**هادی تمیمی 9622762408 پروژه 2 هوش محاسباتی**

**1)**

**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.

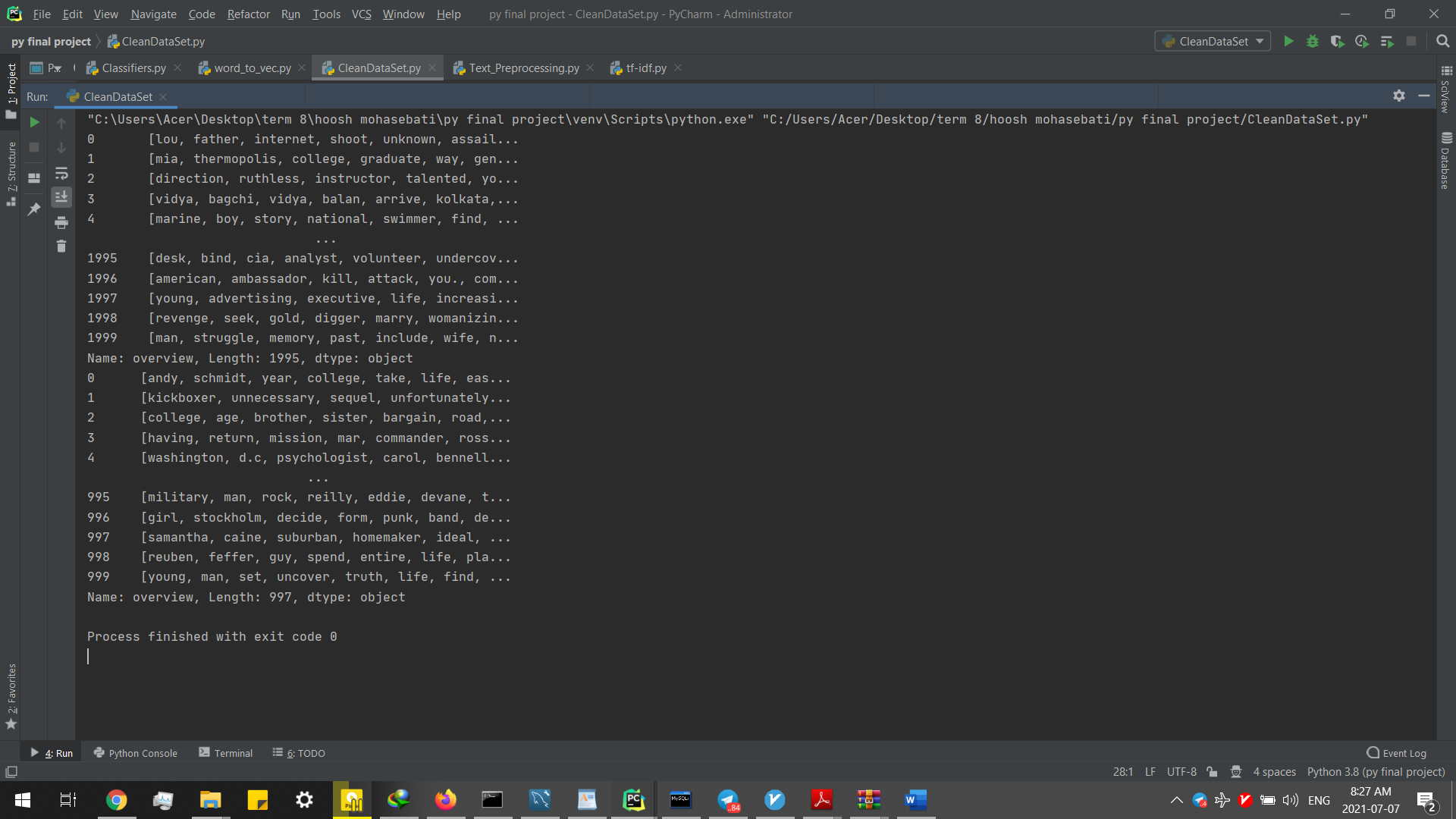
شرح کد:

