

(۱)

Word2vec is a technique for natural language processing . The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.

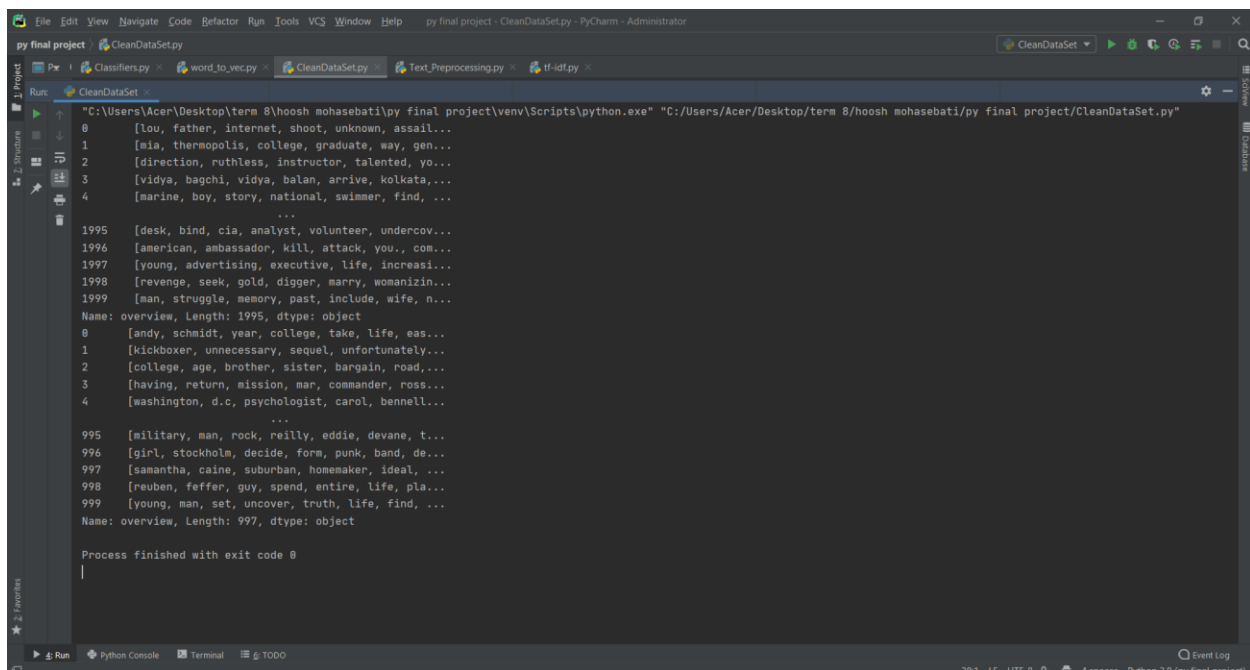
شرح کد:

در فایل `Text_Preprocessing` تعدادی تابع تعریف شده است که وظیفه پیش پردازش داده های دریافتی و لیست کردن کلمات اصلی در هر پاراگراف را دارد. از جمله وظایف این تابه ها میتوان به موارد زیر اشاره کرد:

- حذف داده های تکراری
- حذف داده های غیر متنی
- حذف علائم نگارشی
- ریشه یابی
- حذف space های اضافه
- تبدیل کلمات خلاصه به شکل کامل شان
- حذف stop words
- و در نهایت بازگرداندن `clean text`

در فایل `CleanDataSet` به خواندن داده های `test` و `train` داده شده در پروژه میپردازیم. ابتدا داده های ستون های `genres` و `overview` میخوانیم، تابع `convert_genres` ، داده فیلد `genres` هر فیلم را میگیرید و لیست `genre` هایش را برمیگرداند و داده ی `overview` نیز با توابع تعریف شده ای که در بالا ذکر شد `clean` میشود. در ادامه با استفاده از تابع `MultiLabelBinarizer` برای داده های تست و ترین جدولی میسازد که سطر های آن `overview` هر فیلم و ستون های آن تمام ژانر های تعریف شده است و در تقاطع هر سطر و ستون اگر آن فیلم دارای ژانر آن ستون باشد مقدار آن خانه 1 میگیرد در غیر اینصورت مقدار 0 میگیرید. در ادامه `overview` های کمتر از 10 کاراکتر را حذف میکنند(داده ی اشتباه) و با توابع `Text_Preprocessing` متن تمیز شده(کلمات کلیدی) `overview` را جایگزین میکند و در نهایت فایل خروجی به دست آمده را با فرمت `pkl` ذخیره میکند.

