Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет Инфокоммуникационных технологий (ИКТ)

Образовательная программа Мобильные и сетевые технологии

О Т Ч Е Т

по Лабораторной работе

Дисциплина: Методы сетевого анализа.

Специальность: 09.03.03 Прикладная информатика.

Выполнил:

Балцат К. И.,

студент группы К33401

Санкт-Петербург
2023

ЗАДАНИЕ

Данная лабораторная будет состоять из двух частей: анализ скрытых тем в постах (topic modeling) и генерация постов при помощи дообучения (fine-tuning) модели gpt-2.

Topic Modeling

Topic modeling используется для анализа большого количества документов с целью выявления скрытых тематик, которые могут быть присутствующими в них. Это может помочь в организации и классификации документов, а также в понимании того, какие темы наиболее популярны в определенной области.

Загрузите набор данных, который содержит тексты постов про covid19

Из исходного датасета оставьте только колонку ‘text’ и подготовьте данные для дальнейшей работы, выполнив этапы предобработки из предыдущей лабораторной работы:

Нижний регистр,

Удаление стоп-слов и пунктуации,

Токенизация,

Лемматизация

На основе получившихся данных после предобработки создайте словарь слов и корпус частот их встречаймости с помощью библиотеки gensim

Постройте и обучите модель LDA, передав входные параметры. Определите наиболее оптимальное количество скрытых тем.

Визуализируйте результаты.

Генерация текста. GPT-2

GPT-2 (Generative Pre-trained Transformer 2) - это генеративная модель на основе нейронных сетей, разработанная компанией OpenAI. Она является продолжением и улучшенной версией GPT-1. GPT-2 обучается на огромном корпусе текстов и может генерировать продолжения текстовых

последовательностей, что может быть полезно для создания новых текстов, автоматического перевода, ответов на вопросы и многого другого. GPT-2 обладает высокой точностью и более естественным стилем генерируемых текстов, чем предыдущие модели.

!Перед выполнением смените среду выполнения коллаба на gpu

Загрузите тот же самый датасет из части 1 (topic modeling)

COVID19 Tweets

Приведите тексты к нижнему регистру, удалите пунктуацию и остальные символы

Из исходного количества данных сделайте случайную подвыборку (или не случайную, используя какие-либо эвристики) размером в 5.000 объектов. Мало, зато хватает ОЗУ коллаба, чтобы не вылететь. Если Ваши ресурсы Вам позволяют, то берите подвыборку большим размером.

Загрузите предобученную модель GPT-2 и дообучите на наших данных.

Процесс состоит из:

Импорта библиотек,

Загрузки предобученных моделей и конфигурация специальных токенов,

Добавления токенов в исходные данные,

Токенизации данных,

Обучения модели.

!Для обучения лучше берите меньший batch_size, например 16 (опять же, чтобы хватило памяти в коллабе)

После дообучения модели выведите примеры сгенерированных постов, которые начинались бы с “covid” или “covid19”

Обратите внимание, что при генерации текста модель получает закодированное слово или фразу (tokenizer.encode)и выдает текст в таком же формате, которые нужно декодировать (tokenizer.decode)

ВЫПОЛНЕНИЕ

Модель LDA

```
coherence = 0                                                                                                                    # Начальный коэффиент согласован
for i in tqdm(range(1, 11)):                                                                                                     # Создаём цикл, который рассмотр
  model=LdaMulticore(corpus=corpus, id2word=my_dictionary, num_topics=i, random_state=200)                                       # Обучаем модель LDA
  coherence_model = CoherenceModel(model=model, texts=covid["tokens"].tolist(), dictionary=my_dictionary, corpus=corpus, coherence='c_v')  # Обучаем модель согласованности
  new_coherence = coherence_model.get_coherence()                                                                                # Находим для модели коэффициент
  if new_coherence > coherence:                                                                                                  # Если он больше текущего, то
    coherence = new_coherence                                                                                                    # задаём переменной coherence но
    number = i
print()                                                                                                                          # выводим количество тем
print(coherence)
print(number)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
100%|██████████| 10/10 [03:40<00:00, 22.06s/it]
0.3393510583603389
3
```

Лучший результат показала модель с 3 скрытыми темами

```
model=LdaMulticore(corpus=corpus, id2word=my_dictionary, num_topics=number, random_state=100)                                   # Обучаем модель LDA на полученных
print('\nPerplexity:', model.log_perplexity(corpus))                                                                             # Вывод перплексии
coherence_model = CoherenceModel(model=model, texts=covid["tokens"].tolist(), dictionary=my_dictionary, corpus=corpus, coherence='c_v')  # Обучаем модель согласованности н
coherence_lda = coherence_model.get_coherence()                                                                                  # Находим коэффициент согласованно
print('\nCoherence Score:', coherence_lda)                                                                                       # Выводим коэффициент согласованно
```

