

Методы семантического анализа текстов

Семинар 14

Вопрос

- Зачем нужен семантический анализ?

TFIDF

- **TF-IDF** (от [англ.](#) *TF* — *term frequency*, *IDF* — *inverse document frequency*) — статистическая мера, используемая для оценки важности слова в контексте [документа](#), являющегося частью коллекции документов или [корпуса](#) (википедия). Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.
- Если кратко, то TF-IDF это *term frequency-inverse document frequency* или, ежели на великом и могучем, *частотность терминов-обратная частотность документов*.
- Идея метрики очень проста. Если слово встречается почти во всех документах - его различительная сила очень мала и само слово не является важным. Если слово часто встречается в данном документе, то оно является важным для него.
- Метрика считается на коллекции документов для каждого слова, каждого документа. Для расчета меры можно использовать TfidfVectorizer.
- Если документ содержит 100 слов, и слово[3] «заяц» встречается в нём 3 раза, то частота слова (TF) для слова «заяц» в документе будет 0,03 (3/100). Вычислим IDF как десятичный логарифм отношения количества всех документов к количеству документов, содержащих слово «заяц». Таким образом, если «заяц» содержится в 1000 документах из 10 000 000 документов, то IDF будет равной: $\log(10\,000\,000/1000) = 4$. Для расчета окончательного значения веса слова необходимо TF умножить на IDF. В данном примере, TF-IDF вес для слова «заяц» в выбранном документе будет равен: $0,03 \times 4 = 0,12$.

TFIDF

- **Зачем это нужно?**

Это простой и удобный способ оценить важность термина для какого-либо документа относительно всех остальных документов. Принцип такой — если слово встречается в каком-либо документе часто, при этом встречаясь редко во всех остальных документах — это слово имеет большую значимость для того самого документа.

- **Чем хороша эта метрика?**

1. Слова, неважные для вообще всех документов, например, предлоги или междометия — получают очень низкий вес TF-IDF (потому что часто встречаются во всех-всех документах), а важные — высокий.
2. Её просто считать

Пример

- https://colab.research.google.com/drive/1xJCmgRaFv8Ty2A2K78o_-01XxXvBRMmz?usp=sharing

Задание

- Реализовать TFIDF в произвольном тексте

Word2Vec

- Работа в семантическом пространстве, а не в пространстве частот
- Word2Vec – сводит пространство «слова в слова» в «слова в вектор тематик»
- Word2Vec позволяет представить каждое слово с помощью числового вектора

Похожие слова

- «Идти» и «шагать» - синонимы

Для компьютера это разные строки (разные буквы и длина)

- Как понять, что они похожи?

На основе данных

Слова со схожим смыслом часто идут в паре с одними и теми же словами (одинаковые контексты)
— основа метода

Векторные представления слов

Хотим каждое слово представить вектором с размерностью n компонент

Требования:

- При этом n должна быть не очень большой
- Похожие слова должны иметь близкие векторы
- Арифметические действия над векторами должны иметь смысл

Word2vec

- Необходимо обучить представления слов так, чтобы они хорошо предсказывали свой контекст
- Одна из задач – вычисление вероятности встретить слово i рядом со словом j

Свойства представлений

- king – man + woman \sim queen (вектор, близкий к слову королева)
- Moscow – Russia + England \sim London

Обучение с учителем

- Проблема мешка слов – слишком большое количество признаков
- Средний word2vec-вектор позволяет получить компактное признаковое описание
- При размерности вектора 100 можно обучать композиции деревьев

Установка пакетов

- Numpy
- Scipy
- Gensim
- Word2vec

Для среды питона ставим через командную строку
pip install пакет

Для анаконды заходим в Anaconda Prompt и даем команду **conda install пакет**

Установка пакетов. Пример

```
Anaconda Powershell Prompt (Anaconda3_1)

The following packages will be UPDATED:

  openssl                                1.1.1f-he774522_0 --> 1.1.1g-he774522_0

Proceed <[y]/n>? y

Downloading and Extracting Packages
openssl-1.1.1g           : 4.4 MB      : ##### : 100%
word2vec-0.9.4           : 99 KB      : ##### : 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(base) PS C:\Users\Sanchs> conda install pymorphy2
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
```

