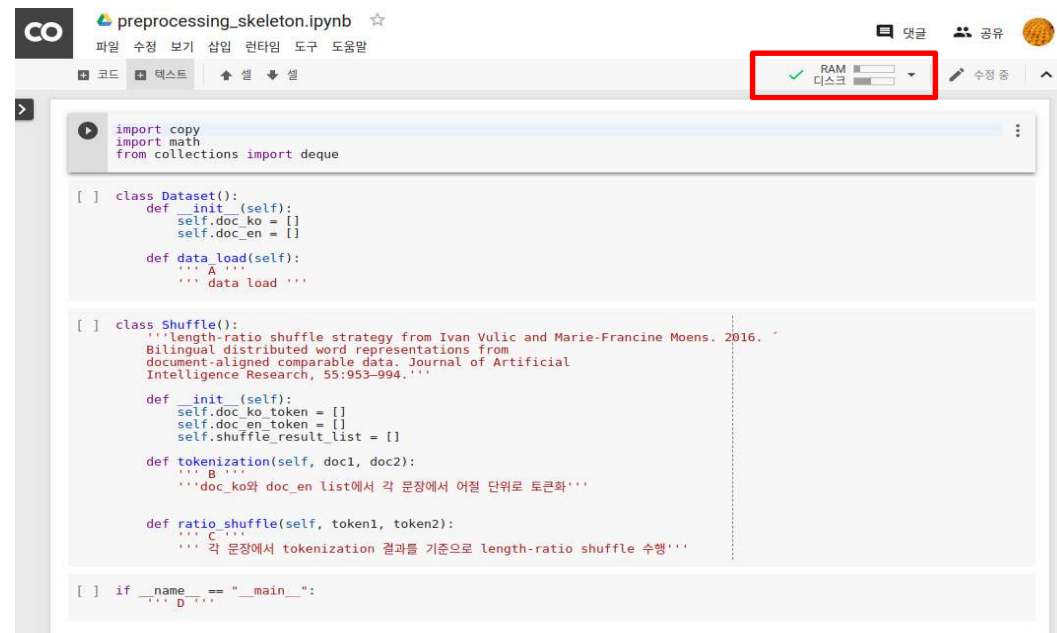
코드 다운로드

- ◆ Skeleton 코드
- ◆ Skeleton코드 url에 접속 후, 자신의 구글 계정 드라이브에 사본 저장 (파일-> 드라이브에 사본저장)



코드 사용 준비 체크

- ◆ 오른쪽 상단에 RAM과 디스크 사용 정보가 떠있으면 코드 수정 및 실행 가능상태



The screenshot shows a Jupyter Notebook titled 'preprocessing_skeleton.ipynb'. The interface includes a top bar with icons for file, edit, view, insert, terminal, and help. Below this is a toolbar with tabs for '코드' (Code), '텍스트' (Text), and '셀' (Cell). A red box highlights the top right corner, showing the 'RAM' and '디스크' (Disk) usage indicators, both with green checkmarks, indicating that the code is ready for execution. The main area displays Python code for a 'Dataset' and 'Shuffle' class. The 'Dataset' class has an 'init' method and a 'data_load' method. The 'Shuffle' class has an 'init' method, a 'tokenization' method, and a 'ratio_shuffle' method. The code is as follows:

```
[ ] class Dataset():
    def __init__(self):
        self.doc_ko = []
        self.doc_en = []

    def data_load(self):
        ''' A '''
        data load '''

[ ] class Shuffle():
    '''length-ratio shuffle strategy from Ivan Vulic and Marie-Francine Moens. 2016. "
    Bilingual distributed word representations from
    document-aligned comparable data, Journal of Artificial
    Intelligence Research, 55:953-994.'''

    def __init__(self):
        self.doc_ko_token = []
        self.doc_en_token = []
        self.shuffle_result_list = []

    def tokenization(self, doc1, doc2):
        ''' B '''
        '''doc_ko와 doc_en list에서 각 문장에서 어절 단위로 토큰화'''

    def ratio_shuffle(self, token1, token2):
        ''' C '''
        ''' 각 문장에서 tokenization 결과를 기준으로 length-ratio shuffle 수행'''

[ ] if __name__ == "__main__":
    D
```

