Введение в обработку естественного языка

Урок 11. Модель Transformer-1

Практическое задание

Домашнее задание к уроку 11

Задание

1. Взять предобученную трансформерную архитектуру и решить задачу перевода (для того же корпуса что вы выбрали из предыдущего дз)
2. скачиваем готовый новостной датасет !wget https://github.com/ods-ai-ml4sg/proj_news_viz/releases/download/data/gazeta.csv.gz

```
# пример работы с ним
from corus import load_ods_gazeta
path = 'gazeta.csv.gz'
records = load_ods_gazeta(path)
next(records)
```

реализовать метод поиска ближайших статей (на вход метода должен приходить запрос (какой-то вопрос) и количество вариантов вывода к примеру топ 5-ть или 3-ри, ваш метод должен возвращать топ-k ближайших статей к этому запросу)
визуально оценить качество

Выполнил ***Соковнин ИЛ***

```
# huggingface.co/models
# Знакомимся с сообществом hugginface, с его библиотекой transformers
!pip install transformers
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting transformers
  Downloading transformers-4.20.1-py3-none-any.whl (4.4 MB)
     |████████████████████████████████| 4.4 MB 8.6 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from transformers) (21.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.8.1-py3-none-any.whl (101 kB)
     |████████████████████████████████| 101 kB 2.3 MB/s
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (596 kB)
     |████████████████████████████████| 596 kB 51.7 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from transformers) (2.23.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from transformers) (4.11.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from transformers) (3.7.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (1.21.6)
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
  Downloading tokenizers-0.12.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (6.6 MB)
     |████████████████████████████████| 6.6 MB 32.2 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (from transformers) (4.64.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (2022.6.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (4.1.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers) (3.0.9)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->transformers) (3.8.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->transformers) (2022.6.15)
Installing collected packages: pyyaml, tokenizers, huggingface-hub, transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.8.1 pyyaml-6.0 tokenizers-0.12.1 transformers-4.20.1
```

```
import transformers
from transformers import AutoTokenizer  # Универсальный токенайзер
# from transformers import BertTokenizer  # Специализированный токенайзер под конкретную модель (Bert)
```

Классификация последовательностей

```
# PipeLine - самые высокоуровневые абстракции библиотеки Hugging Face
from transformers import pipeline


# Задача (task) Sentiment Analysis
# preprocessing: tokenizer, model
#  postprocessing
nlp = pipeline(task="sentiment-analysis", model="Tatyana/rubert-base-cased-sentiment-new")
```

```
Downloading: 100%  943/943 [00:00<00:00, 5.45kB/s]
Downloading: 100%  679M/679M [00:21<00:00, 49.8MB/s]
Downloading: 100%  499/499 [00:00<00:00, 11.7kB/s]
Downloading: 100%  1.34M/1.34M [00:00<00:00, 1.62MB/s]
Downloading: 100%  112/112 [00:00<00:00, 2.52kB/s]
```

```
# Выводим результат

result = nlp("очень хороший фильм")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")

result = nlp("так себе фильм")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")

result = nlp("плохой фильм")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")
```

```
label: POSITIVE, with score: 0.9809
label: NEUTRAL, with score: 0.8302
label: NEGATIVE, with score: 0.7516
```

Задание

1. Взять предобученную трансформерную архитектуру и решить задачу перевода
   (для того же корпуса что вы выбрали из предыдущего дз)

