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

Отчет

по заданию №10
«Тематическое моделирование LDA»

по дисциплине «Компьютерная лингвистика»

Автор: Лакиза Александр Николаевич

Факультет: ИКТ

Группа: К3242

Преподаватель: Чернышева Анастасия Владимировна

Санкт-Петербург 2021

Цель работы: произвести тематическое моделирование корпуса на основе LDA

Ход работы:

Перед тематическим моделированием необходимо было проделать следующие операции:

Токенизация, лемматизация, нормализация, удаление стоп-слов, удаление служебных частей речи (не вошедших в стоп-слова)

Тоценизация, лемматизация и нормализация мною были уже сделаны и в файле **corpus_as_dict_of_norms.json** у меня уже хранился предобработанный корпус.

```
with open("../data/corpus_as_dict_of_norms.json") as f:
    normed_dict = json.load(f)
```

Затем я составил функцию для удаления слов на английском и стоп слов

```
def del_stops(lst):
    """
    Функция удаления стоп-слов
    """
    for word in lst:
        word = re.sub("[^а-яА-ЯёЁ]"," ", re.sub(r'\((.*?)\)', "", word))
    lst = [word for word in lst if not re.match(r'[a-zA-Z]+', word)]  # Удаление английских слов
    lst = [word for word in lst if morph.parse(word)[0].tag.POS not in ['CONJ', 'PRCL', 'INTJ', 'PREP', 'NPRO', 'NUMR']]
    # Удаление служебных частей речи и личных местоимений
    stops = stopwords.words("english") + stopwords.words("russian") + ["это", "который", "наш", "мочь", "год",
                                                "такой", "знать", "мы", "свой", "один", "другой", "хотеть",
                                                "человек", "всё", "все", "весь", "очень", "думать", "нужно",
                                                "большой", "время", "использовать", "говорить", "сказать",
                                                "иметь", "сделать", "первый", "каждый", "день", "её", "ваш",
                                                "стать", "больший", "ваше", "день", "самый", "понять",
                                                "просто", "ещё", "проблема", "также", "например", "м", "с"]
    return [w for w in lst if w not in stops]

for i in range(len(corpus)):
```

```
20.     corpus[i] = del_stops(corpus[i])
```

Затем я подготовил корпус к тематическому моделированию, создав мешок слов

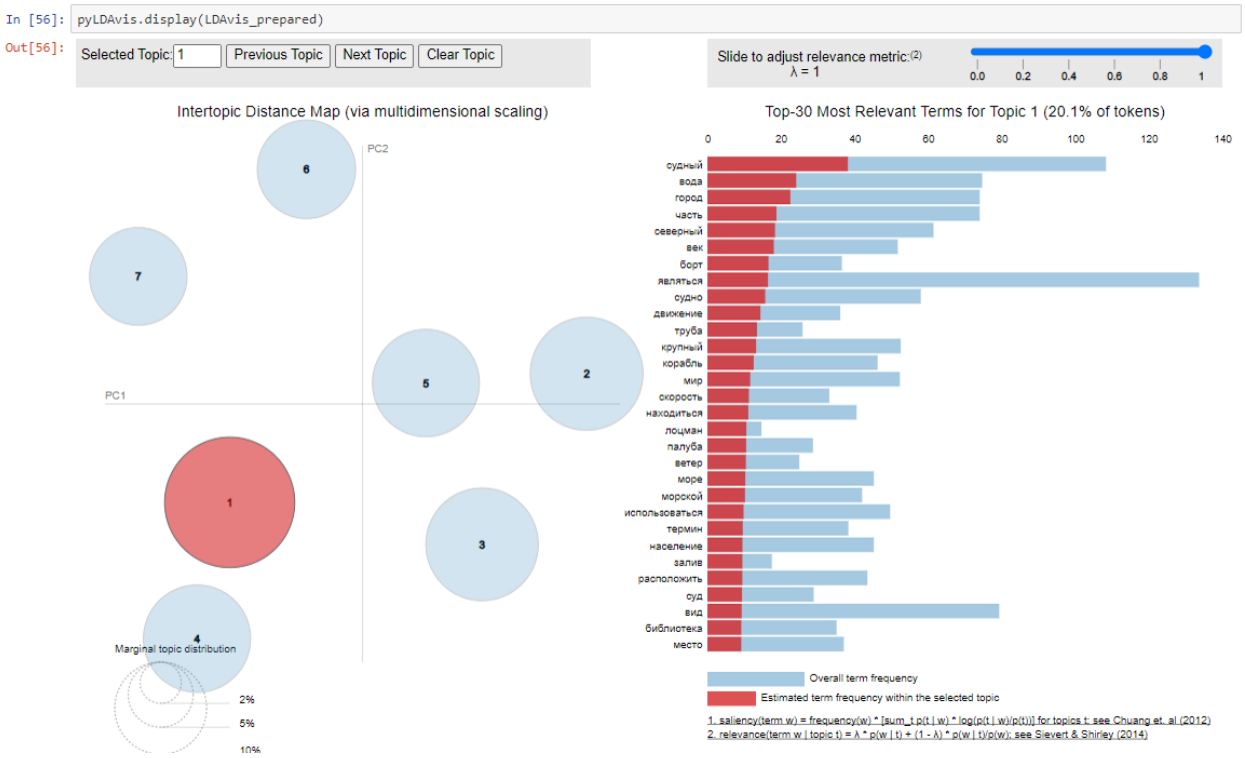
```
1. id2word = corpora.Dictionary(corpus)
2. docs = corpus
3. corpus = [id2word.doc2bow(doc) for doc in docs]
4. from gensim.models import LdaMulticore
5. lda_model = LdaMulticore(corpus=corpus, id2word=id2word, num_topics=7)
```

