Baktash Ansari

Advantages and disadvantages of each model are written at the end of the notebook.

```
import torch
import re
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
import numpy as np
import math
from gensim.models import Word2Vec


cos = torch.nn.CosineSimilarity(dim=0, eps=1e-6)


def find_k_nearest_neighbors(word, embedding_dict, k):
  words_cosine_similarity = dict()
  for token in embedding_dict.keys():
    words_cosine_similarity[token] = cos(embedding_dict[word], embedding_dict[token]).item()
  words_cosine_similarity = dict(sorted(words_cosine_similarity.items(), key=lambda item: item[1]))
  return list(words_cosine_similarity.keys())[-k:][::-1]

def delete_hashtag_usernames(text):
  try:
    result = []
    for word in text.split():
      if word[0] not in ['@', '#']:
        result.append(word)
    return ' '.join(result)
  except:
    return ''

def delete_url(text):
  text = re.sub(r'http\S+', '', text)
  return text

def delete_trash(text) :
  text = re.sub(r'\u200c','',text)
  return text


word = 'زندگی'
k = 10
```

0. Data preprocessing

```
!pip install json-lines
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: json-lines in /usr/local/lib/python3.8/dist-packages (0.5.0)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from json-lines) (1.15.0)
```

```
import json_lines


# 1. extract all tweets from files and save them in memory base on each year.
# 2. remove urls, hashtags and usernames.

frame =  pd.read_csv("/content/mahsa_amini_data.csv")

frame.Text

result = []

for text in frame.Text :
  new_text = delete_hashtag_usernames(text)
  new_text = delete_url(new_text)
  new_text = delete_trash(new_text)
  result.append(new_text)
```

```
result
```

```
 'سرق نداارن به خرای تک تک خریراش بسینه ',
 'وقتی خیابون جا مردم شده پره پلیس انقلاب میشه تنها میانبر گریز ..',
 'شما در دو جهت میتونی به شکل افقی و عمودی',
 'بیماری زمینه ای داشته و در بچگی عمل جراجی',
 'بسلامت برمیگردی',
 'لطفا اطلاع رسانی کنید',
 'اردبیل ۲۳ مهر ۱۴۰۱ کتک خوردن یک بسیجی از مردم ',
 'زنده باد کردستان...',
 'تو روزای عادی هرچی کیر پرت میکنی اخونده که میپره با دندون میگیره تو ببین این روزا یکیشون تخم میکنه نفس بکشه؟ سوراخ موشا
رو بگردید به‌رخودان ژیانه✌',
 'ما هستیم:)',
 'برای آنانیموس که شونه به شونمون داره میجنگه برای خواهرم',
 'برای ایران',