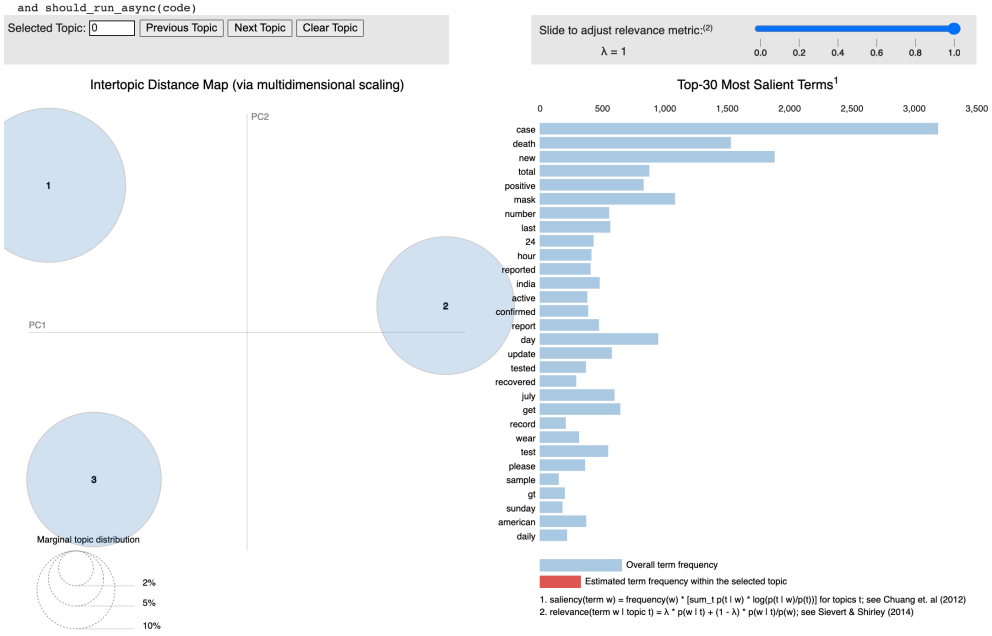
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)

Perplexity: -8.707250339322494

Coherence Score: 0.33459632477955986
```

```
model.print_topics()  # Вывод тем
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
[(0,
  '0.046*"case" + 0.023*"new" + 0.022*"death" + 0.013*"total" + 0.012*"positive" + 0.010*"day" + 0.008*"number" + 0.008*"last" + 0.007*"people" + 0.007*"india"'),
 (1,
  '0.008*"covid19" + 0.006*"get" + 0.006*"u" + 0.004*"new" + 0.004*"one" + 0.004*"2020" + 0.004*"time" + 0.004*"people" + 0.004*"july" + 0.003*"hospital"'),
 (2,
  '0.013*"mask" + 0.007*"people" + 0.006*"pandemic" + 0.004*"like" + 0.004*"u" + 0.004*"covid19" + 0.004*"american" + 0.004*"please" + 0.004*"home" + 0.004*"need"')]
```

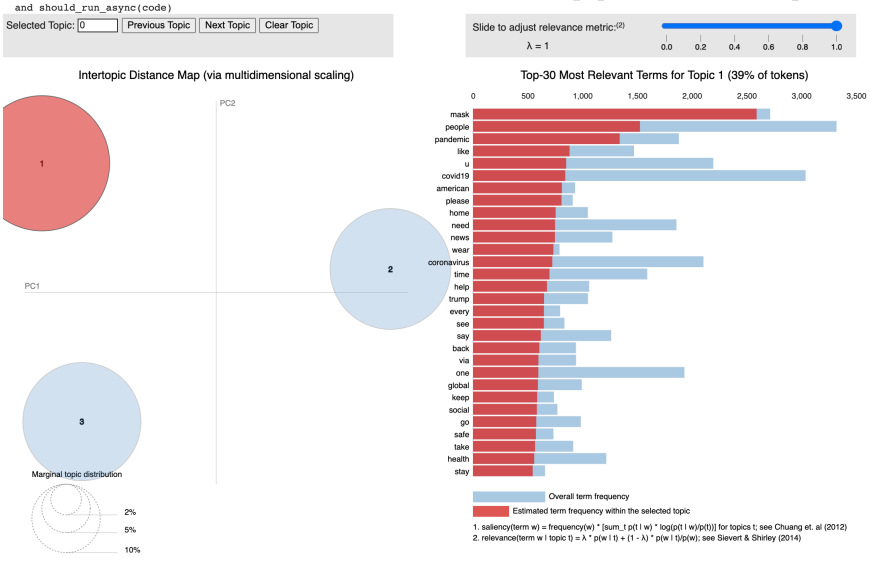
```
pyLDAvis.enable_notebook()                                  # Включение библиотеки pyLDAvis
vis = pyLDAvis.gensim_models.prepare(model, corpus, my_dictionary)  # Подготовка к построению визуализации
vis
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```



