Домашнее задание №2

Асланов А.Б., ИУ9-21М

Задание 2

Запросы - это проанализированные факты из Википедии.
Документы - это предложения из статей Википедии, указанных в этих фактах.

Обработать морфологическим анализатором и найти наиболее релевантные предложения:

- По векторной модели без `idf`;
- По TF-IDF (`df` в данном случае - количество предложений, в которых встретилось слово).

Нормировать запросы и предложения, то есть выстроить все предложения из статей по мере сходства с запросом по векторной модели.
В отчёте должны быть показаны веса выдаваемых предложений.

Запросы (== факты):

1) В медленном движении медленно читают;
2) Петру I доставляли из Китая посылки с золотым песком;
3) Зоны отдыха экипажей дальнемагистральных авиарейсов находятся над или под пассажирским салоном.

Документы:

К факту 1:
Д1.1: Медленное чтение как часть медленного движения;
Д1.2: Однако в последнее время наблюдается повышенный интерес к медленному чтению как направлению в рамках медленного движения.

К факту 2 прямо относящихся предложений из статьи не было. Наиболее подходящие косвенные:
Д2.1: Коробчатое золото — золотой песок, который поставлялся в Российскую империю из Китая для чеканки червонцев.
Д2.2: Коробчатое золото использовалось для чеканки червонцев в 1701 и 1704 годах.

К факту 3:
Д3.1: Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лайнерах и могут располагаться над пассажирским салоном (в этом случае в них нужно подниматься по лестнице) или рядом с ним.

In [1]:
```
from pymystem3 import Mystem
m = Mystem()

from collections import Counter
import re
import string

import pandas as pd
import numpy as np
```

In [2]:
```
def make_morph_analysis(fact):
    """Проводим морфологический анализ для последующей обработки"""

    # Удлить все знаки препинания
    fact = ''.join(ch for ch in fact if ch not in "[«»;,()'`-—.?!:']")

    # Сделать все слова предложения в нижнем регистре
    fact = fact.lower()

    # Лемматизация
    fact = m.lemmatize(fact)
    fact = ''.join(fact)
    return fact
```

In [3]:
```
docs = ['Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лай
нерах и могут располагаться над пассажирским салоном (в этом случае в них нужно по
дниматься по лестнице) или рядом с ним',
        'Зона отдыха экипажа — отсек или специально отведённое место внутри самол
ёта, предназначенное для отдыха членов экипажа при выполнении дальних рейсов.',
        'Члены экипажа могут воспользоваться зонами отдыха в нерабочее время',
        'Модуль либо имеет форму стандартного грузового контейнера и помещается в
 грузовой отсек самолёта, расположенный под пассажирским салоном (вход в него осущ
ествляется через специальный люк, скрытый от пассажиров), либо предназначен для уст
ановки внутри пассажирского салона',
        'Так, на дальнемагистральных самолётах Аэрофлота обособленный отсек отдых
а есть только у лётного экипажа, бортпроводники могут отдыхать только на задних ряд
ах салонов, если там нет пассажиров',
        'Отсеки отдыха также могут присутствовать на транспортных и военно-трансп
ортных самолётах',
        'Секции для пилотов, расположенные в передней части самолёта, отделены от
 расположенных сзади секций для бортпроводников.',
        'Исключения могут быть сделаны для отсеков, оборудованных креслами с рем
нями']

query = ['Зоны отдыха экипажей дальнемагистральных авиарейсов находятся над или п
од пассажирским салоном']
```

In [4]:
```
def extract_all_docs_words(docs):
    """Извлекаем все слова из документа или запроса и прогоняем их через морфолог
ический анализатор"""

    all_docs_words = []
    for doc in docs:
        all_docs_words += doc.split()
    all_docs_words = ' '.join(all_docs_words)
    all_docs_words = make_morph_analysis(all_docs_words)
    all_docs_words = list(set(all_docs_words.split()))
    return all_docs_words

all_docs_words = extract_all_docs_words(docs)
```