Translation

```
!pip install sacremoses
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting sacremoses
  Downloading sacremoses-0.0.53.tar.gz (880 kB)
     |████████████████████████████████| 880 kB 8.8 MB/s
Requirement already satisfied: regex in /usr/local/lib/python3.7/dist-packages (from sacremoses) (2022.6.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from sacremoses) (1.15.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from sacremoses) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from sacremoses) (1.1.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from sacremoses) (4.64.0)
Building wheels for collected packages: sacremoses
  Building wheel for sacremoses (setup.py) ... done
  Created wheel for sacremoses: filename=sacremoses-0.0.53-py3-none-any.whl size=895260 sha256=387cef4829c7909702005d843c5407e8b669244a2f06c8414aa01e827762d597
  Stored in directory: /root/.cache/pip/wheels/87/39/dd/a83eeef36d0bf98e7a4d1933a4ad2d660295a40613079bafc9
Successfully built sacremoses
Installing collected packages: sacremoses
Successfully installed sacremoses-0.0.53
```

```
from transformers import FSMTForConditionalGeneration, FSMTTokenizer

mname = "facebook/wmt19-en-ru"
tokenizer = FSMTTokenizer.from_pretrained(mname)
model = FSMTForConditionalGeneration.from_pretrained(mname)

input = "Machine learning is great, isn't it?"
input_ids = tokenizer.encode(input, return_tensors="pt")
outputs = model.generate(input_ids)
decoded = tokenizer.decode(outputs[0], skip_special_tokens=True)
print(decoded) # Машинное обучение - это здорово, не так ли?
```

```
Downloading: 100%  624k/624k [00:00<00:00, 944kB/s]
Downloading: 100%  758k/758k [00:00<00:00, 860kB/s]
Downloading: 100%  308k/308k [00:00<00:00, 881kB/s]
Downloading: 100%  67.0/67.0 [00:00<00:00, 1.71kB/s]
Downloading: 100%  827/827 [00:00<00:00, 17.3kB/s]
Downloading: 100%  1.08G/1.08G [00:30<00:00, 49.3MB/s]
Машинное обучение - это здорово, не так ли?
```

Через pipeline

```
mname = "facebook/wmt19-en-ru"
translation = pipeline('translation', model=mname)

# Выводим результат

print(translation("Machine learning is great, isn't it?"))
print(translation("The man worked as a [MASK]."))
print(translation(" In this week's practice, you'll learn how to download, apply and modify pre-trained transformers for a range of tasks. Buckle up, we're going in!"))
```

```
[{'translation_text': 'Машинное обучение - это здорово, не так ли?'}]
[{'translation_text': 'Мужчина работал [MASK].'}]
[{'translation_text': 'На практике на этой неделе вы узнаете, как загружать, применять и модифицировать предварительно обученные трансформаторы для решения целого ряда задач
```

```
mname = "facebook/wmt19-ru-en"
translation = pipeline('translation', model=mname)
```

```
Downloading: 100%  826/826 [00:00<00:00, 14.5kB/s]
Downloading: 100%  1.08G/1.08G [00:40<00:00, 48.7MB/s]
Downloading: 100%  67.0/67.0 [00:00<00:00, 1.67kB/s]
Downloading: 100%  758k/758k [00:00<00:00, 1.53MB/s]
Downloading: 100%  624k/624k [00:00<00:00, 868kB/s]
Downloading: 100%  387k/387k [00:00<00:00, 882kB/s]
```

```
import pandas as pd
```

```
pd.options.display.max_colwidth = None

def translate(sentence):
  tr = translation(sentence)[0]['translation_text']

  final_result.append([sentence, tr])
  # df = pd.DataFrame(final_result, columns=['текст', 'перевод'])

  return


final_result = []
# translate("Machine learning is great, isn't it?")
translate('Здесь хорошо.')
translate('Отлично, поехали.')
translate(u'Вы еще дома?')
translate(u'Это слишком дорого для меня.?')
translate(u'Попробуй сделать это.')
translate(u'Я люблю, когда идет снег.')
translate(u'Я никогда такого не делаю.')
translate('А счастье было так возможно, так близко!.')
translate('Интересно, а если написать длинное предложение и попробовать его перевести, какой результат мы увидим?')

df = pd.DataFrame(final_result, columns=['текст', 'перевод'])
df
```

	текст	перевод
0	Здесь хорошо.	Here's the good thing.
1	Отлично, поехали.	Great, we went.
2	Вы еще дома?	Are you still at home?
3	Это слишком дорого для меня.?	It's too expensive for me.???
4	Попробуй сделать это.	Try it.
5	Я люблю, когда идет снег.	I love when it's snowing.
6	Я никогда такого не делаю.	I never do.
7	А счастье было так возможно, так близко!.	And happiness was so possible, so close!.
8	Интересно, а если написать длинное предложение и попробовать его перевести, какой результат мы увидим?	I wonder, if we write a long sentence and try to translate it, what result will we see?