И с помощью библиотек genism и pyLDAvis произвёл моделирование и визуализировал его

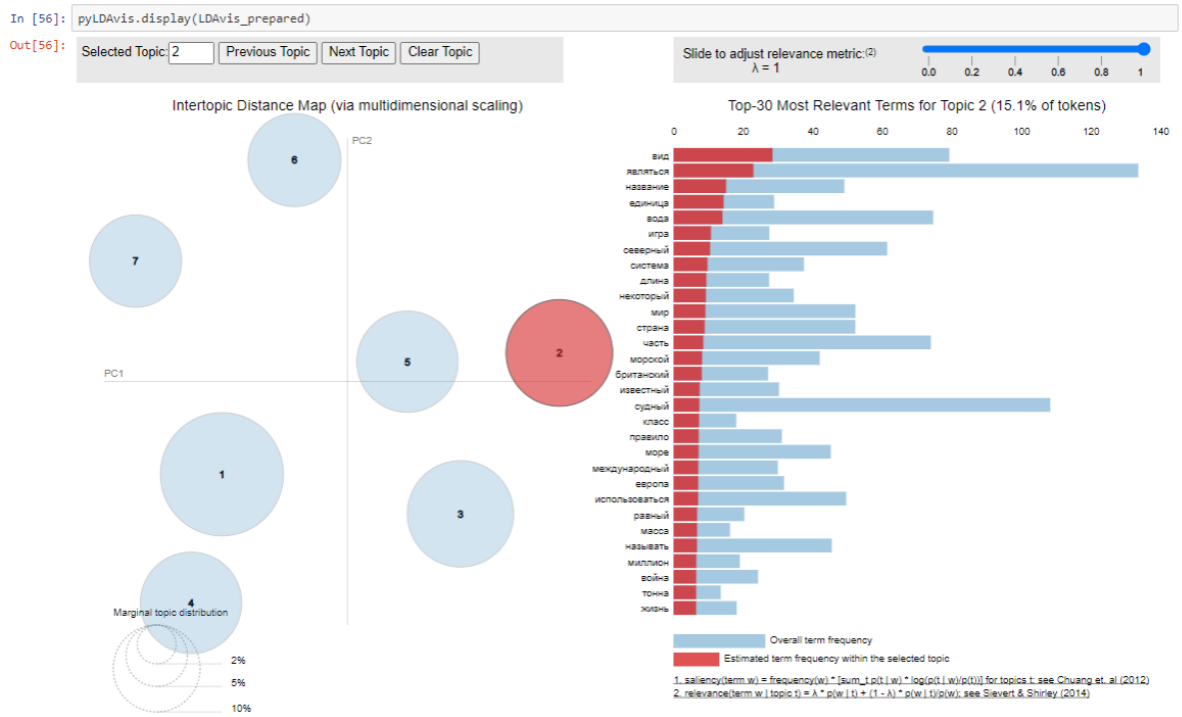
```
1. pyLDAvis.enable_notebook()
2. LDAvis_prepared = pyLDAvis.gensim_models.prepare(lda_model, corpus,
   id2word)
3. pyLDAvis.display(LDAvis_prepared)
```