Вычисление вероятности встретить следующее слово

```
In [80]: import pandas as pd
import re
import numpy
import scipy
import nltk
import nltk.data
from tqdm.notebook import tqdm

nltk.download('punkt')

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Sanchs\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!

Out[80]: True
```

```
In [81]: with open("D:\MIEM_masters_ML_2019-20\MIEM_masters_ML_2019-20\data\lenta2018_summer2.txt", encoding="utf-8") as f:
    text = f.read()
    div_text = text.split("====\n")[1:]
    text_news = [(n.split('\n')[0], n.split('\n')[1], n.split('\n')[2], '\n'.join(n.split('\n')[3:]))
                  for n in div_text]
    news = pd.DataFrame(text_news, columns = ['Header', 'Date', 'Tag', 'News'])

    texts = list(news['News'])
    texts[100]
```

```
Out[81]: 'В православных храмах нельзя строить туалеты. Такое мнение в интервью радиостанции «Говорит Москва» высказал
член Патриаршего совета по культуре Леонид Калинин. По его словам, это противоречит религиозным нормам. «Храм
— это место молитвы, а не место отправления нужд», — сказал представитель РПЦ. Он напомнил, что традиционно уб
орные в храмах не размещали. «Это нонсенс, что сейчас пытаются в угоду каким-то гражданским нормативам как для
общественных зданий внести это как обязательную норму», — отметил Калинин. По его словам, в туалетах в зданиях
храма нет необходимости, поскольку, если речь идет о городских храмах, «там всегда есть достаточное количество
учреждений общепита, где всегда существуют по нормам туалеты». По мнению члена совета, строить туалеты можно т
олько на прихрамовой территории, в трапезных или воскресных школах. Поводом для критики стали новые правила пр
оектирования для православных храмов, разработанные по поручению Минстроя. Как сообщает информгентство «Релит
ия», согласно документам, в притворе православного храма предлагается делать туалеты. Эта норма был прописана
в предыдущем СанПиНе, но распространялась на новые сооружения. Теперь уборными предлагают оборудовать и памятн
ики истории и культуры храмового назначения: там установка вентиляции практически невозможна без смены планиро
вки. Против этой инициативы также выступили НКО «Благотворительный фонд Рождества Богородицы» и «Московская ор
ганизация инвалидов». Участники организаций пришли к выводу, что «святость православных храмов не сочетается с
```

Вычисление вероятности встретить следующее слово

```
In [82]: def text_to_wordlist(text):  
         text = re.sub('[^a-zA-Zа-яА-ЯёЁ]', ' ', text)  
         words = text.lower().split()  
         return words
```

```
In [83]: text_to_wordlist(texts[100])
```

```
'богородицы',  
'и',  
'московская',  
'организация',  
'инвалидов',  
'участники',  
'организаций',  
'пришли',  
'к',  
'выводу',  
'что',  
'святость',  
'православных',  
'храмов',  
'не',  
'сочетается',  
'с',  
'понятием',  
'туалет']
```

```
In [84]: def text_to_sentences(text):  
         sentences = []  
         tokenizer = nltk.data.load('tokenizers/punkt/russian.pickle')  
         raw_sentences = tokenizer.tokenize(text.strip())  
         for raw_sentence in raw_sentences:  
             if len(raw_sentence) > 0:  
                 sentences.append(text_to_wordlist(raw_sentence))  
         return sentences
```

```
In [85]: text_to_sentences(texts[100])
```

```
'сообщает',  
'информатентство',  
'религия',  
'согласно',  
'документам',  
'в',  
'притворе',  
'православного',  
'храма',  
'предлагается',  
'делать',  
'туалеты'],  
'эта',
```