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 علايم نگارشی
* ریشه یابی
* حذف spaceهای اضافه
* تبدیل کلمات خلاصه به شکل کامل شان
* stop words حذف
* و درنهایت بازگرداندن clean text

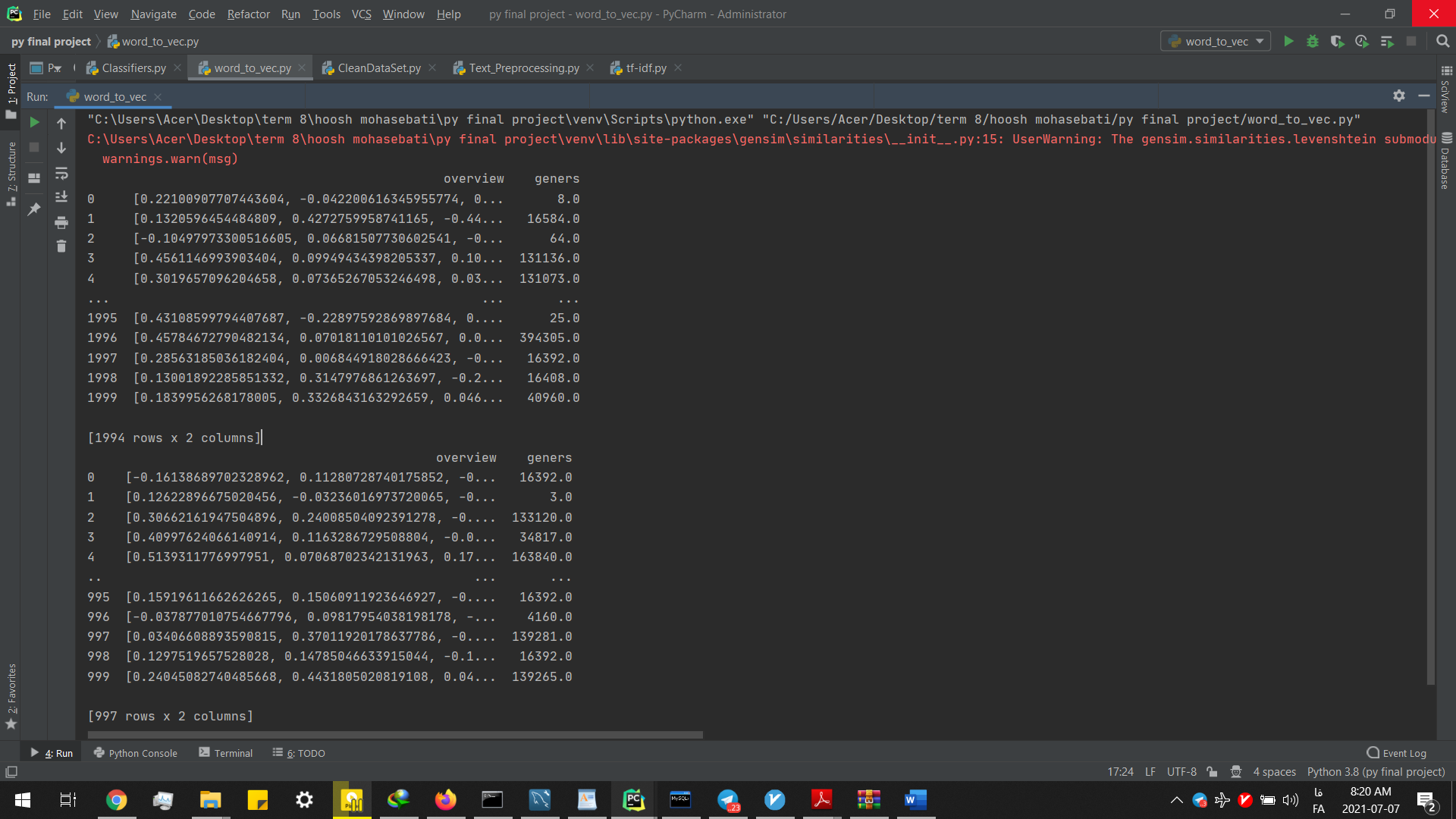
در فایل CleanDataSet به خواندن داده های test و train داده شده در پروژه میپردازیم. ابتدا داده های ستون های genres و overview میخوانیم، تابع convert\_genres ، داده فیلد genres هر فیلم را میگیرید و لیست genre هایش را برمیگرداند و داده ی overview نیز با توابع تعریف شده ای که در بالا ذکر شد clean میشود. در 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 میکند(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:



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:



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 میدهیم 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.