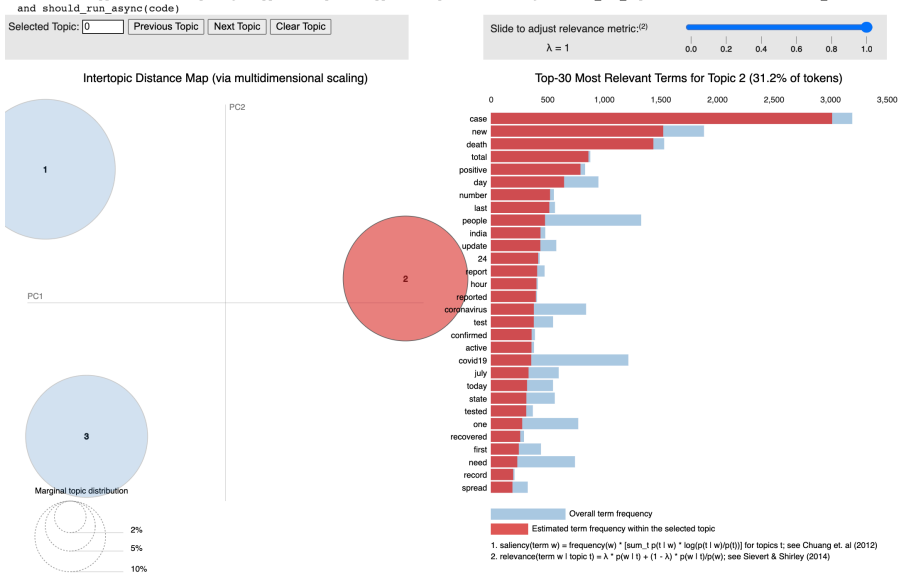
```
pyLDAvis.enable_notebook()                                  # Включение библиотеки pyLDAvis
vis = pyLDAvis.gensim_models.prepare(model, corpus, my_dictionary)  # Подготовка к построению визуализации
vis
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```



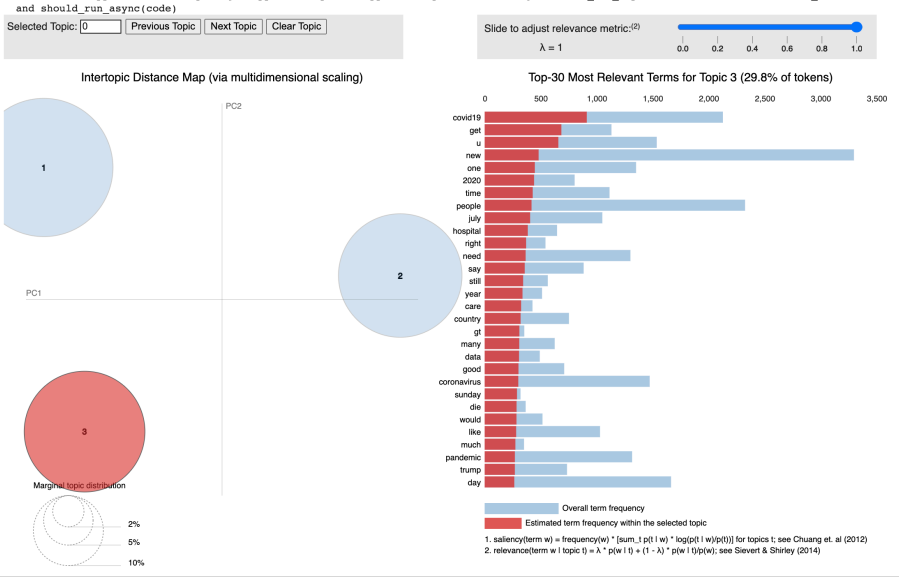

```
pyLDAvis.enable_notebook()                                  # Включение библиотеки pyLDAvis
vis = pyLDAvis.gensim_models.prepare(model, corpus, my_dictionary)  # Подготовка к построению визуализации
vis
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```




```
pyLDAvis.enable_notebook()                                  # Включение библиотеки pyLDAvis
vis = pyLDAvis.gensim_models.prepare(model, corpus, my_dictionary)  # Подготовка к построению визуализации
vis
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```