خروجی این مرحله:

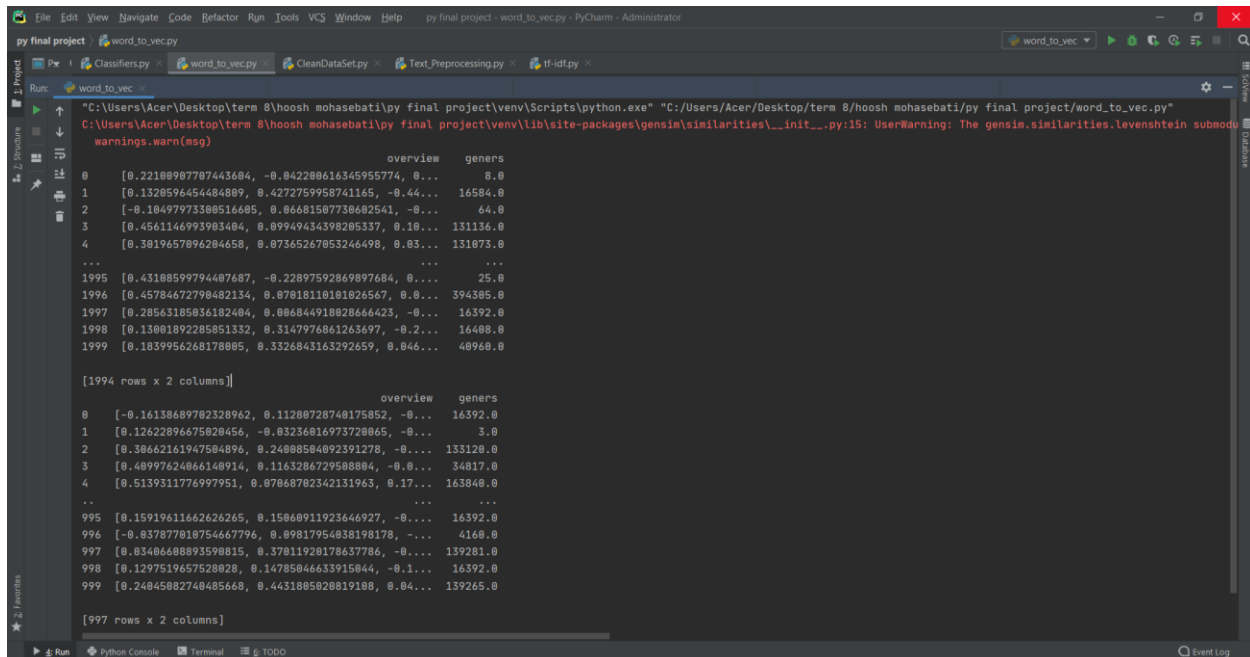


```
"C:\Users\Acer\Desktop\term 8\hoosh mohasebati\py final project\venv\Scripts\python.exe" "C:/Users/Acer/Desktop/term 8/hoosh mohasebati/py final project/CleanDataSet.py"
0 [lou, father, internet, shoot, unknown, assail...
1 [mia, thermopolis, college, graduate, way, gen...
2 [direction, ruthless, instructor, talented, yo...
3 [vidya, bagchi, vidya, balan, arrive, kolkata,...
4 [marine, boy, story, national, swimmer, find, ...
...
1995 [desk, bind, cia, analyst, volunteer, undercov...
1996 [american, ambassador, kill, attack, you., com...
1997 [young, advertising, executive, life, increasi...
1998 [revenge, seek, gold, digger, marry, womanizin...
1999 [man, struggle, memory, past, include, wife, n...
Name: overview, Length: 1995, dtype: object
0 [andy, schmidt, year, college, take, life, eas...
1 [kickboxer, unnecessary, sequel, unfortunately...
2 [college, age, brother, sister, bargain, road,...
3 [having, return, mission, mar, commander, ross...
4 [washington, d.c, psychologist, carol, bennell...
...
995 [military, man, rock, reilly, eddie, devane, t...
996 [girl, stockholm, decide, form, punk, band, de...
997 [samantha, caine, suburban, homemaker, ideal, ...
998 [reuben, feffer, guy, spend, entire, life, pla...
999 [young, man, set, uncover, truth, life, find, ...
Name: overview, Length: 997, dtype: object

Process finished with exit code 0
```