 'برای اینکه رسم ماست، ایستاده مردن.',
 'برای',
 'ایرانو پس میگیریم',
 'چون حس تنفر رو شما به ما یاد دادید. متنفرم. متنفرم. متنفرم.',
 'برای نویسندگان و شخصیت های سیاسی کشته شده در قتل های زنجیره ای دهه ۷۰',
 'ما معترضیم نه اغتشاشگر.',
 'یه حس دوستی قشنگی میبینم بین اونایی که حتی اسمشونم نمیدونم',
 'منم همینطور اولین پارا این همه هشتگ میزنم',
 'اهای اکراین پهبادهایی که به شما حمله کرده ایرانی نیست، جمهوری اسلامیه، ایرانی ماییم، مقتدر، آرمانخواه، آزاده ،انسان
انسان انسان',
 'برای خواهرم',
 'من صد شدم بچه ها هر کسی کم داره پاشه بگه فالو کنم هوای همدیکه رو داشته باشین',
 'حجم تیرندازی در شهرستانها به مراتب بیشتر از تهران است تهران توچشمه می ترسن زیاد بکشند صدای شهرستانها باشیم',
 'بخاطر چشایی ک دوباره مجبور شدم عینک بزنم:(',
 'تیر مستقیم میزنن حروم زاده ها',
 'چقدر هممون شبیه ی خانواده شدیم :))))\U0001fac2💜',
 'برای ایرانم🖤',
 'میگه اسپری فلفل جرم ، خواهر گلم کلا اغتشاشات یه فایل مخصوص داره دیگه شما رو اونجا بگیرن بای دیفالت مجرمی چه با اسپری
فلفل چه بدون اون ... پس یه چی همراهتون باشه برای دفاع از خودتون',
 'بچه ها یه هُلی بدید به ۱۰۰ برسم بک میدم دمتون گرم\U0001fac2',
 'دو',
 'واقعن متاسفم براشون!',
 'واااای واااای از اون صبح لعنتی',
 'برای مهسا (شمت و سه)',
 'برای اینکه جلو بچهای ایندمم برای اینکه کاری نکردم تا زندگیشون بهتر بشه شرمنده نباشم',
 'این همه سال جنایت، مرگ بر این ولایت» شنبه ۱۶ مهرماه ۱۴۰۱» ',
 'برای ایران آزاد',
 'هاها',
 'و راستی تولدت مبارک:)))',
 'منظورشون همون پسرایی که خودشون و میندازن جلو ضربات تا خدایی نکرده به دخترا نخوره؟ کثافت بزنه به اون مغز نجستون اخه
دشمنان ما از تمام توانشون در مقابل ما ایستاده اند و اکنون با حیله هاي کثیف تفرقه افکني در حال سمپاشي هستند، فریب این
تبهکاران را نخوریم.',
 'اگه همه ی تیم ها همینجوری عمل کنند نمیتونن ببرنشون بازداشتگاه',
 'ریدم دهنت که بیشتر متوجه بددهنی ما بشی .رهبر جاکشتم تیکه تیکه میکنیم که بیشتر حجم خشونت ما رو ببینی✌',
 'جدی به کمکتون نیاز دارم میشه ریت کنین فالورام بیشتر بشه هشتگام حساب بشه؟ بکم میدم به همه.',
 'هشت',
 'برای وطنم',
 'خوانندهی آهنگ another love کرده تقدیم ایرانمون رو کنسرتهاش تام ادل، یکی از (= ',
 'بی شرف های مزدور',
 '۶۴. 64 برای. For',
 'برای تمام زنان ایران',
 'این جونهایی که گرفتن بی جواب نمیمونه',
 'برای روزگاری که جز نام آزادی ((تکرار)) نمی کنیم.',
 'برای تمام سالهایی که به حای خدمت کردن مردم رو با هم دشمن کردن، اعتماد رو از بین بردن، تخم نفاق پاشیدن و اخر سر هم شکست
خوردن',
 'برای آزادیمون🖤',
```