Задание

2. скачиваем готовый новостной датасет !wget https://github.com/ods-ai-ml4sg/proj_news_viz/releases/download/data/gazeta.csv.gz

```
# пример работы с ним
from corus import load_ods_gazeta
path = 'gazeta.csv.gz'
records = load_ods_gazeta(path)
next(records)
```

реализовать метод поиска ближайших статей (на вход метода должен приходить запрос (какой-то вопрос) и количество вариантов вывода к примеру топ 5-ть или 3-ри, ваш метод должен возвращать топ-k ближайших статей к этому запросу)
визуально оценить качество

```
!pwd
```

```
/content
```

```
!ls -la
```

```
total 16
drwxr-xr-x 1 root root 4096 Jun 29 13:44 .
drwxr-xr-x 1 root root 4096 Jul  4 11:20 ..
drwxr-xr-x 4 root root 4096 Jun 29 13:43 .config
drwxr-xr-x 1 root root 4096 Jun 29 13:44 sample_data
```

```
!wget  https://github.com/ods-ai-ml4sg/proj_news_viz/releases/download/data/gazeta.csv.gz
```

```
--2022-07-04 11:24:46--  https://github.com/ods-ai-ml4sg/proj_news_viz/releases/download/data/gazeta.csv.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/150244024/32420400-b8b5-11ea-8264-2539b75fc310?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
--2022-07-04 11:24:46--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/150244024/32420400-b8b5-11ea-8264-2539b75fc310?X-Amz-Algorithm=AWS4-HMA
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 477029050 (455M) [application/octet-stream]
Saving to: ‘gazeta.csv.gz’

gazeta.csv.gz       100%[===================>] 454.93M  22.2MB/s    in 17s

2022-07-04 11:25:04 (26.8 MB/s) - ‘gazeta.csv.gz’ saved [477029050/477029050]
```

```
# !mkdir data
```

```
# !gunzip gazeta.csv.gz data

# !ls data
```

```
ls: cannot access 'data': No such file or directory
```

```
# !mv gazeta.csv data/gazeta.csv
# !ls
```

```
!pip install corus
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting corus
  Downloading corus-0.9.0-py3-none-any.whl (83 kB)
     |████████████████████████████████| 83 kB 1.8 MB/s
Installing collected packages: corus
Successfully installed corus-0.9.0
```

```
!ls
```

```
gazeta.csv.gz  sample_data
```

```
#  пример работы с ним

from corus import load_ods_gazeta

path = 'gazeta.csv.gz'
records = load_ods_gazeta(path)
next(records)
```

```
NewsRecord(
    timestamp=datetime.datetime(2008, 11, 21, 15, 19, 14),
    url='https://www.gazeta.ru/news/business/2008/11/21/n_1298950.shtml',
    edition=None,
    topics='Бизнес',
    authors=None,
    title='Госдума сокращает срок действия ставки экспортных пошлин на нефть',
    text='Госдума приняла сегодня в первом чтении и сразу в целом поправки в закон «О таможенном тарифе», сокращающие срок действия ставки экспортных пошлин на нефть с 2-х до 1-го месяца.nnДля установления средних цен на нефть марки Urals и расчета экспортных пошлин правительство России в течение двух месяцев проводит мониторинг на международных рынках нефтяного сырья (средиземноморском и роттердамском), экспортные пошлины на нефть устанавливаются также раз в два месяца.nnСокращение на месяц периода мониторинга (с 15-го числа каждого календарного месяца по 14-е число следующего месяца) и соответственно срока действия ставок экспортных пошлин «позволит более оперативно реагировать на изменения экономической ситуации в стране и сэкономить нефтяникам миллиарды рублей», считают разработчики. Экспортные ставки будут вводиться с 1 числа календарного месяца, следующего за окончанием периода мониторинга.',
    stats=Stats(
        fb=None,
        vk=None,
        ok=None,
        twitter=None,
        lj=None,
        tg=None,
        likes=None,
        views=None,
        comments=None
    )
)
```

```
articles = []

for record in records:
   topics = record.topics
   authors = record.authors
   text = record.text
   title = record.title
   articles.append([topics, authors, title, text])

# df = pd.DataFrame(articles, columns=['topics', 'authors', 'title', 'text'])
df.head(3)
```