در فایل word2vec با استفاده از glove-wiki-gigaword-50 که یک مدل word2vec از قبل ترین شده آماده است هر کلمه ی موجود در overview هر فیلم را تبدیل به وکتور میکند و تابع avg میانگین وکتور های overview هر فیلم را به دست میاوریم و همچنین با استفاده از to_numpy ترکیب ژانر ها در نهایت در فایل هایی با فرمت pkl ذخیره میکنیم.

خروجی این مرحله:



```
"C:\Users\Acer\Desktop\term 8\hoosh mohasebati\py final project\venv\Scripts\python.exe" "C:/Users/Acer/Desktop/term 8/hoosh mohasebati/py final project/word_to_vec.py"
C:\Users\Acer\Desktop\term 8\hoosh mohasebati\py final project\venv\lib\site-packages\gensim\similarities\_init_.py:15: UserWarning: The gensim.similarities.levenstein submodule is deprecated.
warnings.warn(msg)

overview generators
0 [0.22108987707443604, -0.042280616345955774, 0.0... 0.0
1 [0.1320596454404809, 0.4272759958741165, -0.44... 16584.0
2 [-0.10497973308516605, 0.0668159730602541, -0.0... 64.0
3 [0.4561146993983404, 0.09949434398205337, 0.10... 131136.0
4 [0.3819657696284658, 0.07365267853246498, 0.03... 131073.0
...
1995 [0.43108599794407687, -0.22897592869897684, 0.0... 25.0
1996 [0.45784672790482134, 0.07818118101826567, 0.0... 394305.0
1997 [0.28563185836182404, 0.086844918028666423, -0.0... 16392.0
1998 [0.13081892285051332, 0.3147976861263697, -0.2... 16400.0
1999 [0.1839956268178005, 0.3526843163292659, 0.046... 40960.0

[1994 rows x 2 columns]

overview generators
0 [-0.16138689702328962, 0.11280728740175852, -0.0... 16392.0
1 [0.12622896675020456, -0.03236016973720865, -0.0... 3.0
2 [0.38662161947504896, 0.24088584092191278, -0.0... 133120.0
3 [0.40997624066140914, 0.1163286729508004, -0.0... 34817.0
4 [0.5139311776997951, 0.07868702342319163, 0.17... 163840.0
..
995 [0.15919611662626265, 0.15060911923646927, -0.0... 16392.0
996 [-0.837877810754667796, 0.098179540838198178, -... 4160.0
997 [0.03406808893590815, 0.37011920178637786, -0.0... 139281.0
998 [0.1297519657528028, 0.14785046633915044, -0.1... 16392.0
999 [0.24845082740485668, 0.4431885028819108, 0.04... 139265.0

[997 rows x 2 columns]
```

در مرحله ی آخر در فایل Classifiers داده های خروجی مرحله قبل را اسکیل و تبدیل به آرایه دوبعدی میکند یک کلسیفایر (RandomForestClassifier) را روی train به دست آمده آموزش میدهم و دقت آن را مسنجیم.

خروجی:

accuracy:

0.11133400200601805

برای دقت بیشتر می شد از مدل en_core_web_trf استفاده کرد اما حجم و زمان اجرای بیشتری نیاز دارد که در این پروژه استفاده نشده.

(۲)

bag of words:

The **bag-of-words model** is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. The bag-of-words model has also been used for computer vision.