문장정렬 코퍼스

◆ OPUS2016

◆ 영화 자막 코퍼스, 33만건

1	폭설 이 내리 고 우박 진눈깨비 가 퍼붓 어도 눈보라
2	황새 아저씨 가 는 길 을 그 누구 가 막 으라
3	황새 아저씨 를 기다리 시 어요
4	찾아오 아 선물 을 주 시 ㄹ 거 예 요
5	가난 하 든 부자 이 든 상관 이 없 답니다
6	백만장자 도 하나 가난뱅이 도 하나
7	황새 아저씨 를 기다리 시 어요
8	도망치 어도 소용없 어요 반드시 찾아내 니까요
9	세상 끝 에 있 어도 하늘 꼭대기 에 있 어도
10	황새 아저씨 는 찾아가 ㄴ답니다
11	황새 아저씨 를 기다리 시 어요
12	한 번 명단 에 오르 면 실 어도 하 ㄹ 수 없 어요
13	신발 속 의 쌍둥이 애기 들 으시 었 조
14	다음 차례 는 당신 이 ㄹ지 도 모르 아요
15	승차 완료

1	through the snow and sleet and hail through
2	he'll get through
3	look out for mr stork that persevering cha
4	he'll come along and drop a bundle in your
5	you may be poor or rich it doesn't matter
6	millionaires they get theirs like the butc
7	so look out for mr stork and let me tell y
8	don't try to get away he'll find you in th
9	he'll spot you out in china or he'll fly t
10	so you'd better look out for mr stork
11	look out for mr stork he's got you on his
12	and when he comes around it's useless to r
13	remember those quintuplets and the woman i
14	maybe he's got his eye on you
15	all aboard all aboard

한국어(형태소 분석 완료 버전)

영어



Preprocessing_skeleton.ipynb

- ◆ Data load
- ◆ Tokenization
- ◆ Length-ratio shuffle



Tokenization

- ◆ Tokenization
 - ◆ Length-ratio shuffle을 위해 필요
 - ◆ 각 문장에서 어절단위의 토큰화

Length-ratio shuffle (1/4)

- ◆ Length-ratio shuffle concepts
 - ◆ 각 문장별 token수의 비율로 shuffle
 - ◆ 예) ko = {포도, 오렌지}, Token number = 2
 - ◆ 예) en = {carrot, apple, pineapple, egg}, Token number = 4
 - ◆ Ratio = 1/2
 - ◆ Result = {포도, carrot, apple, 오렌지, pineapple, egg}

Length-ratio shuffle (2/4)

[한국어(형태소 분석 완료)]

```
1 폭설 이 내리 고 우박 진눈깨비 가 퍼붓 어도 눈보라
2 황새 아저씨 가 는 길 을 그 누구 가 막 으랴
3 황새 아저씨 를 기다리 시 어요
4 찾아오 아 선물 을 주 시 ㄹ 거 예 요
5 가난 하 든 부자 이 든 상관 이 없 답니다
6 백만장자 도 하나 가난뱅이 도 하나
```