	topics	authors	title	text
0	Бизнес	None	Госдума сокращает срок действия ставки экспортных пошлин на нефть	Госдума приняла сегодня в первом чтении и сразу в целом поправки в закон «О таможенном тарифе», сокращающие срок действия ставки экспортных пошлин на нефть с 2-х до 1-го месяца.nnДля установления средних цен на нефть марки Urals и расчета экспортных пошлин правительство России в течение двух месяцев проводит мониторинг на международных рынках нефтяного сырья (средиземноморском и роттердамском), экспортные пошлины на нефть устанавливаются также раз в два месяца.nnСокращение на месяц периода мониторинга (с 15-го числа каждого календарного месяца по 14-е число следующего месяца) и соответственно срока действия ставок экспортных пошлин «позволит более оперативно реагировать на изменения экономической ситуации в стране и сэкономить нефтяникам миллиарды рублей», считают разработчики. Экспортные ставки будут вводиться с 1 числа календарного месяца, следующего за окончанием периода мониторинга.
1	Наука	None	Японские физики повторили синтез 113-го элемента	Японские ученые из физического центра RIKEN заявляют, что им удалось синтезировать атом 113-го элемента таблицы Менделеева - этот элемент был впервые получен в 2003 году российскими и американскими физиками в экспериментах по синтезу 115-го элемента, но это открытие еще не признано Международным союзом теоретической и прикладной химии, сообщаетпРИА «Новости»n.nВ природе не существует элементов с атомными номерами (числом протонов в ядре атома) больше 92, то есть тяжелее урана. Более тяжелые элементы, например, плутоний, могут нарабатываться в атомных реакторах, а элементы тяжелее 100-го (фермия) можно получать только на ускорителях, путем бомбардировки мишени тяжелыми ионами. При слиянии ядер мишени и «снаряда» и возникают ядра нового элемента.nnГруппа ученых под руководством Косуке Морит в статье, опубликованной в Journal of Physical Society of Japan, описывают результаты многолетних экспериментов на линейном ускорителе, расположенном в городе Вако в окрестностях Токио. С 2003 года исследователи пытались получить 113-й элемент, бомбардируя на ускорителе мишень из висмута-209 пучком ионов цинка-70, разогнанных до одной десятой скорости света и висмута.nВ результате им удалось зафиксировать три цепочки распада, соответствующие событию рождения 113-го элемента -23 июля 2004 года, 2 апреля 2005 года и 12 августа 2012 года. Время жизни ядра нового элемента составило от 4,9 до 0,3 миллисекунды.nКак считают японские ученые, их открытие может стать основанием для Международного союза теоретической и прикладной химии признать их открытие. В таком случае японские ученые впервые в истории получат право дать название новому элементу. Однако для признания требуется независимое подтверждение - эксперимент должен быть повторен в другой лаборатории или в перекрестной реакции.n«Это выдающийся результат, но воспроизвести его будет крайне трудно, поскольку регистрируется примерно одно событие в год, а перекрестных реакций не существует», - сказал Андрей Попеко, заместитель директора Лаборатории имени Флерова Объединенного института ядерных исследований в Дубне.
2	Армия	None	Times: Россия строит новую авиабазу в Сирии	Россия готовится расширить свою военную операцию в Сирии и строит там вторую авиабазу, сообщает газетаnThe Timesn.nКак сообщает издание, таким образом, у России появится возможность посылать в регион большее количество боевых самолетов. Автор статьи отмечает, что новая база Аль-Шайрат появится недалеко от сирийского города Хомс.nПо словам местного активиста, там уже размещены российские боевые вертолеты, а также команда, которая прибыла на базу около месяца назад и в настоящее время занимается подготовкой объекта.nnРанеепсообщалось,пчто президент России Владимир Путин согласился с предложением Минобороны перебросить на авиабазу Хмеймим в Сирии новейшие ракетные комплексы С-400. Помимо этого, Россияпнаправитпв Сирию дополнительно 10–12 самолетов для обеспечения прикрытия каждому из 24 российских бомбардировщиков.nС 30 сентября Россия начала проводить военную операцию в Сирии, которая, по официальным данным, направлена на уничтожение боевиков «Исламского государства» n«Исламское государство» — террористическая группировка

```
# df[["topics", "title"]].head()
df_text = df[df.title.notna()][["topics", "title"]]
df_text.isnull().values.any()
```

```
False
```

```
# df.topics.unique()
# df["text"].isnull().values.any()
```

```
False
```

```
df.to_csv("df_gazeta.csv")
```

```
# df_gazeta = pd.read_csv('df_gazeta.csv')
# df_gazeta.head()
```

```
df.shape
```

```
(865847, 4)
```

```
import numpy as np
import tensorflow as tf
from transformers import TFAutoModel, AutoTokenizer
```

```
# Загружаем токенайзер и саму модель

bert = TFAutoModel.from_pretrained("Geotrend/bert-base-ru-cased")
tokenizer = AutoTokenizer.from_pretrained("Geotrend/bert-base-ru-cased")
```

```