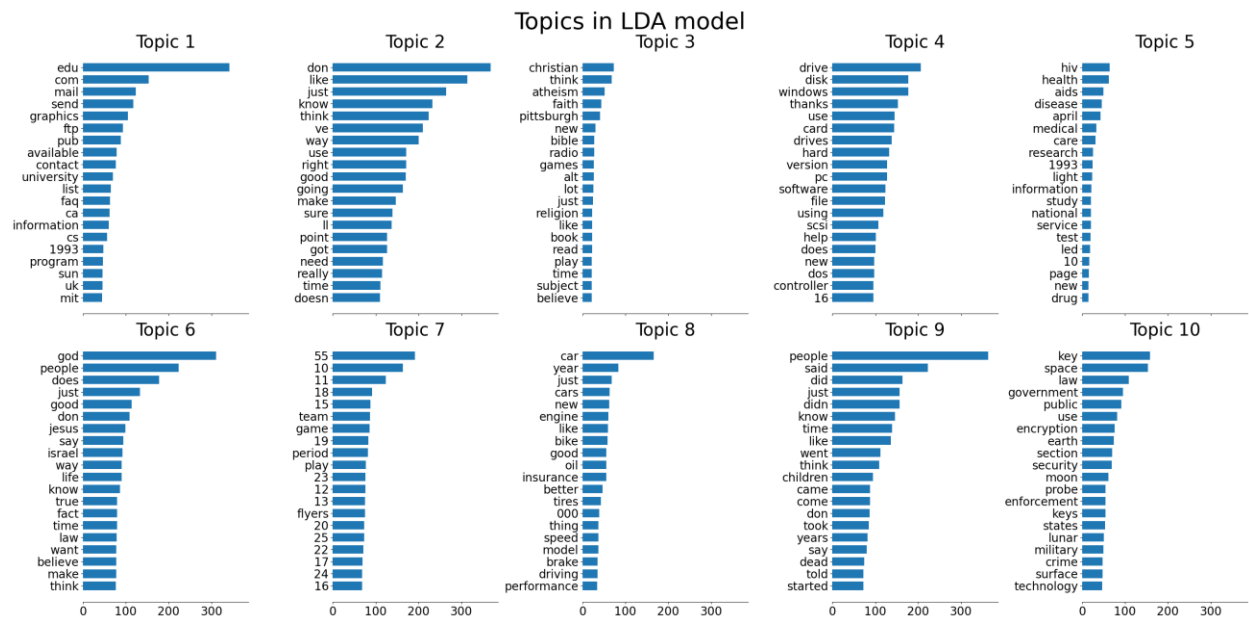
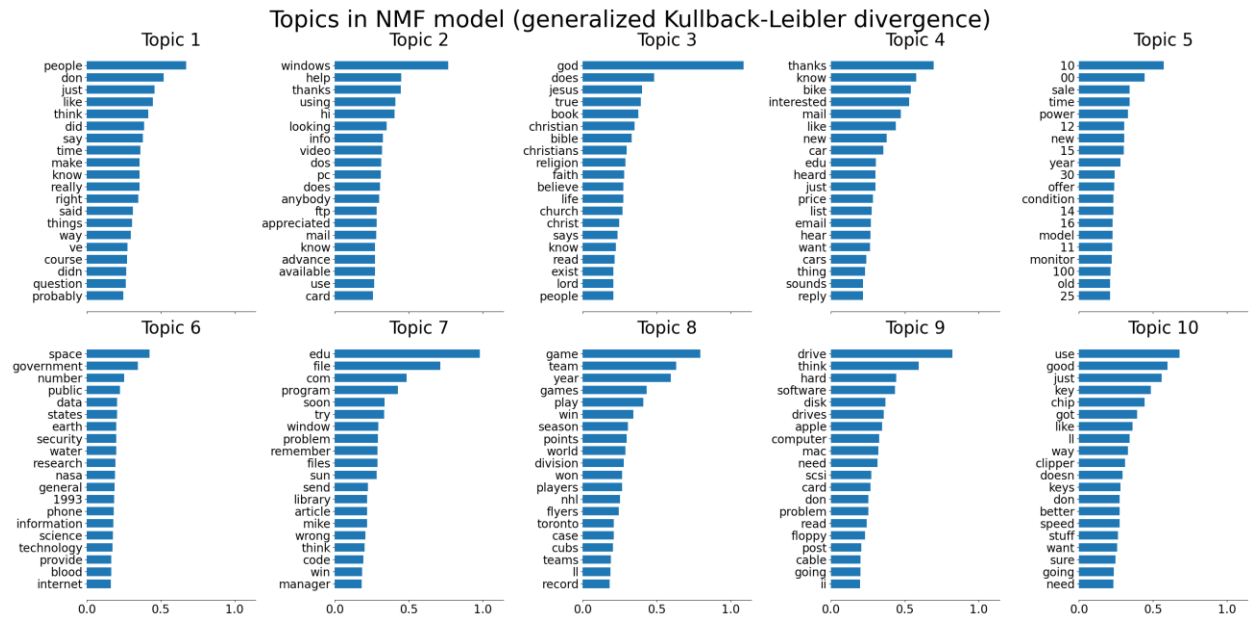
The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier.