Как я написал в самом коде, данное моделирование вряд ли можно назвать удачным, так как на сколько бы тем я не делил, темы всё равно чётко не прослеживаются. Я в итоге решил выбрать параметр: 7 тем, так как это максимальное кол-во, при котором круги не пересекаются.

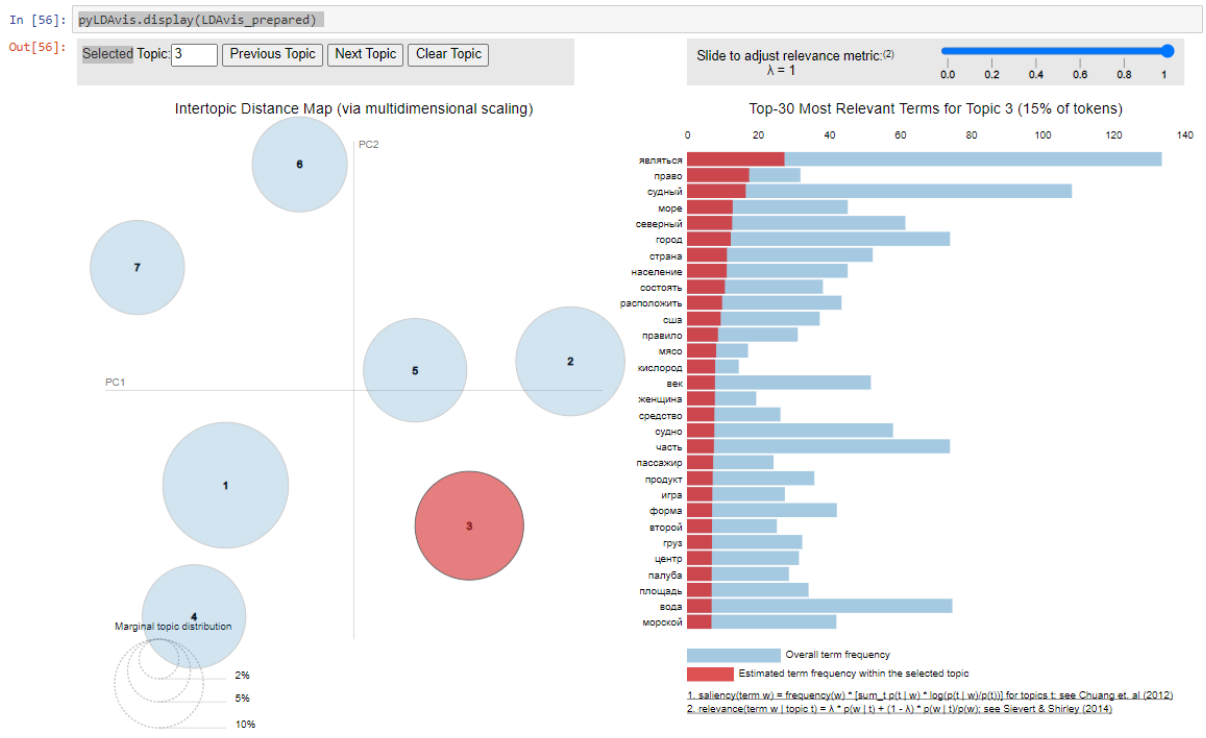
1 тема: Я бы назвал: «Самые примитивные слова про судно и город»