Downloading: 100%  659/659 [00:00<00:00, 6.05kB/s]

Downloading: 100%  416M/416M [00:13<00:00, 46.2MB/s]

Some layers from the model checkpoint at Geotrend/bert-base-ru-cased were not used when initializing TFBertModel: ['mlm___cls']
- This IS expected if you are initializing TFBertModel from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequer
- This IS NOT expected if you are initializing TFBertModel from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassificati
All the layers of TFBertModel were initialized from the model checkpoint at Geotrend/bert-base-ru-cased.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further training.

Downloading: 100%  49.0/49.0 [00:00<00:00, 818B/s]

Downloading: 100%  130k/130k [00:00<00:00, 329kB/s]
```

```
tok = tokenizer([df["title"][0]],
                max_length=100, truncation=True, padding='max_length', return_token_type_ids=False, return_tensors='tf')
```

```
tok
```

```
{'input_ids': <tf.Tensor: shape=(1, 100), dtype=int32, numpy=
array([[   11,   138,  2684, 13250,   281,   710,   495,   771,  6289,
         9774,  4627,  2424,   553,   192,  2834,  5013,   714,   369,
         3982,   348,   231, 13382,    12,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,
            0]], dtype=int32)>, 'attention_mask': <tf.Tensor: shape=(1, 100), dtype=int32, numpy=
array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], dtype=int32)>}
```

```
import gensim
import gensim.downloader as download_api
```

```
# список предобученных моделей
download_api.info()['models'].keys()
```

```
dict_keys(['fasttext-wiki-news-subwords-300', 'conceptnet-numberbatch-17-06-300', 'word2vec-ruscorpora-300', 'word2vec-google-news-300', 'glove-wiki-gigaword-50', 'glove-
wiki-gigaword-100', 'glove-wiki-gigaword-200', 'glove-wiki-gigaword-300', 'glove-twitter-25', 'glove-twitter-50', 'glove-twitter-100', 'glove-twitter-200',
'__testing_word2vec-matrix-synopsis'])
```

```
# russian_model = download_api.load('word2vec-ruscorpora-300')  # загрузим предтренированные вектора слов из gensim-data
```

```
# list(russian_model.index_to_key)[:10]
```

```
# Загрузить данные
documents = []

documents = df_text["title"].values.tolist()
```

```
# documents[:5]
for line in documents[0:5]:
  print(line)
```

```
Госдума сокращает срок действия ставки экспортных пошлин на нефть
Японские физики повторили синтез 113-го элемента
Times: Россия строит новую авиабазу в Сирии
Власти Египта гарантируют безопасность российским туристам
Гордума Новочеркасска приняла отставку мэра города Анатолия Кондратенко
```

```
# !pip3 install utilities
!pip install utilities-package
```

```
# # -*- coding: utf-8 -*-
# import sys
# import logging
# import os
# import gensim
```

```
# # Представляем doc2vec
# from gensim.models import Doc2Vec
# from utilties import ko_title2words


# # Загрузить данные
# # documents = []
# # Используйте count в качестве "тега" каждого предложения, тег соответствует каждому предложению один к одному
# count = 0
# for line in documents:
#   # Вырезать слова, возвращаемый результат - тип списка
#   words = ko_title2words(title)
#   # Здесь каждый элемент в документах представляет собой двухкортежный кортеж, подробности вы можете проверить в функциональном документе
#   documents.append(gensim.models.doc2vec.TaggedDocument(words, [str(count)]))
#   count += 1
#   if count % 10000 == 0:
#       logging.info('{} has loaded...'.format(count))


# # Модельное обучение
# model = Doc2Vec(documents, dm=1, size=100, window=8, min_count=5, workers=4)
# # Сохранить модель
# model.save('models/ko_d2v.model')


%matplotlib inline
import pandas as pd
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
import jieba as jb
import re
from sklearn import utils
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from gensim.models.doc2vec import TaggedDocument

import multiprocessing


# Doc2Vec Практическая глава текста Мульти-классификация
# https://russianblogs.com/article/7839783911/

# Создать документ с тегами
df_tagged = df_text.apply(lambda r: TaggedDocument(words=r['title'], tags=[r['topics']]), axis=1)



df_tagged.head(3)

    0    (Госдума сокращает срок действия ставки экспортных пошлин на нефть, [Бизнес])
    1                      (Японские физики повторили синтез 113-го элемента, [Наука])
    2                           (Times: Россия строит новую авиабазу в Сирии, [Армия])
    dtype: object


type(df_tagged)

    pandas.core.series.Series


!mkdir models


# Конфигурация модели Doc2vec

# Распределенный мешок слов (DBOW)

# Номер ядра ЦП
cores = multiprocessing.cpu_count()

# Приступим к созданию глоссария

from gensim.models import Doc2Vec
from tqdm import tqdm

model = Doc2Vec(dm=0,  negative=5, hs=0, min_count=2, sample = 0, workers=cores)
model.build_vocab([x for x in tqdm(df_tagged.values)])

# Save the model
model.save('models/ko_d2v.model')

    100%|██████████| 865416/865416 [00:00<00:00, 2373157.39it/s]
```