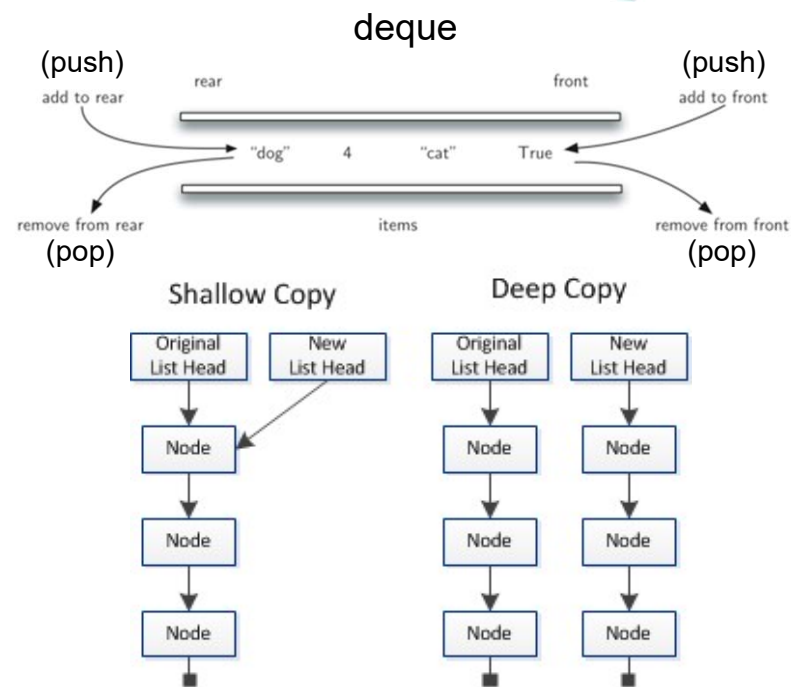
[영어]

```
1 through the snow and sleet and hail through
2 he'll get through
3 look out for mr stork that persevering cha
4 he'll come along and drop a bundle in your
5 you may be poor or rich it doesn't matter
6 millionaires they get theirs like the butc
```

[한국어-영어 length ratio shuffle 결과]

```
1 through/en 폭설/ko the/en snow/en 이/ko and/en sleet/en
2 황새/ko 아저씨/ko 가/ko he'll/en 는/ko 길/ko 을/ko 그/ko
3 look/en 황새/ko out/en 아저씨/ko for/en 를/ko mr/en stork
4 찾아오/ko he'll/en 아/ko come/en 선물/ko along/en 을/ko a
5 가난/ko you/en 하/ko may/en 든/ko be/en 부자/ko poor/en
6 millionaires/en 백만장자/ko they/en get/en 도/ko theirs/
```


Length-ratio shuffle (3/4)



Main

Preprocessing_skeleton.ipynb

```
data = Dataset()
run = Shuffle()

data.data_load()
doc_ko_token, doc_en_token = run.tokenization(data.doc_ko, data.doc_en)
run.ratio_shuffle(doc_ko_token, doc_en_token)

shuffle_result_file = open("ko_en_shuffle.txt", 'w')
shuffle_result_file.write("\n".join(run.shuffle_result_list))
shuffle_result_file.close()
```

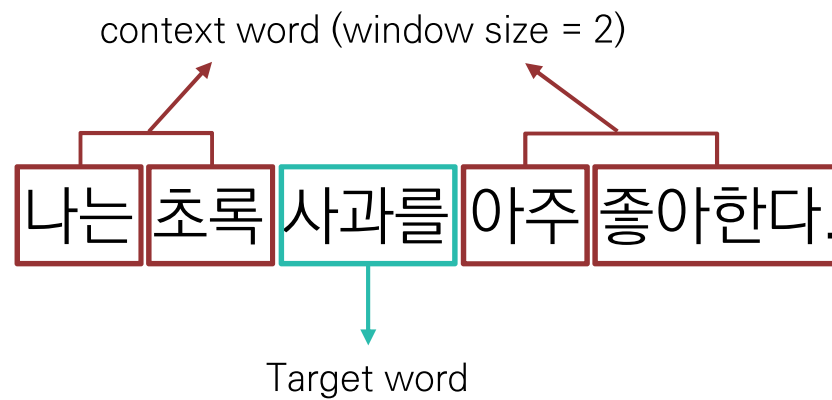


Output

ko_en_shuffle.txt

Window size

- ◆ 한국어와 영어의 어순을 반영하여 window size는 크게!