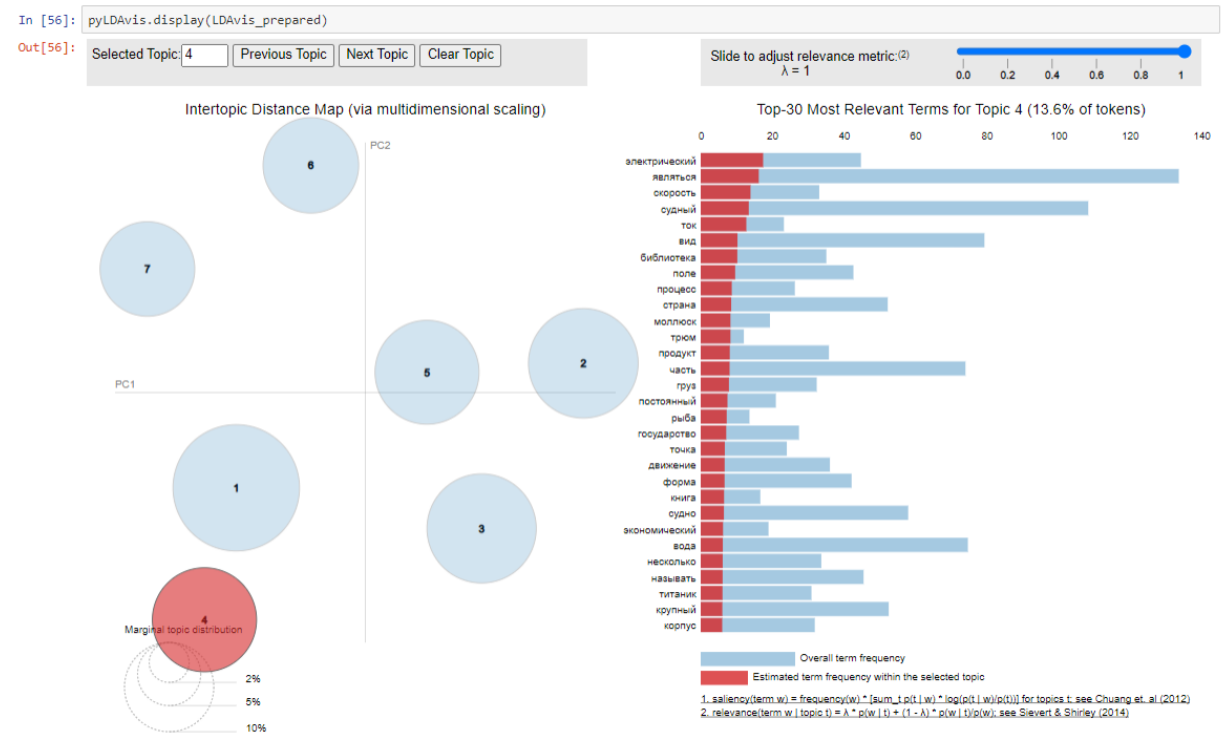
2 тема: Я бы назвал «Что-то про системы, немного про корабли»