Обучения модели

```
%%time
epochs = 5
for epoch in range(epochs):
    model.train(utils.shuffle([x for x in tqdm(df_tagged.values)]), total_examples=len(df_tagged.values), epochs=1)
    model.alpha -= 0.002
    model.min_alpha = model.alpha

     100%|██████████| 865416/865416 [00:00<00:00, 3014821.37it/s]
     100%|██████████| 865416/865416 [00:00<00:00, 2904126.06it/s]
     100%|██████████| 865416/865416 [00:00<00:00, 2951757.72it/s]
     100%|██████████| 865416/865416 [00:00<00:00, 3036091.82it/s]
     100%|██████████| 865416/865416 [00:00<00:00, 2951176.94it/s]
     CPU times: user 8min 42s, sys: 21.7 s, total: 9min 4s
     Wall time: 5min 3s


# https://radimrehurek.com/gensim/models/doc2vec.html
# https://russianblogs.com/article/7839783911/
# https://russianblogs.com/article/99251438894/
```

```
def test_doc2vec():
    # Загрузить модель
    model = Doc2Vec.load('models/ko_d2v.model')
    # Наиболее похож на ярлык "0"
    print(model.docvecs.most_similar('Наука'))
    # Провести корреляционное сравнение
    print(model.docvecs.similarity('Наука','Бизнес'))
    # Вывести вектор предложений с меткой «Наука»
    print(model.docvecs['Наука'])
    # Вы также можете вывести вектор предложения (не появляющийся в корпусе)
    # words = u"Госдума сокращает срок действия ставки экспортных пошлин на нефть"
    words = u"Метод Ньютона на языке Голанг для нахождения квадратного корня"
    print(model.infer_vector(words.split()))
    # Вы также можете выводить векторы слов
    print(model[u'Бизнес'])

test_doc2vec()
```

```
       4.13163751e-03 -2.29327753e-03 -1.24182447e-03 -2.71367794e-03
      -1.31458597e-04 -2.04759603e-03 -4.84082568e-03  8.88931390e-04
       2.29352663e-04 -1.22150232e-03 -2.82565947e-03 -1.88622845e-03
      -4.63173678e-03  3.54228541e-03 -4.66716290e-03  3.10206367e-03
       3.99459526e-03 -1.37323898e-03 -1.85564952e-03  2.28223391e-03
      -4.56038211e-03  4.91888968e-05 -1.28283689e-03  1.14929586e-04
       4.05448535e-03  4.92797885e-03  4.40443726e-03 -4.73309867e-03]
     [-4.8626694e-03 -1.8466437e-03 -4.5283404e-03  2.6220018e-03
       4.7104359e-03  8.9034811e-04  3.2977501e-03  6.7712850e-04
       4.6656332e-03  2.7435164e-03  1.2071584e-03  1.0011658e-03
      -2.9354326e-03 -1.6508691e-03 -2.9381267e-03 -6.3149270e-04
       4.6764745e-04 -2.4346637e-03 -1.6485014e-03  4.7256239e-03
      -2.6621178e-03  2.2414555e-03  1.4166966e-03  2.6946743e-03
       2.5584055e-03 -3.9380849e-03 -2.5929553e-03 -9.3422568e-04
       2.2591108e-03 -2.0766635e-03 -3.2098442e-03 -7.0595925e-05
       4.6667224e-03 -2.1109239e-03 -2.7237798e-03 -1.7034274e-03
      -7.3651184e-04  4.6086973e-03  4.1067945e-03  2.7508425e-04
      -2.6044226e-03  2.3315710e-03 -3.1718763e-03 -3.8094309e-03
       2.1485230e-03 -2.7823625e-03 -4.2924243e-03  4.2988211e-03
      -4.6863696e-03 -4.7209058e-03  3.7438392e-03 -3.7901029e-03
      -4.5677760e-05  1.1315237e-03 -2.5882246e-03  1.0299520e-03
      -1.8622089e-03  2.7750551e-03 -8.6591439e-04 -4.5000562e-03
      -1.0046725e-03 -2.9627988e-03  2.9401418e-03  2.7848943e-03
       4.8243706e-03  4.8566577e-03  7.0206652e-04 -3.1626152e-03
       4.5526135e-03  7.9689751e-04 -4.9987966e-03  4.9503022e-03
       4.5446809e-03  1.9827699e-03 -4.7695134e-03  3.2475726e-03