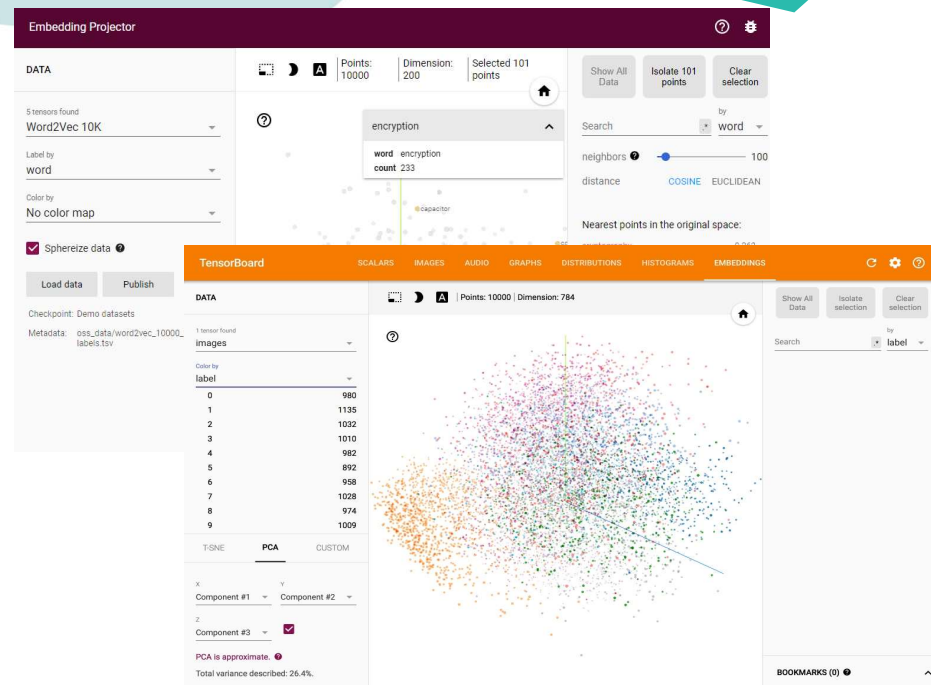


Visualization

Tensorboard

gensim_vis.ipynb

- ◆ Google에서 제공해주는 visualization tool
 - ◆ 온라인상에서 원하는 파일을 업로드하여 시각화하는 방식
 - ◇ Embedding projector (url로 접속)
 - ◆ 로컬 pc내에서 시각화하는 방식
 - ◇ Tensorboard (설치 후 이용가능)



Tensorboard

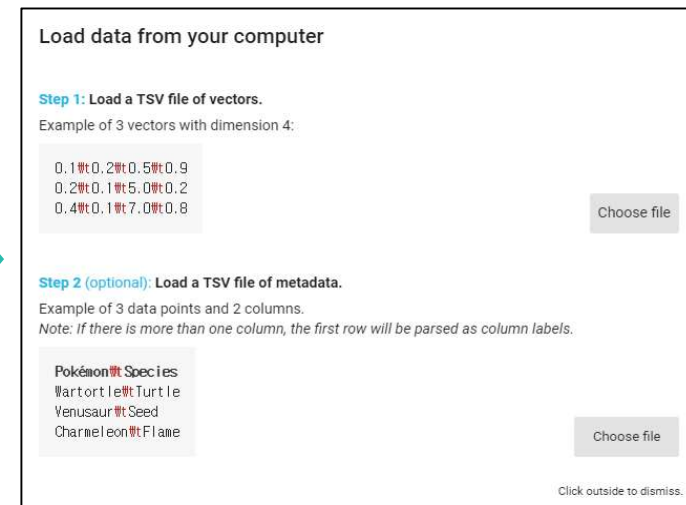
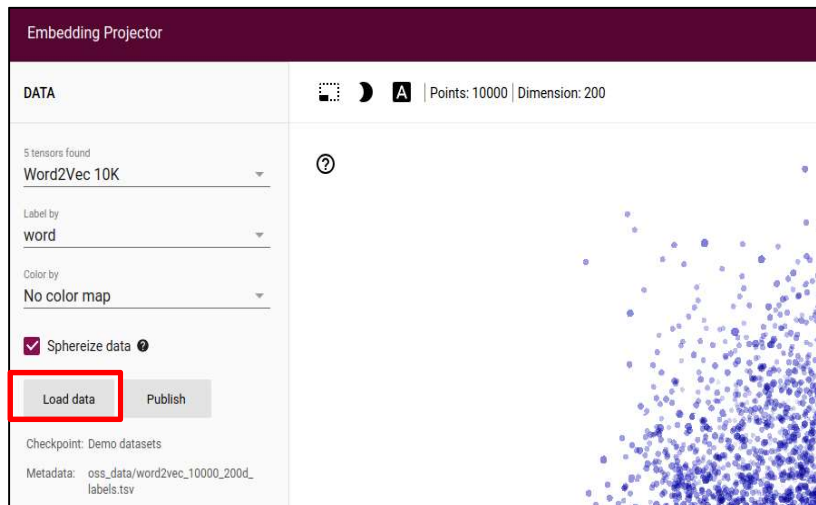
gensim_vis.ipynb