Вычисление вероятности встретить следующее слово

```
In [86]: all_sentences = [text_to_sentences(x) for x in tqdm(texts)]
```

```
100% ██████████ 1476/1476 [00:04<00:00, 368.89It/s]
```

```
In [87]: all_sentences = sum(all_sentences, [])
```

```
In [88]: from gensim.models.word2vec import Word2Vec
```

```
In [89]: %%time
```

```
# список параметров, которые можно менять по вашему желанию
num_features = 300 # итоговая размерность вектора каждого слова
min_word_count = 5 # минимальная частотность слова, чтобы оно попало в модель
num_workers = 3 # количество ядер вашего процессора, чтоб запустить обучение в несколько потоков
context = 10 # размер окна
downsampling = 1e-3 # внутренняя метрика модели

model = Word2Vec(all_sentences, workers=num_workers, size=num_features,
                 min_count=min_word_count, window=context, sample=downsampling)
```

```
Wall time: 4.46 s
```


```
In [90]: model.wv.most_similar('автомобиль')
```

```
Out[90]: [('российских', 0.9999282360076904),
          ('х', 0.9999279975891113),
          ('которая', 0.9999168515205383),
          ('место', 0.9999164342880249),
          ('между', 0.9999119639396667),
          ('одним', 0.9999030828475952),
          ('последний', 0.9998945593833923),
          ('квартире', 0.9998783469200134),
          ('доходов', 0.9998779892921448),
          ('число', 0.9998762607574463)]
```

```
In [91]: model.wv.most_similar('кризис')
```

```
Out[91]: [('соцсетях', 0.9985260367393494),
          ('ответ', 0.9984731674194336),
          ('важным', 0.9984704256057739),
          ('своих', 0.9984630942344666),
          ('ученные', 0.9984629154205322),
          ('фотографии', 0.9984580278396606),
          ('адрес', 0.9984526038169861),
          ('киев', 0.9984467029571533),
          ('позже', 0.9984411001205444),
          ('разных', 0.9984381198883057)]
```

Соединение слов

```
jupyter Untitled4 Last Checkpoint: час назад (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3

In [183]: import re
          from tqdm import tqdm
          import pandas as pd
          from gensim.models.word2vec import Word2Vec # Собственно модель.
          from gensim.models import KeyedVectors # Семантические вектора.

          import nltk

          import matplotlib.pyplot as plt

          plt.rcParams['figure.figsize'] = (12, 8)

In [184]: with open("D:/1234567.txt") as newsfile:
          text_news = [(n.split("-----\n")[0].split('\n')[0],
                        n.split("-----\n")[0].split('\n')[1],
                        n.split("-----\n")[1]) for n in newsfile.read().split("=====\n")[1:]]
          news = pd.DataFrame(text_news, columns = ['Header', 'Date', 'News'])

In [185]: news.head()

Out[185]:
           Header      Date      News
0  Ukraine president: Putin has one year to strik...  2018/06/01  Ukraine's president, Volodymyr Zelenskiy, beli...
1  Ukraine's Zelenskyy urged by Putin to stick to...  2018/06/01  Russian President Vladimir Putin on Friday urg...
2    Ukraine's recovery is a headache for Putin  2018/06/01  Ukraine can seem a prisoner of negative headli...

In [186]: NewsTitles=news['News'].values


In [187]: NewsTitles

Out[187]: array(['Ukraine's president, Volodymyr Zelenskiy, believes he can negotiate a deal with Vladimir Putin to end
the war in Ukraine, but has threatened to walk away from talks after a year if there is no progress with his R
ussian counterpart.\n\n"Time is ticking," he told the Guardian in a rare interview. "The government can spend
one year on the entire agreement. Then it should be implemented. Any longer is prohibited."\n\nZelenskiy, an a
ctor and comedian who had no political experience when he won presidential elections last spring, said his mee
ting with Putin in Paris in December had "a few emotional parts". He believes he got through to Putin during t
he meeting: "I think he listened to me. I had that feeling. I hope it's not a false feeling."\n\nIn the wide-r
anging interview, conducted in Kyiv in late February, Zelenskiy also discussed his unwanted role in the attemp
ted impeachment of Donald Trump and his relations with the US following the conclusion of the trial.\n',
'Russian President Vladimir Putin on Friday urged Ukrainian leader Volodymyr Zelenskyy to stick to peac
e agreements to de-escalate the five-year conflict between Kyiv forces and Moscow-backed separatists, the Krem
lin said in a blunt statement on Friday.\n\nPutin and Zelenskyy discussed the settlement of the conflict, the
Kremlin said, adding that Putin stressed the importance of the "complete and unconditional implementation" of
Western-brokered peace agreements.\n',
'Ukraine can seem a prisoner of negative headlines. The chief executive of PrivatBank, the country's la
rgest lender nationalised in 2016 with a $5.5bn hole in its accounts, has warned the FT of capital flight and
a plunging currency if President Volodymyr Zelensky does not stand behind the bank's continuing clean-up. Last
week, an offensive by Russian-controlled rebels briefly took fighting in the still-simmering conflict in break
away eastern regions to its highest level in months. Ukraine was also dragged against its will into the narrat
ive of impeachment of US president Donald Trump.\n\nUnder the radar, however, economic news has turned mostly
positive. Guided by its reformist president and parliament, Ukraine is emerging from the downturn that followe
```