      -2.6780583e-03  3.3993034e-03  3.8851476e-03 -8.6576829e-04
       1.7707549e-04 -4.5086374e-03 -4.4805119e-03  3.0015393e-03
      -1.5670822e-03  4.6426267e-03  3.2528744e-03  4.4374224e-03
       3.7824193e-03  2.7699941e-03  3.1364562e-03 -3.3089349e-03
       2.3600501e-03  4.3433565e-03  2.4950446e-03 -2.7929181e-03
      -1.4268869e-03 -7.2455389e-04 -1.7781268e-03 -1.7882423e-03]
    [-3.15622077e-04 -7.35230977e-04  3.83345323e-05  4.37708246e-03
     -5.93591103e-05 -2.88804551e-03  2.18416168e-03 -3.38260108e-03
     -3.03672464e-03 -9.58652236e-05  1.47658784e-03 -4.35344269e-03
      9.68924782e-04  5.74200123e-04 -1.76404184e-03 -3.45461536e-03
      3.59111698e-03 -3.60710127e-03  4.24651575e-04  1.18195999e-03
     -3.75930755e-03  2.34884719e-04 -2.56848033e-03  3.87210771e-03
     -1.32717460e-03 -1.18688884e-04 -3.50748328e-03 -4.20895265e-03
      3.07406718e-03  1.27731066e-03 -3.93299712e-03 -3.25189950e-03
     -4.41036653e-03 -6.69467379e-04 -1.48020918e-03 -2.86638364e-03
      3.33375111e-03  4.21494339e-03  1.61509146e-03 -2.76678219e-03
      4.85085562e-04 -1.25541096e-03 -2.84588605e-04  3.46373348e-03
     -1.57482480e-03  1.12834503e-03 -1.58161332e-03  2.39335815e-03
      7.89242447e-04 -7.40197196e-04  2.89963651e-03  2.22687726e-03
     -2.45915027e-03 -2.90378719e-03 -3.37493559e-03 -1.31828163e-03
     -2.24431185e-03  1.45657279e-03  2.78001360e-04 -1.05721678e-03
      1.79738994e-03 -1.89888960e-04  4.74948291e-04 -1.18843641e-03
      3.02405516e-03  8.94792378e-04 -3.66577529e-03  3.83272208e-03
      7.60436815e-04  1.89527008e-03  2.90254899e-03  3.88547848e-03
     -3.97512643e-03  3.29856900e-03  4.45918599e-03 -3.73122818e-03
      2.84092152e-03  2.37706467e-03 -4.82835900e-03  4.61401325e-03
     -2.56170309e-03  3.35336044e-05 -4.08109650e-03 -4.56401147e-03
     -3.11749708e-03 -3.97867849e-03 -9.03529231e-04 -1.81666936e-03
      4.05654922e-04 -3.76652810e-03 -1.21914176e-03  1.80768128e-03
      1.35947112e-03 -3.15303379e-03  1.26034301e-03 -3.57201998e-03
      1.04486675e-03  1.89731852e-03  4.55569895e-03  3.88781191e-04]
```

```
# # Infer (выведите) vector for a new document:
vector = model.infer_vector(["system", "response", "Ньютона"])
vector
```

```
array([-0.00122219,  0.00363273,  0.00302503, -0.00140261,  0.00362237,
        0.00019597,  0.00451896, -0.00215742,  0.00310215, -0.00058866,
       -0.00357609, -0.00219439, -0.0011065 ,  0.00331069,  0.00454499,
        0.00424967,  0.00397653,  0.00446123, -0.00348275, -0.00031109,
       -0.00289802,  0.00118728,  0.00480447, -0.00195218,  0.00408714,
       -0.00291135, -0.00172013,  0.00238746, -0.00236066, -0.00154688,
       -0.0011886 , -0.00037868,  0.00343838, -0.00220772, -0.00374576,
       -0.00303615,  0.00336423,  0.00022374, -0.00468001, -0.00476465,
       -0.00262769, -0.00421199, -0.00143737, -0.00429741, -0.00458773,
       -0.00436254,  0.00207585, -0.00129268, -0.00136202, -0.00126066,
        0.00332077,  0.00095704, -0.00280067, -0.0034281 ,  0.00098341,
       -0.00107731,  0.00144922,  0.004834  ,  0.00490574,  0.00404829,
       -0.00213068, -0.00233549, -0.00269263, -0.00491949, -0.00313097,
