```
from google.colab import drive
drive.mount('/content/gdrive')

SOURCE_DIR = '/content/mahsa_amini_data.csv'
```

```
Mounted at /content/gdrive
```

```
import torch
import re
import pandas as pd
from sklearn import preprocessing
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_ezafe(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/
Collecting json-lines
  Downloading json_lines-0.5.0-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from json-lines) (1.15.0)
Installing collected packages: json-lines
Successfully installed json-lines-0.5.0
```

```
import json_lines


# 1. extract all tweets from files and save them in memory base on each year.
# 2. remove urls, hashtags and usernames.
data = pd.read_csv("mahsa_amini_data.csv")
corpus = []
#print(data.Text)
for text in data.Text:
  new_text = delete_hashtag_usernames(text)
  corpus.append(delete_ezafe(delete_url(new_text)))

corpus
```

```
  'مهسا',
  ' برای آزادی',
  'ما جاییایم که آسمون هست ولی واسه پرواز شده بالمون بسته',
  'مادرجندهها تو خونشون و روز تولدش حجابش این بوده سگبگاد تکتکتون رو',
  'برای ترویج فساد و دروغ',
  'وفردای ازادی هیچ چیز رافراموش نخواهیم کرد. نون خورهای ج.ا. از صدا و سیما گرفته تا اون بالا مالاها...',
  'برای پدرهایی که دستشان خالی بود و چشمانشان پر.',
  'برای آزادی',
  'برای',
  'وای',
  ' استوری فارسی شاهان گوکباکار (رجب اودیک)',
  'به خاطر مردم شجاع و قویمون',
  'هیچوقت هم اسم این جاکش مسلکها رو نتونستم درست بخونم، ابرآوران؟ ابرآروان؟ برآبروان؟',
  'فالو کردم:) بک میدی اکانت منم دیده بشه🙂',
  'چهار',
  'برای آزادی',
  'برای مهسا',
  'این بده که😐 ',
  'نه دشمن دراین بوم و بر لانه داشت نه بیگانه جایی در این خانه داشت از آنروز دشمن بما چیره گشت که ما را روان و خرد تیره گشت از آنروز این خانه ویرانه شد که نان آورش مرد بیگانه شد چو ناکس به ده کدخدایی کند کشاورز باید گدایی کند 1/2',
  'بیست و هشت',
  'بفرستیم برای آدمای مختلف، کامنت کنیم و نمیدونم، فکر کنم توی توییتر جا نمیشه. یکی بهم بگه لیمیت توییتر چقدره و این متن چقدر طولانیه، مرتبش کنیم دوباره و توییت کنید تا میتونید. هشتگها رو هم هرطور که ترنده تغییر بدیم.',
  'فالورام کمه حساب نمیشه😭',
  'برای پیروزی',
  'براي غواصان جوان عمليات فريب! كربلاي ٤ كه برادران عراقيتون با دست بسته زنده زنده خاكشون كردند. عملياتي كه لو رفته بود ولي جوان هاي ايراني رو عمدا فرستادند جلو.',
  'همه باهم برای ایران❤',
  'برای بلوچها',
  'همین که هشتکا بره بالا کافیه اصلا هیچکس منو نبینه',
  'مهسا هرگز تنها نخواهد ماند',
  'اخوند همان آخوندک است',
  'این خیلی خوبه دانشگاه الزهرا تظاهرات اونجا دو گل محسوب میشه زمین میزبانه😁',
  'ارتش باید بیاد به کمک مردم#مهسا_امینی',
  'تجمع اعتراضی یکم اکتبر ۲۰۲۲ در مرکز شهر وین اتریش پ.ن: برای احترام به حقوق اخلاقی اشخاص روی ویدیو فیلتر اعمال شده. ',
  'امینی',
  'دمت گرم. همه ی تلاشمونو میکنیم که بی نتیجه نمونه.',
  'نه این وری نه اون وری. میریم به بیت رهبری. ... آره دیگه، اونم بهمون چسبیده.',
  'برای ژینا',
  'دمت گرم مَرد',
  'برای همهی خشمهایی که تو ما انباشته شده',
  'اواین توییت امشب تقدیم به که عاشق ماهی سیاه کوچولو بود. فرزاد شهید راه آگاهی شد و هنوزم خانوادش نمیدونن کجا دفن شده.',
  'ای وای ....ای وای ...',
  'حسین رونقی نشونمون داد چطوری نترسیم :) اونجا که وسط مصاحبه گفت "ایراد نداره"',
  'میریم واسه ۸۰ میلیووووووون :)',
  "برای بغض هامون وقتی به این پیام برمیخوریم «this item isn't available in your country»",
  'کاشکی اون و میبستم',
  'برای ، خانواده بزرگ ایران .',
  'این سیس ، سیس یه ایرانیه به خدا ک ایرانی دارن بینشون',
  'زنده باد',
  'به کیرم به کیرم دیشب تو خواب هلال ماهو دیدم',
  'سبز میشم دوباره💚',
  'منشن میزاریییم',
  'هرچی میتونین هشتگ بزنین',
  'من گلو درد ام نمیتونم گریه کنم بچه دو ساله ام اشک هامو ببینه',
  'به امید آزادی \U0001f972❤ولی من عاشقق متحد شدنمونم دختر و پسر مرد و زن کرد و ترک فارس و لر نوجوون و جوون و مسن خواهش میکنم کم نیارید ادامه بدید',
  'به زودی باید برن تو همون استادیوم با اشک چشم سرود خدافظ فرمانده رو بخونن',
  'هنوز باورم نمیشه شروین رو گرفتن...',
...]
```

