Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по заданию №9
«Кластеризация»

по дисциплине «Компьютерная лингвистика»

Автор: Лакиза Александр Николаевич

Факультет: ИКТ

Группа: К3242

Преподаватель: Чернышева Анастасия Владимировна

Санкт-Петербург 2021

Цель работы: произвести кластеризацию своего корпуса по именованным сущностям

Ход работы:

Весь использованный функционал библиотеки natasha был взят мной из файла NER, LDA

Сначала подготовил свои данные

```
with open("../data/corpus_as_dict.json") as f:
    docs = json.load(f)
titles = list(docs.keys())
texts = list(docs.values())
df = pd.DataFrame.from_dict({'title':titles, 'text':texts})
```

Далее воспользовался функцией get_ner

```
def get_ner(transcript):
  script = Doc(re.sub(r'\((.*?)\)', "", transcript))
  script.segment(segmenter)
  script.tag_morph(morph_tagger)
  for token in script.tokens:
    token.lemmatize(morph_vocab)
  script.tag_ner(ner_tagger)
  for span in script.spans:
    span.normalize(morph_vocab)
  named_ents = [(i.text, i.type, i.normal) for i in script.spans]
  normed_ents = []
  for word, tag, norm in named_ents:
    if len(word.split()) == 1 and tag == "LOC":
      for gram in range(len(analyzer.parse(word))):
        if "Geox" in analyzer.parse(word)[gram].tag:
          normed_ents.append((analyzer.parse(word)[gram].normal_form))
          break
        elif gram == len(analyzer.parse(word)) - 1:
          normed_ents.append((norm.lower().strip(".,!?;-")))
    else:
      normed_ents.append((norm.lower().strip(".,!?;-")))
  return sorted(normed_ents)
```

Далее я посчитал, что из 363 документов лишь в 223 есть именованные сущности, именно с этими документами дальше и работал

```
vocabulary = sorted(set(ner_voc))
corpus = df_ner.named_entities.apply(str).tolist()

```

```
pipe = Pipeline([('count', CountVectorizer(vocabulary=vocabulary)),
                 ('tfid', TfidfTransformer())]).fit(corpus)
X = pipe.fit_transform(corpus)
km = KMeans(n_clusters=18, init='k-means++', max_iter=600,
            algorithm="full", precompute_distances=True)

km.fit(X)
```

Затем оценил свою кластеризацию

```
print(metrics.silhouette_score(X, km.labels_, sample_size=1000))
print(metrics.davies_bouldin_score(X.toarray(), km.labels_))
```

Результат:

```
0.0868968053447224
1.3695426552227496
```

Вообще, мне кажется странным судить по данным метрикам, так как первый показатель всё время колеблется у одинакового значения, а второй становится больше (что плохо) при уменьшении кол-ва кластеров и наоборот уменьшается при увеличении числа кластеров

После предсказания получилось следующее распределение по кластерам:

```
2     170
13     10
14      8
3       7
12      4
15      4
17      3
6       3
8       3
5       2
11      2
16      1
0       1
10      1
1       1
7       1
4       1
9       1
```

(В первом столбце номер кластера, во втором – кол-во документов, относящихся к нему)

Посмотрим на результат кластеризации

```
Cluster 0: солонина японское море ехидну заморская территория залив белфас
т зал слава закавказье жюль мишле жюль ардуэн-мансар жоау фернандеша лэвра
дур
Cluster 1: итчен тест портсмут англия великобритания европа европейский со
юз евразия заморская территория залив белфаст
Cluster 2: канада уайт сша европа титаник великобритания белфаст франция м
орзе кунард
Cluster 3: сша америка nautilus вмс штат бифштекс парадайз кларк стрип нев
ад
```

```
Cluster 4: англия шотландия нидерланды оранский японское море ж. картье за
морская территория залив белфаст зал слава закавказье
Cluster 5: атлантика северная азовское евразия шпицберген черное америка а
тлантический жоау фернандеша лэврадур жан кальвин
Cluster 6: гренландия америка северная земля атлантический южная австралия
европа исландия антарктида
Cluster 7: непотопляемость трюм японское море ж. б. ламарком запад заморск
ая территория залив белфаст зал слава закавказье жюль мишле
Cluster 8: испания мадрид европа африка португалия андорра марокко канарск
ие гибралтар нато
Cluster 9: массачусетс бостон содружество англия атлантический сша жан кал
ьвин женева жоау фернандеша лэврадур японское море
Cluster 10: эллипс западное полушарие западная европа запад заморская терр
итория залив белфаст зал слава закавказье жюль мишле жюль ардуэн-мансар
Cluster 11: конгресс сша японское море ехидну заморская территория залив б
елфаст зал слава закавказье жюль мишле жюль ардуэн-мансар
Cluster 12: россия российская англия сша эстония ленинград нева снг кронш
адт эрмитаж
Cluster 13: великобритания ирландия лондон англия северная шотландия уэльс
георг уайтхолл европа
Cluster 14: земля луна российская солнце марс тейи венера арктика меркурий
жан кальвин
Cluster 15: европа париж шотландия франция ришелье сена азия океания зевсо
м евразия
Cluster 16: грирсон японское море западное полушарие запад заморская терри
тория залив белфаст зал слава закавказье жюль мишле жюль ардуэн-мансар
Cluster 17: франция париж республика сена нато парижии сите евросоюз верса
ль оон
```

Видно, что в отличие от тематического моделирования из Задания 10, тут можно в каждом кластере примерно четко определить тему, что говорит нам о хорошем результате. Число кластеров – 18 взято, можно сказать, случайно. При любом кол-ве кластеров, тематика каждого кластера хорошо прослеживается

Ссылка на исходный код и json файл с корпусом:
https://github.com/alexanderlakiza/cs224/tree/main/task9