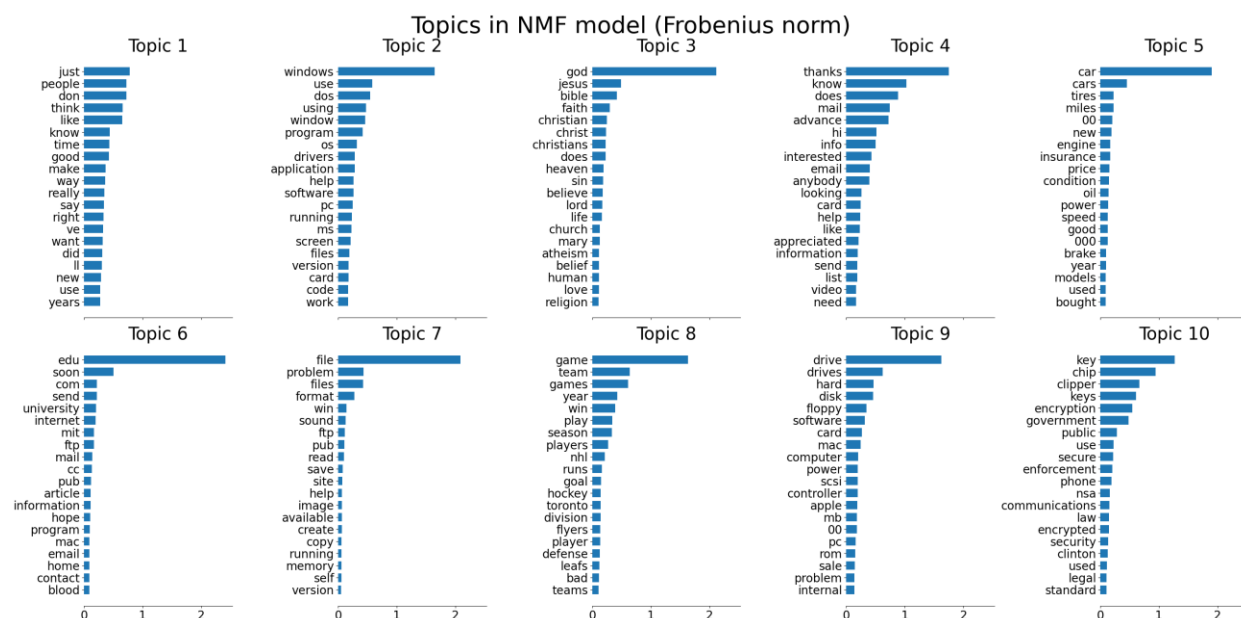
tf-idf:

In information retrieval, **tf-idf**, **TF*IDF**, or **TFIDF**, short for **term frequency-inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.^[1] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf-idf is one of the most popular term-weighting schemes today. A survey conducted in 2015 showed that 83% of text-based recommender systems in digital libraries use tf-idf.

قسمت دوم این پروژه پیاده سازی نشده اما برای درک این الگوریتم از یک نمونه کد آماده استفاده شده است.

خرجی این قسمت:





(۳)

برای بهبود عملکرد میتوان از ضرایب وزن در tfidf برای word2vec استفاده کرد.

(ضرب وزن هر کلمه به دست آمده در tfidf در عدد متناظر با آن کلمه در word2vec و سپس ترین کردن با این دیتاست)

توضیحات بیشتر در مقاله موجود در لینک زیر:

<https://aclanthology.org/S17-2100.pdf>