Соединение слов

```
jupyter Untitled4 Last Checkpoint час назад (autosaved) Python 3
File Edit View Insert Cell Kernel Widgets Help
In [188]: nltk.download('punkt')
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Sanche\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
Out[188]: True
In [189]: newsVec=[nltk.word_tokenize(News) for News in NewsTitles]
In [190]: newsVec
Out[190]: [['Ukraine',
            ',',
            's',
            'president',
            ',',
            'Volodymyr',
            'Zelenskiy',
            ',',
            'believes',
            'he',
            'can',
            'negotiate',
            'a',
            'deal',
            'with',
            'Vladimir',
            'Putin',
            'to',
            'end',
            'that
In [191]: model=Word2Vec(newsVec,min_count=1,size=32)
In [192]: model.most_similar('Putin')
D:\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: Call to deprecated 'most_similar'
(Method will be removed in 4.0.0, use self.wv.most_similar() instead).
"""Entry point for launching an IPython kernel.
Out[192]: [('Volodymyr', 0.489463210105896),
            ('eastern', 0.4356519877910614),
            ('study', 0.43106773495674133),
            ('staff', 0.41563647985458374),
            ('sell', 0.4005813002586365),
            ('Some', 0.3892856240272522),
            ('been', 0.37717610597610474),
            ('currency', 0.3701048493385315),
            ('shortcomings', 0.3688458502292633),
            ('abroad', 0.3642590343952179)]
```

Соединение слов

```
jupyter Untitled4 Last Checkpoint: час назад (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [191]: model=Word2Vec(newsVec,min_count=1,size=32)

In [192]: model.most_similar('Putin')

D:\Anaconda3_1\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: Call to deprecated 'most_similar'
` (Method will be removed in 4.0.0, use self.wv.most_similar() instead).
    """Entry point for launching an IPython kernel.

Out[192]: [('Volodymyr', 0.489463210105896),
('eastern', 0.4356519877910614),
('study', 0.43106773495674133),
('staff', 0.41563647985458374),
('sell', 0.4005813002586365),
('Some', 0.3892856240272522),
('been', 0.37717610597610474),
('currency', 0.3701048493385315),
('shortcomings', 0.3688458502292633),
('abroad', 0.3642590343952179)]

Type Markdown and LaTeX:  $\alpha^2$ 

In [193]: vec=model['president']-model['Putin']+model['Zelenskyy']

D:\Anaconda3_1\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: Call to deprecated '__getitem__'
(Method will be removed in 4.0.0, use self.wv.__getitem__() instead).
    """Entry point for launching an IPython kernel.

In [194]: model.most_similar([vec])

D:\Anaconda3_1\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: Call to deprecated 'most_similar'
` (Method will be removed in 4.0.0, use self.wv.most_similar() instead).
    """Entry point for launching an IPython kernel.

Out[194]: [('Zelenskyy', 0.6495070457458496),
('consumers', 0.6436102390289307),
('president', 0.521751344203949),
('81', 0.4841344654560089),
('urged', 0.4207967221736908),
(''''', 0.4171024560928345),
(' ', 0.39912089705467224),
('settlement', 0.3971959054470062),
('cent', 0.38802048563957214),
('actor', 0.38332638144493103)]
```

Задание

- Реализовать примеры соединения слов из текстового файла с использованием инструмента Word2Vec.