In [5]:
```
def check_the_same_words(query, doc, counter):
    """Находим количество совпадений слов в запросе и документе"""

    # Морфологическая обработка поданного на вход документа
    doc = make_morph_analysis(doc)

    # Разбиваем предложение формата <str> на список
    doc = doc.split()

    # Словари с частотами слов в запросе и в документе
    d_doc = Counter(doc)

    # Если i-ое слово из запроса есть в документе, то считаем частотность этог
о i-го слова в документе.
    # Если же такого слова нет, ставим 0.
    doc_vec = []
    for word in all_docs_words:
        if word in doc:
            doc_vec.append(d_doc[word])
        else:
            doc_vec.append(0)

    d = {f'd_{counter}':doc_vec}
    dataframe = pd.DataFrame(data=d, index=all_docs_words)
    return dataframe


def make_termdoc_matrix(query, docs):
    """Создаём матрицы терм-документ"""

    counter = 0 # счётчик документов
    dataframe = pd.DataFrame()
    for doc in docs:
        counter += 1
        dataframe_curr = check_the_same_words(query, doc, counter)
        dataframe = pd.concat([dataframe, dataframe_curr], axis=1)

    # Считаем df как кол-во предложений, в которых встретилось i-ое слово запр
оса.
    # Его проще вычислить по датафрейму: df будет равен количеству ненулевых с
толбцов для i-го слова.
    dataframe['df'] = pd.Series((dataframe != 0).astype(int).sum(axis=
1))


    # Добавляем в датафрейм столбец с запросом
    query = make_morph_analysis(query)
    query = query.split()
    d_query = Counter(query)
    query_vec = []
    for word in all_docs_words:
        if word in query:
            query_vec.append(d_query[word])
        else:
            query_vec.append(0)

    dataframe['q'] = query_vec

    return dataframe


dataframe = make_termdoc_matrix(query, docs)
dataframe
```

Out[5]:

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	df	q
располагать	0	0	0	1	0	0	2	0	2	0
ряд	0	0	0	0	1	0	0	0	1	0
при	0	1	0	0	0	0	0	0	1	0
оно	0	0	0	1	0	0	0	0	1	0
иметь	0	0	0	1	0	0	0	0	1	0
аэрофлот	0	0	0	0	1	0	0	0	1	0
там	0	0	0	0	1	0	0	0	1	0
люк	0	0	0	1	0	0	0	0	1	0
под	0	0	0	1	0	0	0	0	1	1
предназначать	0	1	0	1	0	0	0	0	2	0
помещаться	0	0	0	1	0	0	0	0	1	0
также	0	0	0	0	0	1	0	0	1	0
располагаться	1	0	0	0	0	0	0	0	1	0
вход	0	0	0	1	0	0	0	0	1	0
контейнер	0	0	0	1	0	0	0	0	1	0
летный	0	0	0	0	1	0	0	0	1	0
по	1	0	0	0	0	0	0	0	1	0
воспользоваться	0	0	1	0	0	0	0	0	1	0
если	0	0	0	0	1	0	0	0	1	0
отделять	0	0	0	0	0	0	1	0	1	0
на	1	0	0	0	2	1	0	0	3	0
он	1	0	0	0	0	0	0	0	1	0
от	0	0	0	1	0	0	1	0	2	0
установка	0	0	0	1	0	0	0	0	1	0
стандартный	0	0	0	1	0	0	0	0	1	0
самолет	0	1	0	1	1	1	1	0	5	0
специальный	0	0	0	1	0	0	0	0	1	0
внутри	0	1	0	1	0	0	0	0	2	0
они	1	0	0	0	0	0	0	0	1	0
секция	0	0	0	0	0	0	2	0	1	0
...	...	...	...	...	...	...	...	...	...	...
этот	1	0	0	0	0	0	0	0	1	0
осуществляться	0	0	0	1	0	0	0	0	1	0
экипаж	1	2	1	0	1	0	0	0	4	1
нужно	1	0	0	0	0	0	0	0	1	0
мочь	1	0	1	0	1	1	0	1	5	0
задний	0	0	0	0	1	0	0	0	1	0
член	0	1	1	0	0	0	0	0	2	0
время	0	0	1	0	0	0	0	0	1	0
быть	0	0	0	0	1	0	0	1	2	0
кресло	0	0	0	0	0	0	0	1	1	0
рядом	1	0	0	0	0	0	0	0	1	0
место	0	1	0	0	0	0	0	0	1	0
либо	0	0	0	2	0	0	0	0	1	0
в	2	0	1	2	0	0	1	0	4	0
через	0	0	0	1	0	0	0	0	1	0
отдых	1	2	1	0	1	1	0	0	5	1
у	0	0	0	0	1	0	0	0	1	0
подниматься	1	0	0	0	0	0	0	0	1	0
салон	1	0	0	2	1	0	0	0	3	1
военнотранспортный	0	0	0	0	0	1	0	0	1	0
нерабочий	0	0	1	0	0	0	0	0	1	0
транспортный	0	0	0	0	0	1	0	0	1	0
отдыхать	0	0	0	0	1	0	0	0	1	0
или	1	1	0	0	0	0	0	0	2	1
дальний	0	1	0	0	0	0	0	0	1	0
часть	0	0	0	0	0	0	1	0	1	0
так	0	0	0	0	1	0	0	0	1	0
отводить	0	1	0	0	0	0	0	0	1	0
нет	0	0	0	0	1	0	0	0	1	0
лайнер	1	0	0	0	0	0	0	0	1	0