1. One hot encoding

```
# 1. find one hot encoding of each word for each year
# 2. find 10 nearest words from "ولنتاین"


new_result = []
for text in result :
  new_result.append(text.split(" "))

category = set()

for li in new_result :
  for text in li :
    category.add(text)
print(len(category))

category = list(category)
```

```
oneHot = np.zeros((len(category),len(category)))

oneHotdic = {}

for i in range(len(category)) :
      oneHot[i,i] = 1
      oneHotdic[category[i]] = torch.tensor(oneHot[i]).float()



find_k_nearest_neighbors(word,oneHotdic,k)
```

```
    32116
    ['زندگی',
     'کوچیکی',
     'آخر',
     'خانوادشون',
     'اینجا',
     'بجای',
     'کرد:)..#OpIran',
     'ترینن',
     'حرامزادهگان',
     'ادامه']
```

2. TF-IDF

```
# 1. find the TF-IDF of all tweets.
# 2. choose one tweets randomly.
# 3. find 10 nearest tweets from chosen tweet.


tf = np.zeros((len(new_result),len(category)))


count = {}

all = 0

for tweet in new_result :
  for word in tweet :
    all += 1
    count[word] = 0

for tweet in new_result :
  for word in tweet :
    count[word] += 1

# saving index of each word in category for reducing Time order
indexWord = {}
for index,word in enumerate(category) :
  indexWord[word] = index

# fill numpy array :
for i,tweet in enumerate(new_result) :

  for word in tweet :

    cnt = 0
    for new_word in tweet :
      if new_word == word :
        cnt +=1
    tf[i,indexWord[word]] = float(cnt)/len(tweet) * math.log(float(all)/count[word])


resultDict = {}

for index,tweet in enumerate(result) :
  resultDict[tweet] = torch.tensor(tf[index])
resultDict

find_k_nearest_neighbors(result[11],resultDict,k)
```

```
    ['برای آزادی ایران تا آخرین قطره خونم میجنگم ✌🏻🇮🇷 (7) ',
     'تا آخرین قطره اینترنت',
```

```
'تا آخرین قطره ی خون',
'برای مردمم میجنگم',
'برای آزادی ایران',
'من امشب خونم رو برای وطنم خواهم داد هر قطره خون من فدای آزادی میهنم',
'برای آزادی ',
'برای ایران ',
'برای ایران برای آزادی',
'تا آزادی']
```

3. Word2Vec

```
# 1. train a word2vec model base on all tweets for each year.
# 2. find 10 nearest words from "ولنتاین"
#category

model = Word2Vec(sentences = new_result)

model.wv.most_similar("آزادی")
```

```
[('0.9874179363250732 ,'ازادی'),
 ('0.9833468794822693 ,'زن'),
 ('0.9817682504653931 ,'زندگی،'),
 ('0.9769082069396973 ,'امید'),
 ('0.9757297039031982 ,'ایران'),
 ('0.9737299680709839 ,'خواهرم'),
 ('0.9733548164367676 ,'زندگی'),
 ('0.9687106609344482 ,'زن،'),
 (',0.96824711561203 ,'زندگی'),
 ('0.9667270183563232 ,'برای')]
```

4. Contextualized embedding

```
# 1. fine tune a bert model base on all tweets for each year.
# 2. find 10 nearest words from "آزادی"
```

Pros and Cons :

One-hot vector :

advantages:

One-hot encoding easily processed by machine learning models because it is easy to implement and have simple structure.

One-hot encoding preserves the original meaning of categorical features and makes them more interpretable for humans.

disadvantages:

One-hot encoding can result in a significant increase in the number of features, which can lead to computational challenges and overfitting.

If embedding of one-hot vector done in a large data set, result matrix can be very large and use lot of memory.

TF-IDF:

advantages:

can help capture the semantic meaning of words.

TF-IDF reduce the number of common words such as "the" and "a" ,which can help reduce their impact on the model.

TF-IDF is widely used in information retrieval systems such as search engines to rank documents based on their relevance to a query.

disadvantages:

TF-IDF does not capture the order or context of words in a document, which can lead to reduce undrestanding of context.

TF-IDF operates at the word-level and does not capture the meaning of phrases or sentences,

Word2Vec:

advantages:

word2vec can capture the relationships between words, such as synonyms, antonyms, which can improve the performance of many NLP tasks.

Word2Vec generates vector representations of words, which can be used as features in machine learning models for text classification, sentiment analysis, and other tasks.

disadvantages:

Training a Word2Vec model can be computationally expensive, especially with large datasets and high-dimensional vector representations.

Word2Vec operates at the word-level and does not capture the meaning of phrases or sentences, which can limit its effectiveness in tasks such as sentiment analysis or text generation.

Contextualized embeddings:

advantages:

Contextualized embeddings are designed to capture the meaning of a word based on the context in which it appears, which can improve the performance of many natural language processing tasks.

Contextualized embeddings have been shown to improve the performance of many natural language processing tasks, such as sentiment analysis, named entity recognition, and machine translation.

disadvantages:

Training and using contextualized embeddings can be computationally expensive, especially with large datasets and high-dimensional vector representations.

As contextualized embedding designed for complex relations between words in context,they hard to examine and interpret.

5m 16s completed at 7:12 PM