2)

**bag of words:**

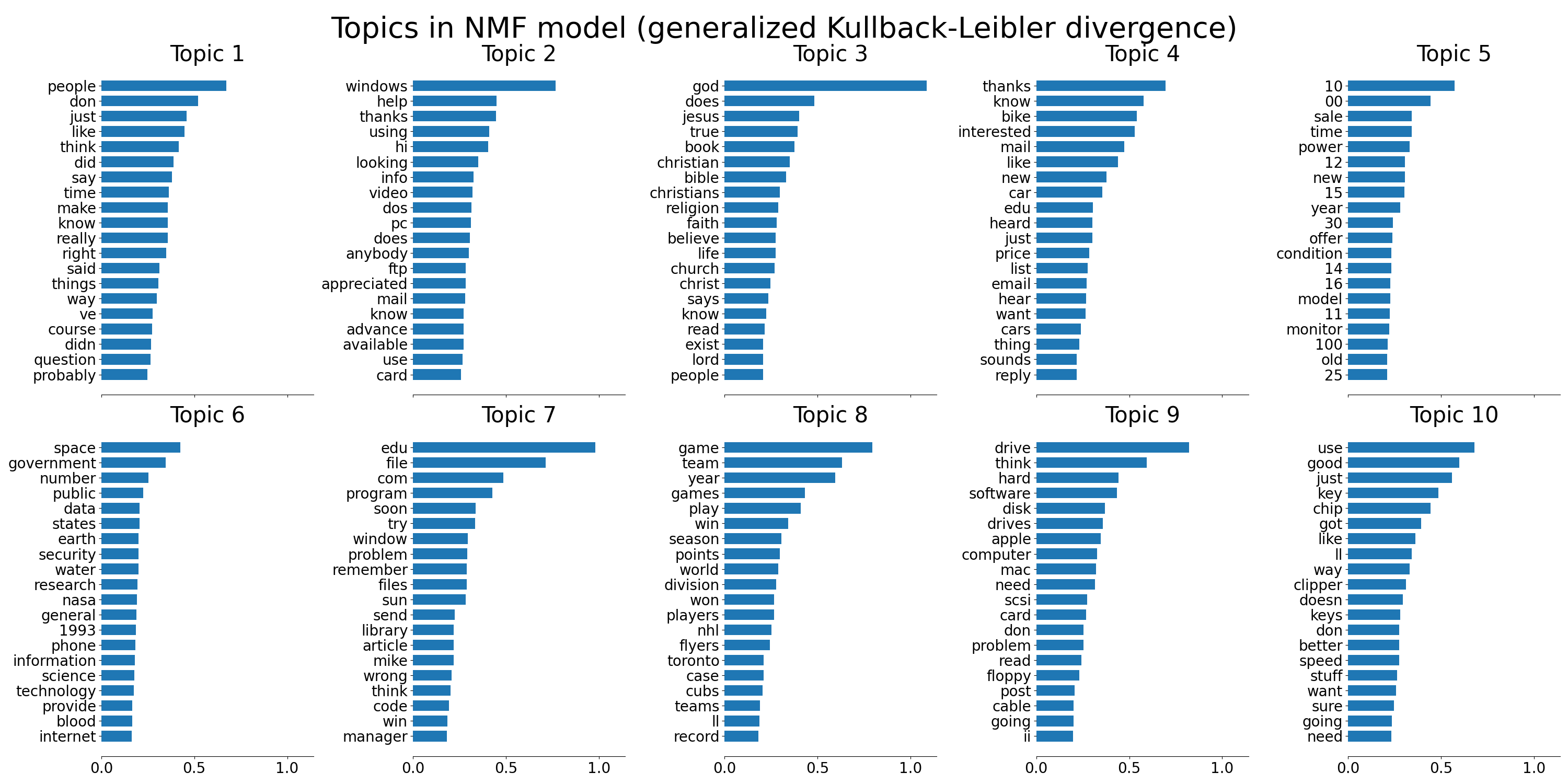
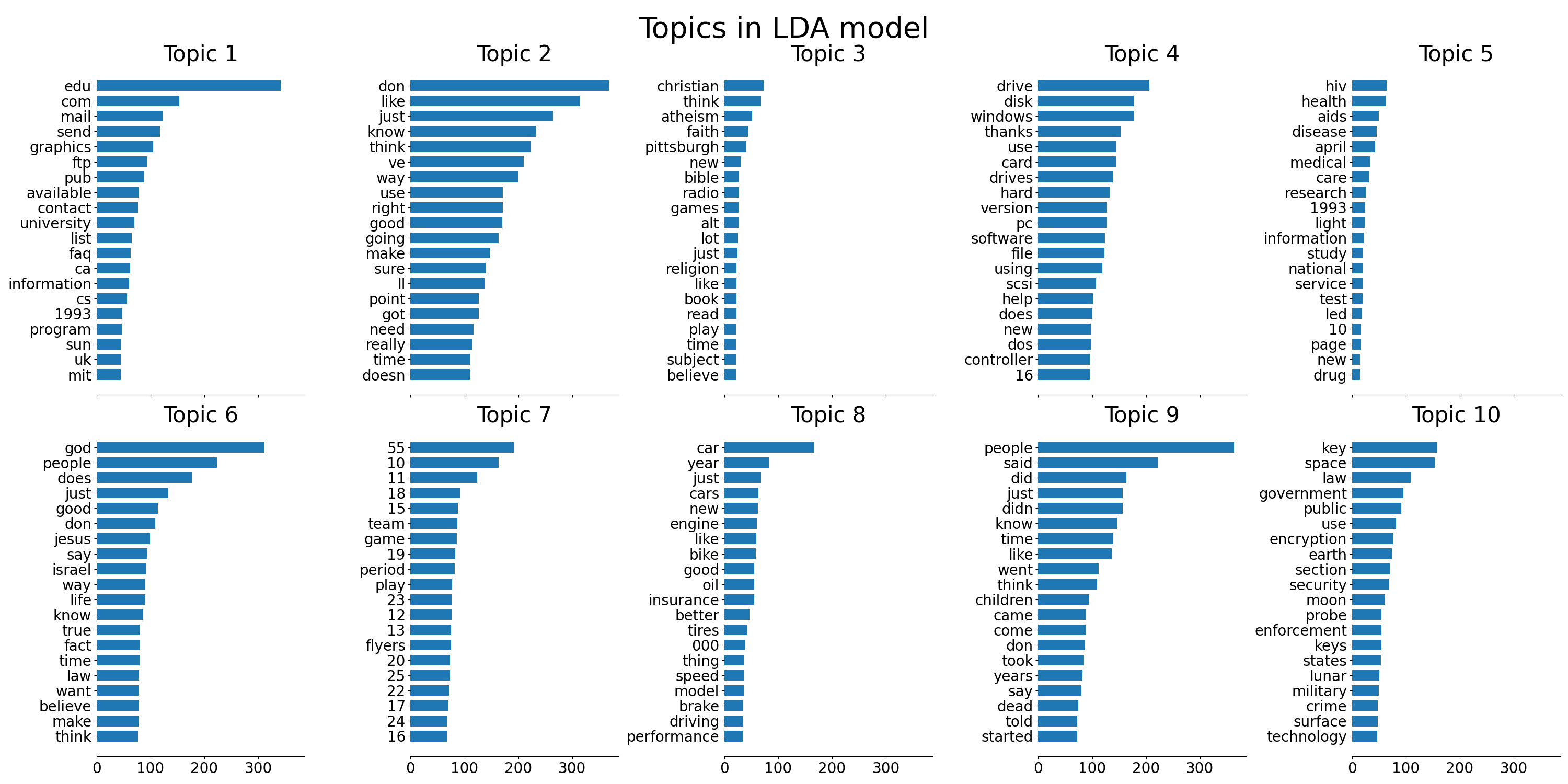
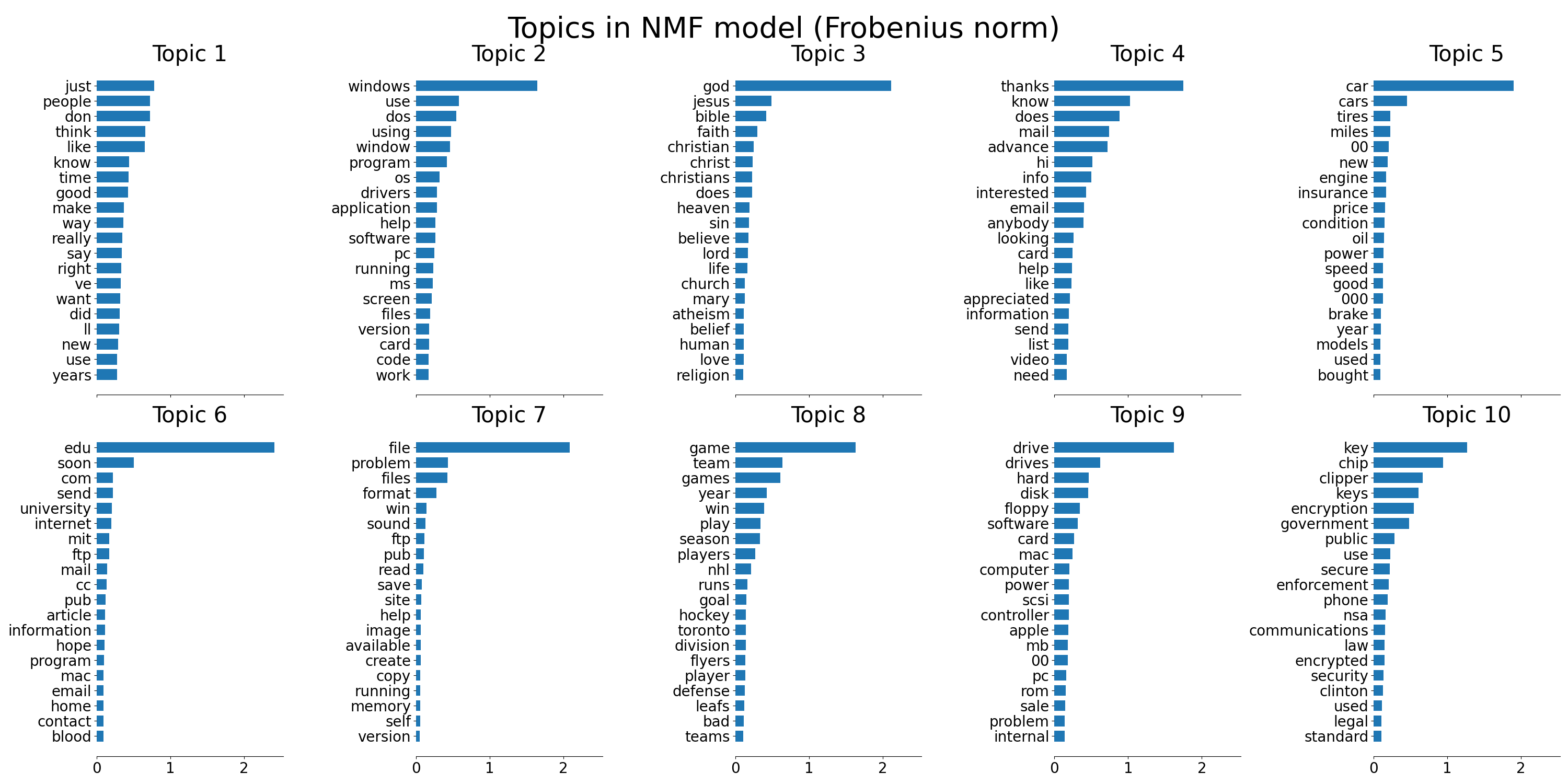
The **bag-of-words model** is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.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

3)

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

(ضرب وزن هر کلمه به دست آمده در tfidfدر 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>