90 rows × 10 columns

1. Векторная модель (без учета `idf`)

Вес определяется как функция от количества вхождений терма в документ.

In [6]:
```
import operator
from scipy.spatial import distance


def calculate_cosine_similarity(dataframe, doc_num):
    """Расчёт косинусного расстояния для векторной модели"""

    document = np.array(dataframe[f'd_{doc_num}'])
    query = np.array(dataframe['q'])
    return 1 - distance.cosine(document, query)

# Запись в словарь всех косинусных расстояний для каждого конкретного документа
answers = {}
for i, doc_num in zip(range(len(docs)), range(1, len(docs)+1)):
    answers[docs[i]] = calculate_cosine_similarity(dataframe, doc_num)

# Сортировка по значениям словаря для упорядочения порядка ранжирования

def sort_answers(answers, query):
    print('Запрос: ', query, '\n'*3)
    for ans in sorted(answers, key=answers.get, reverse=True):
        print(ans, answers[ans])

final_answers = sort_answers(answers, query)
```

```
Запрос:  ['Зоны отдыха экипажей дальнемагистральных авиарейсов находятся над или
под пассажирским салоном']



Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лайнерах и
могут располагаться над пассажирским салоном (в этом случае в них нужно поднимать
ся по лестнице) или рядом с ним 0.42600643361512924
Зона отдыха экипажа — отсек или специально отведённое место внутри самолёта, пред
назначенное для отдыха членов экипажа при выполнении дальних рейсов. 0.41702882
811414954
Члены экипажа могут воспользоваться зонами отдыха в нерабочее время 0.333333333
33333326
Модуль либо имеет форму стандартного грузового контейнера и помещается в грузовой
отсек самолёта, расположенный под пассажирским салоном (вход в него осуществляетс
я через специальный люк, скрытый от пассажиров), либо предназначен для установки
внутри пассажирского салона 0.25125945381480297
Так, на дальнемагистральных самолётах Аэрофлота обособленный отсек отдыха есть то
лько у лётного экипажа, бортпроводники могут отдыхать только на задних рядах салон
ов, если там нет пассажиров 0.24759378423606915
Отсеки отдыха также могут присутствовать на транспортных и военно-транспортных са
молётах 0.10540925533894596
Секции для пилотов, расположенные в передней части самолёта, отделены от располож
енных сзади секций для бортпроводников. 0.0
Исключения могут быть сделаны для отсеков, оборудованных креслами с ремнями 0.0
```

2. Модель $TF-IDF$

Вес определяется как произведение функции от количества вхождений терма в документ и функции от величины, обратной количеству документов коллекции, в которых встречается этот терм.