- ◆ 모든 단어를 시각화하기에는 load가 크기때문에 등장 고빈도 단어 순으로 5k개를 시각화함
- ◆ 코드 실행을 위해 필요한 파일
 - ◆ seed_bwe_model
- ◆ 시각화를 위해 해당 코드에서 도출되는 output
 - ◆ Wordvectors (단어 벡터)
 - ◆ Labels (vocabulary)

Tensorboard

gensim_vis.ipynb

- ◆ Tensorboard 에서 시각화 확인을 위해 필요한 파일
 - ◆ Word vectors file (.tsv)
 - ◆ Metadata file (.tsv)

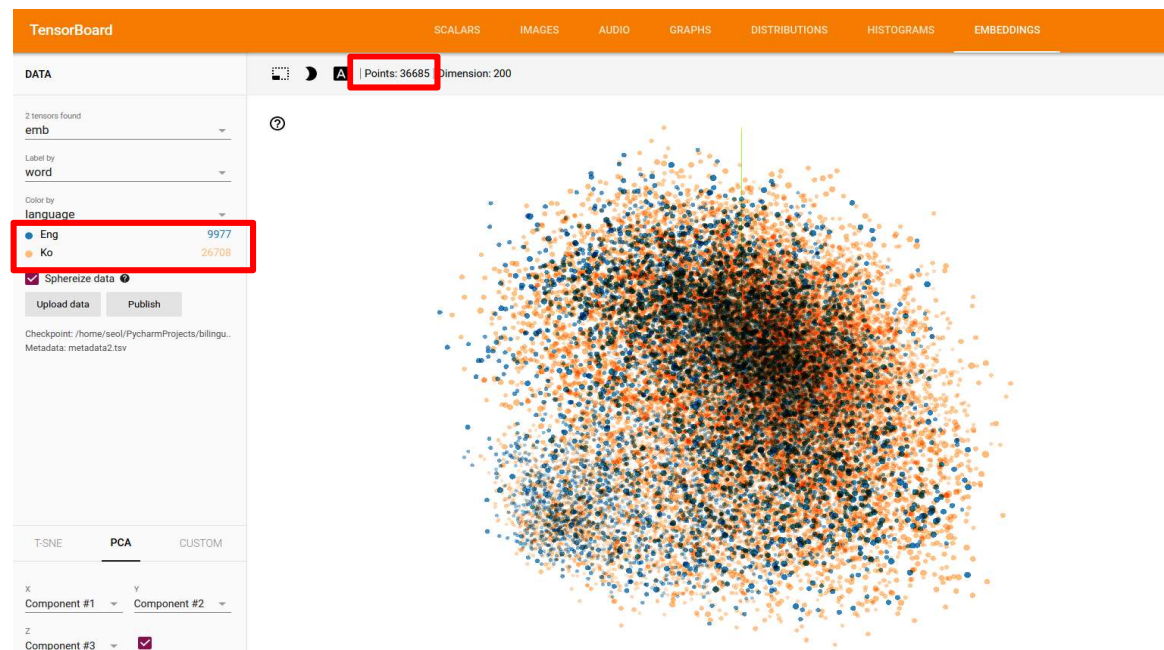


Tensorboard (Embedding projector) 접속

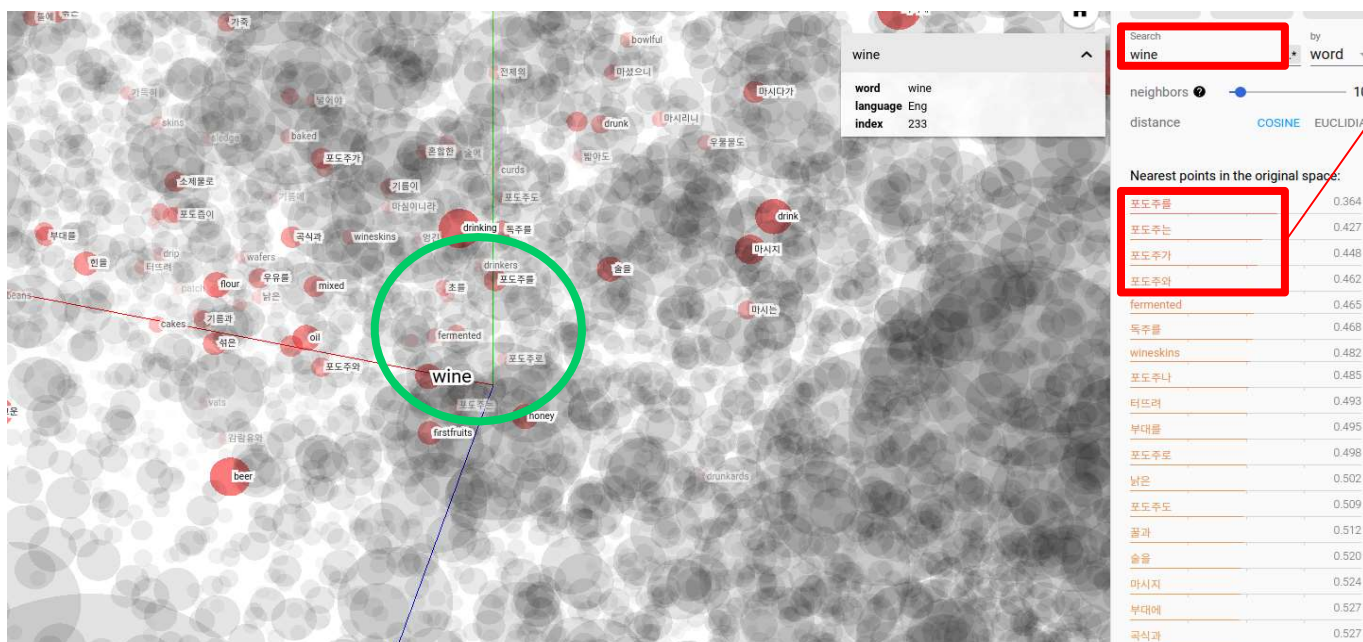
◆ <http://projector.tensorflow.org/>

Example of Visualization (한국어 어절)