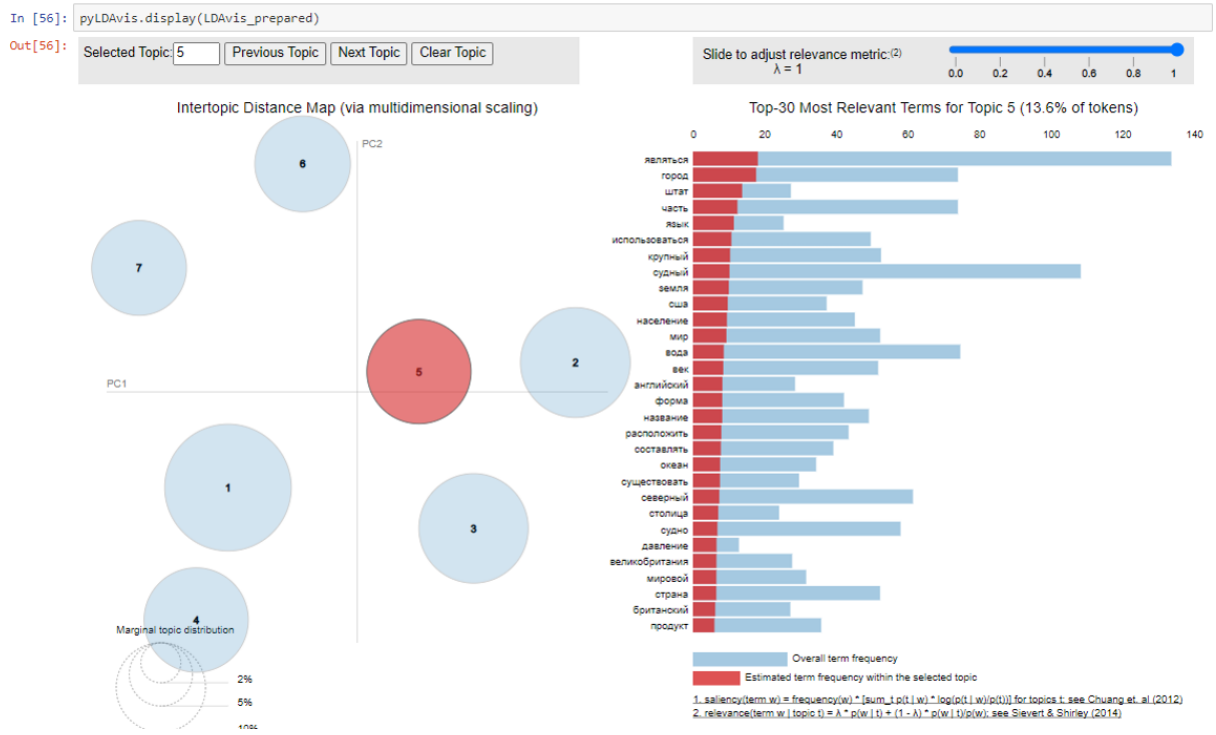
3 тема: «Город, судно, немного про общество»



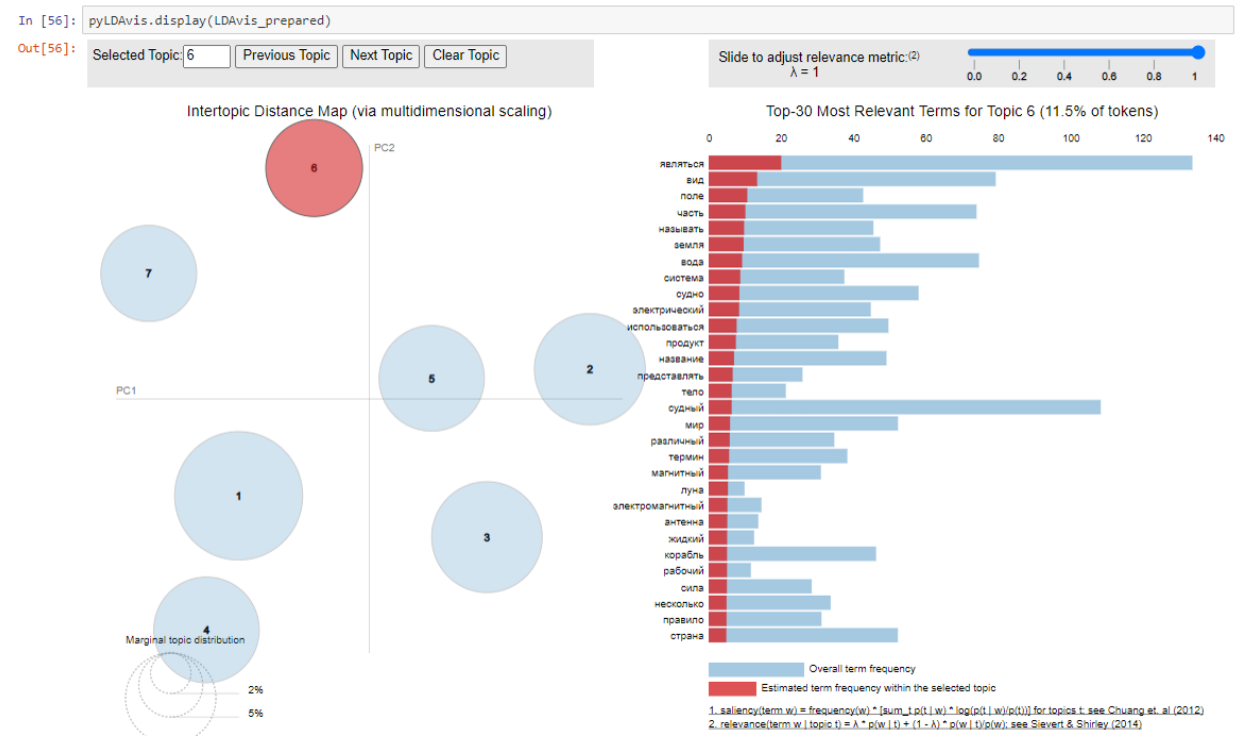
4 тема: «Что-то про физику и море»