Расчёт косинусного расстояния для векторной модели

Косинусное расстояние позволяет вычислить меру сходства двух документов как $cos(\vec{q}, \vec{d}) = \frac{\vec{q}\vec{d}}{||\vec{q}||*||\vec{d}||}$, где q - вектор запроса, а d - вектор документа.

In [11]:
```
from scipy.spatial import distance
import math

def calculate_cosine_similarity_tfidf(dataframe, doc_num, N=len(docs
)):

    tf = np.array(dataframe[f'd_{doc_num}'])
    df = np.array(dataframe['df'])
    idf = np.log10(N / df)

    document = tf * idf
    query = np.array(dataframe['q'])
    return 1 - distance.cosine(document, query)


# Запись в словарь всех косинусных расстояний для каждого конкретного документа
answers_tfidf = {}
for i, doc_num in zip(range(len(docs)), range(1, len(docs)+1)):
    answers_tfidf[docs[i]] = calculate_cosine_similarity_tfidf(datafra
me, doc_num)

sort_answers(answers_tfidf, query)
```

```
Запрос:  ['Зоны отдыха экипажей дальнемагистральных авиарейсов находятся над или
под пассажирским салоном']



Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лайнерах и
могут располагаться над пассажирским салоном (в этом случае в них нужно поднимать
ся по лестнице) или рядом с ним 0.32961809135077114
Зона отдыха экипажа — отсек или специально отведённое место внутри самолёта, пред
назначенное для отдыха членов экипажа при выполнении дальних рейсов. 0.25856776
2497498
Модуль либо имеет форму стандартного грузового контейнера и помещается в грузовой
отсек самолёта, расположенный под пассажирским салоном (вход в него осуществляетс
я через специальный люк, скрытый от пассажиров), либо предназначен для установки
внутри пассажирского салона 0.2039570167126472
Члены экипажа могут воспользоваться зонами отдыха в нерабочее время 0.199098183
92541212
Так, на дальнемагистральных самолётах Аэрофлота обособленный отсек отдыха есть то
лько у лётного экипажа, бортпроводники могут отдыхать только на задних рядах салон
ов, если там нет пассажиров 0.1327452154007247
Отсеки отдыха также могут присутствовать на транспортных и военно-транспортных са
молётах 0.037382952568958516
Секции для пилотов, расположенные в передней части самолёта, отделены от располож
енных сзади секций для бортпроводников. 0.0
Исключения могут быть сделаны для отсеков, оборудованных креслами с ремнями 0.0
```

Результаты

Жирным шрифтом выделены предложения, которые, как ожидал при выделении вручную, должны быть наиболее релевантны запросу. Однако сейчас соглашусь с моделями, которые утверждают, что предложение *Коробчатое золото использовалось для чеканки червонцев в 1701 и 1704 годах* не является настолько подходящим, как предложения, которым присвоены бОльшие вероятности.

В целом системы на данных примерах сопоставимы по качеству, однако TF-IDF присваивает вариантам в целом меньшие веса, то есть более "не уверен" в верности выдачи.

Запрос: ['В медленном движении медленно читают']