- ◆ 형태소 분석없이 어절단위로 학습하였기 때문에 vocabulary 사이즈가 큼



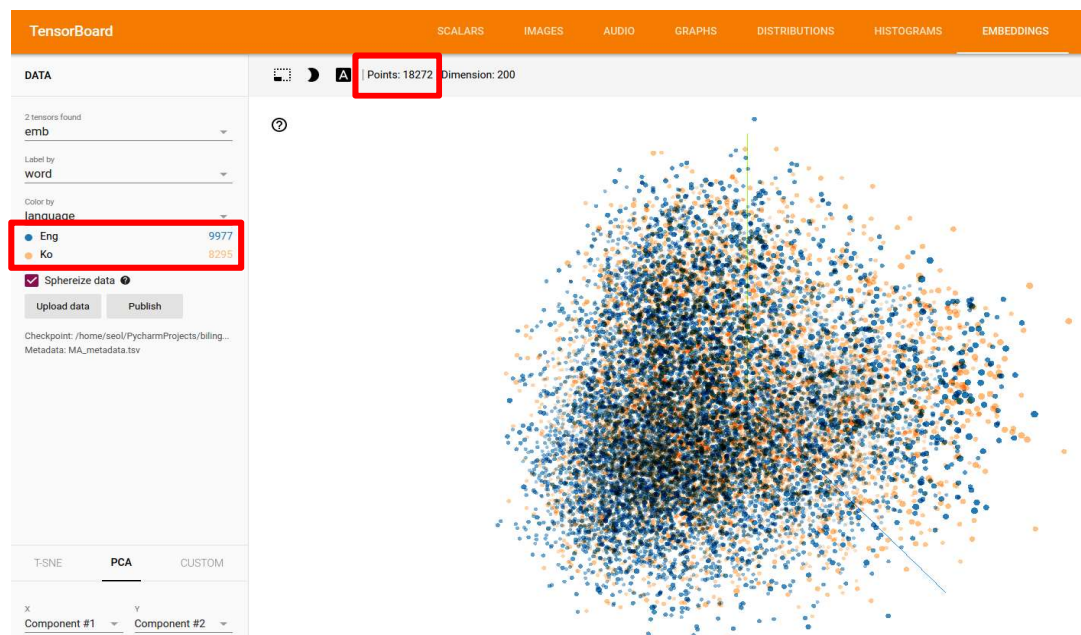
Exaple of Visualization (한국어 어절)



어절 단위의 결과

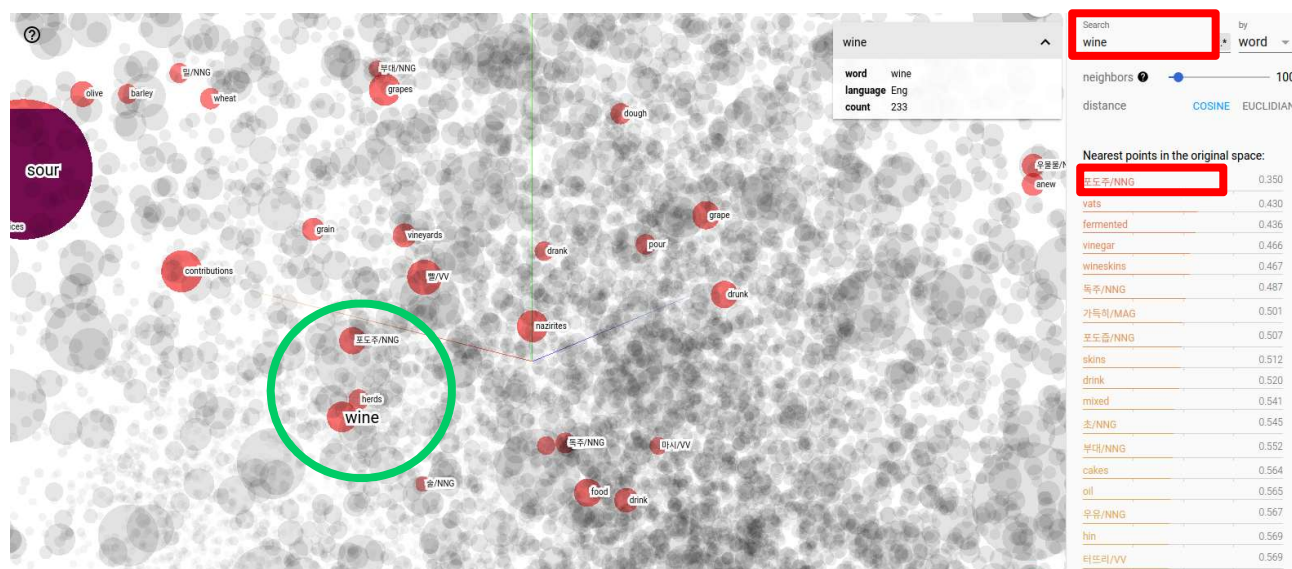
Example of Visualization (한국어 형태소 단위)

- ◆ 어절단위보다는 vocabulary의 수가 줄어듦 (영어와 한국어의 vocab 수가 비등해짐)
=> 골고루 잘 섞임



Example of Visualization (한국어 형태소 단위)

“Wine” 검색 => “포도주”가 가장 유사한 결과로 나옴





Thanks!