```
trainer = Trainer(
    model=base_model,                       # the instantiated 🤗 Transformers model to be trained
    args=training_args,                  # training arguments, defined above
    data_collator=data_collator,
    train_dataset=tokenized_train_dataset,         # training dataset
    eval_dataset=tokenized_val_dataset            # evaluation dataset
)
trainer.train()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py:391: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version.
  warnings.warn(
```

[1692/1692 15:57, Epoch 6/6]

Step	Training Loss
500	10.172600
1000	4.652400
1500	4.361800

```
TrainOutput(global_step=1692, training_loss=6.15411694393654, metrics={'train_runtime': 960.4484, 'train_samples_per_second': 56.224, 'train_steps_per_second': 1.762,
'total_flos': 1708523274240000.0, 'train_loss': 6.15411694393654, 'epoch': 6.0})
```

```
trainer.save_model()
base_tokenizer.save_pretrained(model_headlines_path)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
('./model_headlines_news/tokenizer_config.json',
 './model_headlines_news/special_tokens_map.json',
 './model_headlines_news/vocab.json',
 './model_headlines_news/merges.txt',
 './model_headlines_news/added_tokens.json')
```

```
for i in range(len(generated_text_samples)):
  print(f"{i+1}: {headlines_tokenizer.decode(generated_text_samples[i], skip_special_tokens=True)}")
```

```
1: Covid19 incidence in us as of 01 august 2020 487 confirmed cases 745 recovered 486 recoveries 1188 recovered deaths
2: Covid19 situation in germany continues to deterior
3: Covid19 testing centre in honduras tests positive for covid 19
4: Covid19 rate of covc in maharashtra reports highest spike of 662467 cases surpassing previous record for first time
5: Covid19 vaccine has proven to be a huge boost for many families in the developing world who do have their children
6: Covid19 is now the 2nd leading cause of death worldwide in a single day
7: Covid19 update for the mid 2020 saturday august 7th 68949 cases reported in total 1248 gmt
8: Covid19 vaccine trial in india
9: Covid19 vaccine is not a panacea for the problem of covids in particular as well it does contain deadly i
10: Covid19 vaccine in children
11: Covid19 is just the beginning of what will be a daunting challenge when it comes to handling coronavirus cases in all parts o
12: Covid19 is an exciting new finding which explains why covids may not be the only factor in predicting the future
13: Covid19 has been linked to the spread of and it remains uncertain whether in people
14: Covid19 update global cases 84843 742 deaths 595 recovered 214820 767 cases reported d
15: Covid19 impact on economic activity including employment and mobility due to covid 19
16: Covid19 in the news
17: Covid19 has not only killed and sickened many more people but has
18: Covid19 response guidance for physicians and pharma industry with comments from the president of c
19: Covid19 level at 3rd degree point is like a fever without heat
20: Covid19 in human blood plasma from patients is much lower than the level of acute infection
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```

```
for i in range(len(generated_text_samples1)):
  print(f"{i+1}: {headlines_tokenizer.decode(generated_text_samples1[i], skip_special_tokens=True)}")
```

```
1: Coronavirus is now the second most deadly infectious disease worldwide after
2: Coronavirus via
3: Coronavirus can damage the liver and kidney because of a lack o
4: Coronavirus covid19 cases in india total number of confirmed positive cases as on august 27 2020 24
5: Coronavirus covid19 cases in india rise to 489747
6: Coronavirus s work to fight infections in 2020
7: Coronavirus covid19 updates in april
8: Coronavirus situation in australia after coronation and pandemic
9: Coronavirus virus has been linked to a range
10: Coronavirus is a global pandemic problem that impacts over 500 million people worldwide according to the latest figures
11: Coronavirus covid19 death toll is still at a ppe level so it s possible that we may have crossed the threshold
12: Coronavirus i m the only one with a vaccine i know how to tell if you have it or not i love these i m
13: Coronavirus is the virus that causes severe disabilities such as mobility and
14: Coronavirus in the us has reached a new low after experiencing an unprecedented surge since midweek we re calling for more c
15: Coronavirus in humans
16: Coronavirus is the world's deadliest coronoviruses have been discovered on a bus in india the virus has not shown any symptoms
17: Coronavirus and covid19 virus related deaths toll rises to 230
18: Coronavirus will not work on all people or countries is a myth the virus that we ve had this whole time s
19: Coronavirus update on the state of covid19 in india
20: Coronavirus is the worst economic crisis since july 23 and all other countries are hit hard by it the second biggest in europe
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please
  and should_run_async(code)
```

ВЫВОД

Я выполнил лабораторную работу и на практике освоил методы анализа сетей: LDA и дообучение трансформера GPT2 и генерация текста.