Векторная модель	
0.5	***Однако в последнее время наблюдается повышенный интерес к медленному чтению как направлению в рамках медленного движения.***
0.4743416490252569	***Медленное чтение как часть медленного движения***
0.33541019662496846	Однако теоретики метода утверждают, что идея медленного чтения зародилась гораздо раньше появления медленного движения.
0.33541019662496846	Создатели сайта утверждают, что лучший способ быть частью медленного движения — это читать хорошие книги
0.3113995776646092	В настоящее время существует сайт медленного движения, который предоставляет доступ к информации, ресурсам, услугам и сетевым возможностям всем, кто…
0.28284271247461895	Последователи методики медленного чтения не являются организованными в своем движении: «нет ни одного бланка, ни совета директоров, ни, о ужас, центра…
0.19999999999999996	Прежде всего, будем говорить медленно
0.10540925533894598	Метод медленного чтения позволяет более полно понять и оценить сложный текст при изучении философии и литературы
Модель TF-IDF	
0.22677868380553634	Прежде всего, будем говорить медленно
0.18527268850463563	Однако в последнее время наблюдается повышенный интерес к медленному чтению как направлению в рамках медленного движения.
0,161980178	Создатели сайта утверждают, что лучший способ быть частью медленного движения — это читать хорошие книги
0,137428803	***Медленное чтение как часть медленного движения***
0,107582101	В настоящее время существует сайт медленного движения, который предоставляет доступ к информации, ресурсам, услугам и сетевым возможностям всем, кто…
0,090019083	Последователи методики медленного чтения не являются организованными в своем движении: «нет ни одного бланка, ни совета директоров, ни, о ужас, центра…
0,042542798	***Однако теоретики метода утверждают, что идея медленного чтения зародилась гораздо раньше появления медленного движения.***
0,008051486	Метод медленного чтения позволяет более полно понять и оценить сложный текст при изучении философии и литературы

Запрос: ['Петру I доставляли из Китая посылки с золотым песком']

Векторная модель	
0.5735393346764044	Из Китая поступал в основном золотой песок, а золотые слитки привозили в намного меньшем количестве
0.5345224838248488	***Коробчатое золото — золотой песок, который поставлялся в Российскую империю из Китая для чеканки червонцев***
0.3779644730092272	Золотой песок упаковывали в коробочки небольших размеров
0.32025630761017426	Золотые двухрублёвики, которые также называли андреевскими золотыми из-за размещения на одной из сторон изображения апостола Андрея Первозванн…
0.15075567228888181	По одним данным, коробчатое золото, поступавшее на Монетный двор через Сибирский приказ, использовалось иногда для чеканки двухрублёвиков, а по…
0.0	***Коробчатое золото использовалось для чеканки червонцев в 1701 и 1704 годах***
0.0	Уже в конце XVII века через китайскую границу коробчатое золото стало поставляться в значительном количестве
0.0	Деньги пошли на оплату работы мастеров, стоимость самого золота, потери при плавке
Модель TF-IDF	
0,39168	***Коробчатое золото — золотой песок, который поставлялся в Российскую империю из Китая для чеканки червонцев***
0,353772759	Из Китая поступал в основном золотой песок, а золотые слитки привозили в намного меньшем количестве
0,168373998	Золотой песок упаковывали в коробочки небольших размеров
0,123042338	Золотые двухрублёвики, которые также называли андреевскими золотыми из-за размещения на одной из сторон изображения апостола Андрея Первозванн…
0,054296564	По одним данным, коробчатое золото, поступавшее на Монетный двор через Сибирский приказ, использовалось иногда для чеканки двухрублёвиков, а по…
0.0	***Коробчатое золото использовалось для чеканки червонцев в 1701 и 1704 годах***
0.0	Уже в конце XVII века через китайскую границу коробчатое золото стало поставляться в значительном количестве
0.0	Деньги пошли на оплату работы мастеров, стоимость самого золота, потери при плавке 0.0

Запрос: ['Зоны отдыха экипажей дальнемагистральных авиарейсов находятся над или под пассажирским салоном']