       -0.00057028, -0.00221833, -0.0045694 ,  0.00329191, -0.00016076,
       -0.00079538,  0.00199139,  0.00190672, -0.00394548, -0.0004729 ,
        0.00309354, -0.00269512, -0.00377755,  0.00323449, -0.00476287,
       -0.00124127, -0.00037502, -0.00390032,  0.00405601, -0.0033523 ,
        0.00459072,  0.00416748,  0.00416351,  0.00295677,  0.00203233,
       -0.00341157,  0.00489519,  0.00052746, -0.00283035, -0.00073521,
        0.00112388,  0.00133379, -0.00393982, -0.00389131,  0.00323933],
      dtype=float32)
```

```
model.corpus_count
```

```
865416
```

```
df_tagged = df_text['title'].values.tolist()
# нахождение наиболее похожего документа
vector_to_search = model.infer_vector(["ищем", "похожий", "текст"])
# три наиболее похожих
```

```
similar_documents = model.docvecs.most_similar([vector_to_search], topn=3)
for s in similar_documents:
  # print(s[0])
  print(df_tagged[s[0]])
```

```
    Политика
    Стиль
    Мнения
```

```
some_texts = df_tagged
```

```
%%time
from gensim.models.doc2vec import Doc2Vec, TaggedDocument

documents = [TaggedDocument(doc, [i]) for i, doc in enumerate(some_texts)]
```

```
    CPU times: user 4.78 s, sys: 377 ms, total: 5.15 s
    Wall time: 5.47 s
```

```
%%time
documents[:5]
```

```
    CPU times: user 6 µs, sys: 0 ns, total: 6 µs
    Wall time: 11 µs
    [TaggedDocument(words='Госдума сокращает срок действия ставки экспортных пошлин на нефть', tags=[0]),
     TaggedDocument(words='Японские физики повторили синтез 113-го элемента', tags=[1]),
     TaggedDocument(words='Times: Россия строит новую авиабазу в Сирии', tags=[2]),
     TaggedDocument(words='Власти Египта гарантируют безопасность российским туристам', tags=[3]),
     TaggedDocument(words='Гордума Новочеркасска приняла отставку мэра города Анатолия Кондратенко', tags=[4])]
```

```
%%time
from multiprocessing import cpu_count
# Кол-во ядер ЦП
cores = multiprocessing.cpu_count()

# Создание и обучение модели Doc2Vec
model_doc2vec = Doc2Vec(documents, vector_size=100, workers=cores, epochs=3)
```

```
    CPU times: user 9min 47s, sys: 40.3 s, total: 10min 27s
    Wall time: 9min 7s
```

```
# сохранение модели для дальнейшего использования
model_doc2vec.save("my_doc2vec_model")

# загрузка модели
model_doc2vec = Doc2Vec.load("my_doc2vec_model")

# нахождение наиболее похожего документа
# vector_to_search = model_doc2vec.infer_vector(["ищем", "похожий", "текст"])
doc="ищем похожий текст"
vector_to_search = model_doc2vec.infer_vector(doc.split())

# пять наиболее похожих
similar_documents = model_doc2vec.docvecs.most_similar([vector_to_search], topn=5)
for s in similar_documents:
    print(some_texts[s[0]])
```

```
    «Элвис Пресли» попытался спровоцировать Овечкина во время матча
    Премьер Украины Гройсман назвал условие своей отставки
    Замглавы администрации Порошенко подал в отставку
    Энтальцев: Айрапетян выступит на ЧЕ по боксу
    Боец Александр Емельяненко провел спарринг с главой Чечни Кадыровым
```

Результат не очень корректный

```
# Обучаем классификатор

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()
clf.fit([model.infer_vector([x.words]) for x in documents], [1, 1, 1, 0, 0])

res = clf.predict([model.infer_vector(['текст', 'номер', 'три'])])
print(res)  # [1]
```

✓ 0 сек. выполнено в 20:59