1. One hot encoding

```
from pyparsing.helpers import List
# 1. find one hot encoding of each word for each year
# 2. find 10 nearest words from "زندگی"
words = set()
for snt in corpus:
  s = snt.split()
  for word in s:
    words.add(word)

one_hot_matrix = np.zeros((len(words) , len(words)), int)
nmd = list(words)
d = {}
for i in range(len(words)):
    one_hot_matrix[i , i] = 1
    d[nmd[i]] = torch.from_numpy(one_hot_matrix[i]).float()

find_k_nearest_neighbors(word, d, 10)
```

```
['آزادی',
 'مادرجنده',
 'ضمنا',
 'Mel',
 'عاشقتش😇',
 'ازادی..',
 'بهش',
 'نریزم؟',
 'راهیندان',
 'کنیم👌']
```

pros :

- better capture relationships between features and increases their predictive power.
- preserves the original meaning of categorical features and makes them more interpretable for humans.
- transforms categorical data into a format that can be easily processed by machine learning models. cons :
- the number of features are sometimes great, which can lead to computational challenges and overfitting.
- when datas are correlated, which can affect the performance of some machine learning models.
- when datas are categorical feature has a large number of unique values, it can result in sparse data, where most of the encoded features are zeros.

2. TF-IDF

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

words = []
d = {}
dd = {}
words = set()
ah = 0

for snt in corpus:
  s = snt.split()
  for word in s:
    words.add(word)
    dd[word] = 0
    ah += 1

for snt in corpus:
    s = snt.split()
    for word in s:
      dd[word] += 1

list_words = list(words)
for i , w in enumerate(list_words):
  d[w] = i

tf_idf_matrix = np.zeros((len(corpus) , len(list_words)), float)

for i in range(len(corpus)):
  s = corpus[i].split()
  for w in s:
    tf_idf_matrix[i][d[w]] = s.count(w) / len(s) * math.log(ah / dd[w])

tf_idf_matrix
dic = {}
for i in range(len(corpus)):
    dic[corpus[i]] = torch.from_numpy(tf_idf_matrix[i]).float()

find_k_nearest_neighbors(corpus[0] , dic , 10 )
```

```
['بنشین تا شود نقش فال ما نقش هم فردا شدن',
 'برای زخمایی که رو بدن پسرامون نقش بسته♡',
 'بدبختانه تو اونم احتمالا نقش بعد ....هستند',
 'برای فردا',
 'فردا فردا فردا خواهیم دید',
 'فردا هم میام',
 'ترند شود',
 'وعده ما فردا',
```

```
'همه با هم برای ایران، فردا بیرون میآییم تا فردا های بهتری داشته باشیم. ',
'کاش برای انتخاب نقش خواهر "حدیث" تو فیلمنامه جدیدتون قبلش ازش یه تست بازیگری میگرفتید ...']
```

Pros:

- it shows importance of a word in a corpus based on its frequency, which can help capture the semantic meaning of words.
- it eliminates words such as "the" and "a" which are used a lot in corpus, so this can help reduce their impact on the model.
- it is used in information retrieval systems such as search engines to rank documents based on their relevance to a query.

cons:

- it does not capture the order or context of words in a document so the ability to capture the full semantic meaning of language is lessen.
- it works in word-level and does not capture the meaning so it is weak in tasks such as sentiment analysis or text generation.
- it's parameteres need to be carefully tuned, such as the choice of the weighting scheme and the selection of stop words.

3. Word2Vec

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

for snt in corpus:
    s = snt.split()
    snts.append(s)

m = Word2Vec(sentences = snts)
m.wv.most_similar(word)
```

```
[('زن', '0.987939715385437'),
 ('امید', '0.9863606691360474'),
 ('ازادی', '0.9854978919029236'),
 ('برای', '0.9816492795944214'),
 ('زندگی،', '0.9796879291534424'),
 ('زندگی', '0.9790499210357666'),
 ('زندگی،', '0.9788276553153992'),
 ('ایران', '0.9785169363021851'),
 ('خواهرم', '0.975979208946228'),
 ('زن،', '0.9707988500595093')]
```

Pros:

- it is used to find synonyms, antonyms, and analogies, which can improve the performance of many natural language processing tasks.
- it can help reduce the sparsity of text data by mapping rare words to similar vector representations as more common words.
- can be used as features in machine learning models for text classification, sentiment analysis, and other tasks.

cons:

- it is computationally expensive, especially with large datasets and high-dimensional vector representations.
- it may not be effective for multilingual or cross-lingual natural language processing tasks.
- it works in word-level and does not capture the meaning so it is weak in tasks such as sentiment analysis or text generation.

4. Contextualized embedding

```
model_checkpoint = "HooshvareLab/bert-base-parsbert-uncased"
```

```
# 1. fine tune a bert model base on all tweets for each year.
# 2. find 10 nearest words from "ولنتاین"
!pip install transformers[sentencepiece]

from transformers import BertModel, BertTokenizer

model = BertModel.from_pretrained(model_checkpoint, output_hidden_states = True)
tokenizer = BertTokenizer.from_pretrained(model_checkpoint)
```

✓ 0s completed at 7:08 PM