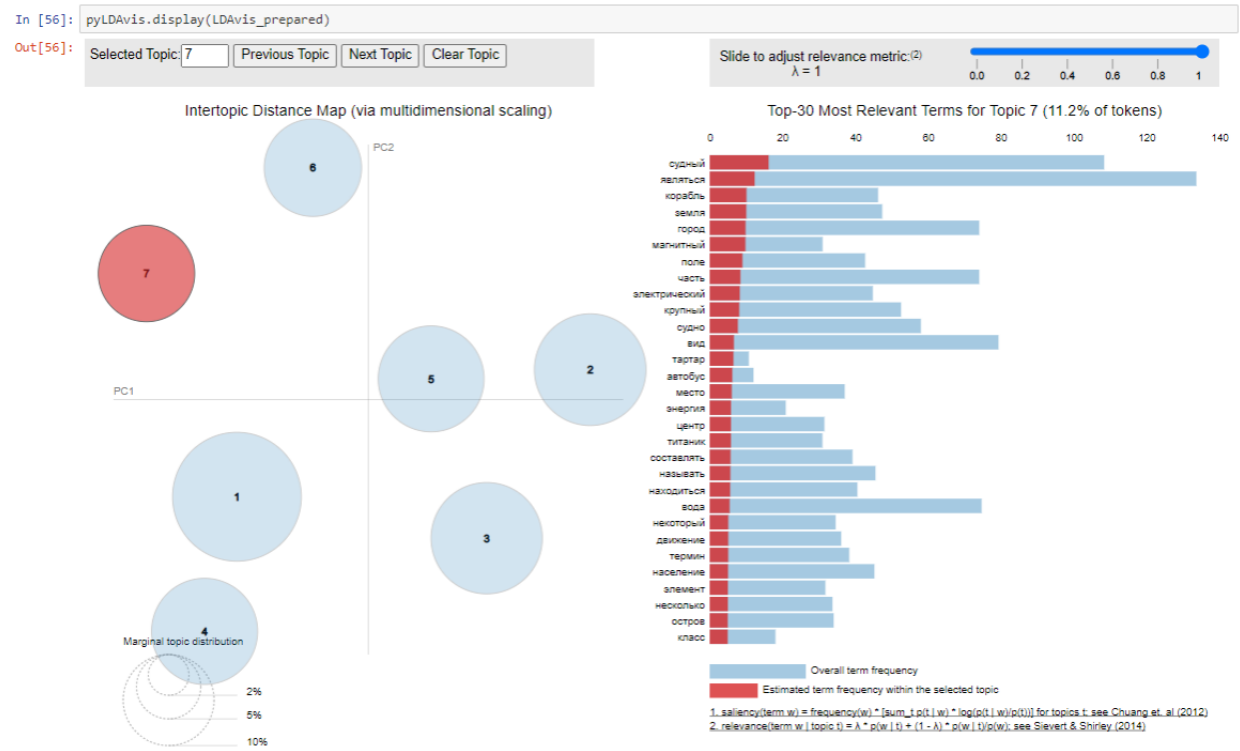
5 тема: «Государство, общество и корабль»



6 тема: «Немного про биологию, среды, физику, корабли»



7 тема: «Физика, транспорт, движение»



Как мы видим, выделить четкие темы на основе слов очень сложно выделить, в отличие от кластеризации, проведенной мной в Задании 9

Ссылка на исходный код и json файл с корпусом:
https://github.com/alexanderlakiza/cs224/tree/main/task10