Векторная модель	
0.42600643361512924	***Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лайнерах и могут располагаться над пассажирским салоном (в этом случае в них нужно подниматься по лестнице) и…***
0.41702882811414954	Зона отдыха экипажа — отсек или специально отведённое место внутри самолёта, предназначенное для отдыха членов экипажа при выполнении дальних рейсов.
0.3333333333333333	Члены экипажа могут воспользоваться зонами отдыха в нерабочее время
0.251259453814803	Модуль либо имеет форму стандартного грузового контейнера и помещается в грузовой отсек самолёта, расположенный под пассажирским салоном (вход в него осуществляется через специальный люк, скры…
0.24759378423606918	Так, на дальнемагистральных самолётах Аэрофлота обособленный отсек отдыха есть только у лётного экипажа, бортпроводники могут отдыхать только на задних рядах салонов, если там нет пассажиров
0.10540925533894598	Отсеки отдыха также могут присутствовать на транспортных и военно-транспортных самолётах
0.0	Секции для пилотов, расположенные в передней части самолёта, отделены от расположенных сзади секций для бортпроводников.
0.0	Исключения могут быть сделаны для отсеков, оборудованных креслами с ремнями
Модель TF-IDF	
0,329618091	***Отсеки для отдыха экипажа обычно присутствуют на дальнемагистральных лайнерах и могут располагаться над пассажирским салоном (в этом случае в них нужно подниматься по лестнице) и…***
0,258567762	Зона отдыха экипажа — отсек или специально отведённое место внутри самолёта, предназначенное для отдыха членов экипажа при выполнении дальних рейсов.
0,203957017	Модуль либо имеет форму стандартного грузового контейнера и помещается в грузовой отсек самолёта, расположенный под пассажирским салоном (вход в него осуществляется через специальный люк, скры…
0,199098184	Члены экипажа могут воспользоваться зонами отдыха в нерабочее время
0,132745215	Так, на дальнемагистральных самолётах Аэрофлота обособленный отсек отдыха есть только у лётного экипажа, бортпроводники могут отдыхать только на задних рядах салонов, если там нет пассажиров
0,037382953	Отсеки отдыха также могут присутствовать на транспортных и военно-транспортных самолётах
0.0	Секции для пилотов, расположенные в передней части самолёта, отделены от расположенных сзади секций для бортпроводников.
0.0	Исключения могут быть сделаны для отсеков, оборудованных креслами с ремнями

Перепроверка

Возьмём результаты векторной модели и посмотрим, правильно ли было вычислено косинусное расстояние между векторами слов.

Здесь запрос q: "В медленном движении медленно читают", а документ d: "Прежде всего, будем говорить медленно" (предложения лемматизированы).

In [8]:
```
from scipy.spatial import distance

q = [1, 1, 1, 1, 1, 0, 0, 0, 0]    # в медленный движение медленно читать
d = [0, 0, 0, 1, 0, 1, 1, 1, 1]    # прежде всего быть говорить медленно


print(1 - distance.cosine(d, q))    # б/м недоставание до 0,2 в этом случае
 - издержка интерпретатора python'а
```

```
0.19999999999999996
```

Lemma	q	d
в	1	0
медленный	1	0
движение	1	0
медленно	1	1
читать	1	0
прежде	0	1
всего	0	1
быть	0	1
говорить	0	1

$$cos(\theta) = \frac{q*d}{||q||*||d||} = \frac{0+0+0+1+0+0+0+0+0}{\sqrt{1^2+1^2+1^2+1^2+1^2+0^2+0^2+0^2+0^2}\sqrt{0^2+0^2+0^2+1^2+0^2+1^2+1^2+1^2+1^2}} = \frac{1}{\sqrt{5}\sqrt{5}} = 0,2$$

Перепроверка 2

Проверка вычисления векторной модели между запросом q =**В медленном движении медленно читают** и документом d =**Медленное движение как часть медленного движения**.

Lemma	q	d
в	1	0
медленный	1	2
движение	1	1
медленно	1	0
читать	1	0
как	0	1
часть	0	1
чтение	0	1

$$cos(\theta) = \frac{\vec{q}*\vec{d}}{||\vec{q}||*||\vec{d}||} = \frac{0+2+1+0+0+0+0+0}{\sqrt{0^2+2^2+1^2+0^2+0^2+1^2+1^2+1^2}\sqrt{1^2+1^2+1^2+1^2+1^2+0^2+0^2+0^2}} = \frac{3}{\sqrt{8}\sqrt{5}} \approx 0,474$$

In [ ]: