ОПРЕДЕЛЕНИЕ ИДЕНТИЧНОСТИ СОВРЕМЕННОГО ВУЗА: АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ КЛЮЧЕВЫХ СЛОВ И ТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РУССКОЯЗЫЧНЫХ МЕДИАТЕКТОВ

Татьяна Евтушенко
К.филол.наук, доцент
Санкт-Петербургский политехнический университет Петра Великого

КЛ 2022-2023 ВШЭ

Постановка общей задачи

Цель - провести анализ медиатекстов на русском языке методом извлечения ключевых слов и тематического моделирования и разработать алгоритм для определения идентичности российского ВУЗа на основе контент-анализа медиатектов.

Исследовательский вопрос

1. Как идентичность проявляется в тех текстах, которые публикуются на сайтах вузов?
2. Какие ресурсы лучше использовать и какие значения параметров?

Практическая и теоретическая значимость

Алгоритм исследования идентичности ВУЗов может быть использован для исследования идентичности разных учреждений, что важно для понимания того, как учреждение позиционирует себя в интернет-пространстве.

К какой задаче /задачам NLP можно свести

Обработка естественного языка: частотный анализ лексики

Категоризация текстов

- Автоматическое извлечение ключевых слов (TfIdf)
- Тематическое моделирование (LDA, RAKE, Yake)

Общая схема решения задачи

- Составление перечня ВУЗов, с сайтами которых будем работать
- Загрузка данных (датасет из 420 медиатекстов)
- Предобработка данных
- Извлечение ключевых слов (униграммы, 2-граммы, 4-граммы)
- Векторизация
- Тематическое моделирование
- Загрузка данных в датафрейм
- Выявление общих и специфических тем для современных ВУЗов
- Определение оптимальных значений параметров (количество тем, ключевых слов)
- Проанализировать частотные 2-или 4-граммы для уточнения тем кластеров.
- Сделать визуализацию (R)

- Сделать синтаксическую разметку текстов.
- Сделать контент-анализ медиатекстов на основе конкордансов.

Библиотеки

```
#@title Импорт библиотек
import os
import pandas as pd
from nltk import word_tokenize

import nltk
nltk.download('punkt')
from nltk.tokenize import WordPunctTokenizer

nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords = stopwords.words('russian')

import string

!pip install pymorphy2
from pymorphy2 import MorphAnalyzer
morph = MorphAnalyzer()

from collections import Counter

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

import numpy as np

from sklearn.decomposition import LatentDirichletAllocation

!pip install pyldavis

import pyLDAvis
#import pyLDAvis.sklearn

import matplotlib.pyplot as plt
```

Сбор и обработка данных (1)

Материал исследования

Корпус из 420 новостных текстов с сайтов университетов

- Классические СПбГУ, БГУ, ТГУ
- Технические: СПбПУ, ТПУ
- Транспортные: ДВГУПС
- Медицинские: САМГМУ
- Военные: ВАС

ФУНКЦИЯ ДЛЯ СБОРА ДАННЫХ

```
def get_data():
    corpus = []
    universities = []
    type_uni = []
    dirpath = '/content/drive/MyDrive/КЛ ФПК/тексты_ВУЗы'
    for filename in os.listdir(dirpath):
        new_path = dirpath+ "/" + filename
        type_ = [k for k, v in uni_types.items() if filename in v][0]
        print(type_)
        for new_filename in os.listdir(new_path):
            if new_filename.endswith("data.txt"):
                text_path = new_path + "/" + new_filename
                with open (text_path, 'r', encoding = 'utf-8') as f:
                    text = f.read()
                    corpus.append(text)
                    universities.append(filename)
                    type_uni.append(type_)
    return corpus, universities, type_uni
```

Сбор и обработка данных (2)

- Сбор и загрузка (Requests, Beautiful Soup)
- Nltk (токенизация)
- Pymorphy (лемматизация)
- Датафреймы (pandas)
- Векторизация (sklearn)
- Синтаксическая разметка (natasha)

```
def preprocess_data(text):
    text = text.lower()
    text_tokens = WordPunctTokenizer().tokenize(text)
    spec_chars = string.punctuation + '\n\xa0«»\t—...'
    text_tokens = [token for token in text_tokens if
                   (token not in stopwords and token not in spec_chars and not any(char.isdigit() for char in token))]

    text_lemmatized = [morph.parse(token)[0].normal_form for token in text_tokens]

    return " ".join(text_lemmatized)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

```
[ ] df["clean_text"] = df["text"].apply(preprocess_data) #добавляем столбец в датафрейм
```

Извлечение ключевых слов

ВЕКТОРИЗАЦИЯ

```
[ ]  tfidf_vectorizer = TfidfVectorizer(ngram_range=(1,2)) # биграммы (если 1,4, то весь диапазон)
     tfidf = tfidf_vectorizer.fit_transform(df["clean_text"].tolist())
     feature_names = np.array(tfidf_vectorizer.get_feature_names_out())
```

```
     def get_top_tf_idf_words(text, tfidf_vectorizer, feature_names, top_n):
         tfidf_vector = tfidf_vectorizer.transform([text])
         sorted_nzs = np.argsort(tfidf_vector.data)[:-(top_n+1):-1]
         return feature_names[tfidf_vector.indices[sorted_nzs]]
```

ЗАГРУЗКА КЛЮЧЕВЫХ СЛОВ ДЛЯ КАЖДОГО ВУЗА? В ТАБЛИЦУ

```
[ ]  df["top_words"] = df["clean_text"].apply(get_top_tf_idf_words, tfidf_vectorizer = tfidf_vectorizer, feature_names = feature_names, top_n = 20)
```

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

```
[ ]  # df['top_words'] = df['clean_text'].apply(lambda x: get_top_tf_idf_words(x, tfidf_vectorizer, feature_names, 20))
```

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

Фрагмент таблицы с данными (pandas)

```
[ ] df.sample(5)
```

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

	text	uni	uni_type	clean_text	top_words
315	1 июля 2022 года в Институте дополнительного о...	ДВГУПС	transport	июль год институт дополнительный образование д...	[ступень, пусть выбрать, диплом слушатель, мен...
43	С 28 по 30 апреля на велотреке «Сатурн» в Пенз...	СибСпорт	sport	апрель велотрек сатурн пенза пройти iii этап к...	[соревновательный день, соревновательный, фина...
358	Тестовая сессия состоялась в Университете Сиен...	СПбГУ news_events	classic	тестовый сессия состояться университет сиена б...	[язык, русский язык, русский, тестирование, эк...
391	4 сентября на плацу Военной академии связи про...	ВАС	milit	сентябрь плац военный академия связь пройти це...	[клятва, кадет, клятва кадет, слово, кадетский...
384	13 марта 2022 года, в Неделю торжества правосл...	ВАС	milit	март год неделя торжество православие курсант ...	[собор, бог, суворов, леонид, курсант, академи...

Модель (LDA) для всех университетов

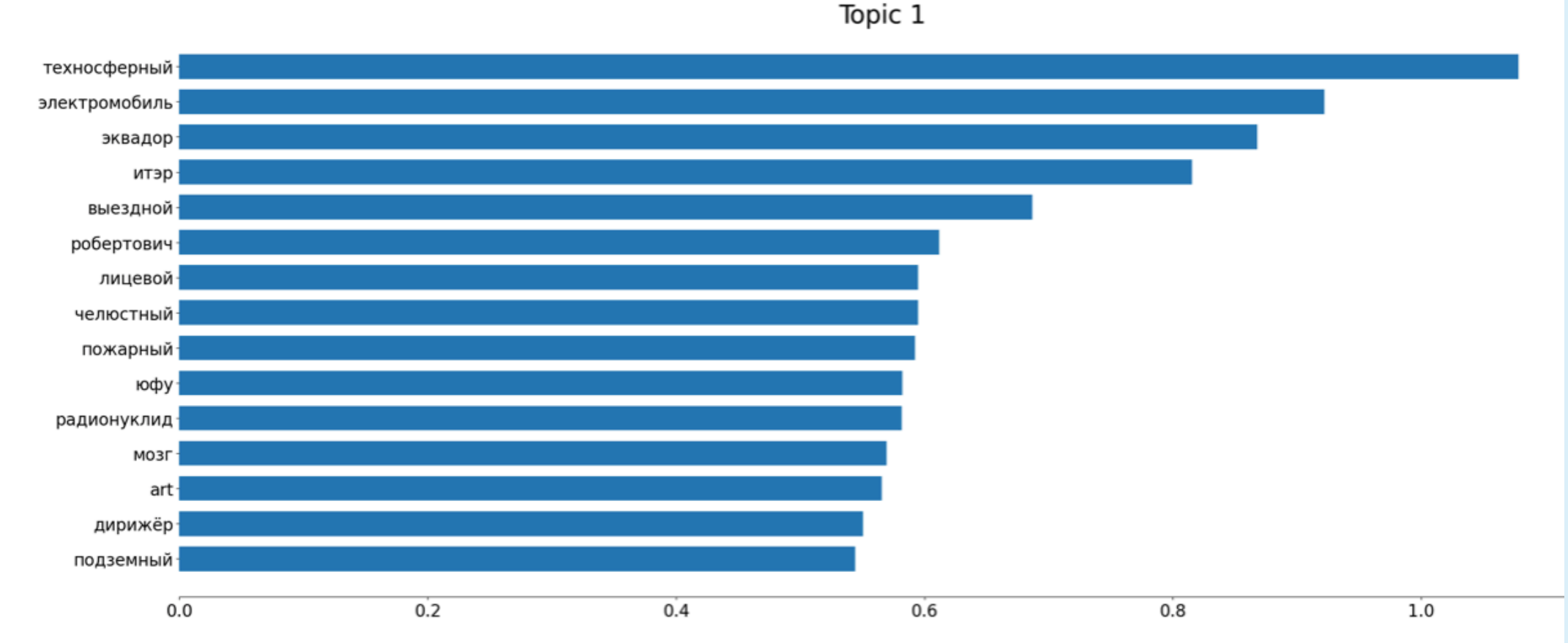
```
df_unitype_topics = pd.DataFrame(columns = ["Unitype", "Topic_words"]) # создаем табличку с перечнем слов


for unitype in uni_types:
    tfidf_vectorizer = TfidfVectorizer(ngram_range=(1,2))
    tfidf = tfidf_vectorizer.fit_transform(df[df["uni_type"] == unitype]["clean_text"].tolist())
    feature_names = np.array(tfidf_vectorizer.get_feature_names_out())

    lda = LatentDirichletAllocation(n_components=7, max_iter=10,learning_offset=10)
    lda.fit(tfidf)

    n_top_words = 15
    for topic_idx, topic in enumerate(lda.components_):
        top_features_ind = topic.argsort()[:-n_top_words - 1:-1]
        top_features = [feature_names[i] for i in top_features_ind]
        dict_topics = {"Unitype": unitype, "Topic_words": top_features}
        df_unitype_topics = df_unitype_topics.append(dict_topics, ignore_index = True)
```

Визуализация данных (на всем корпусе)

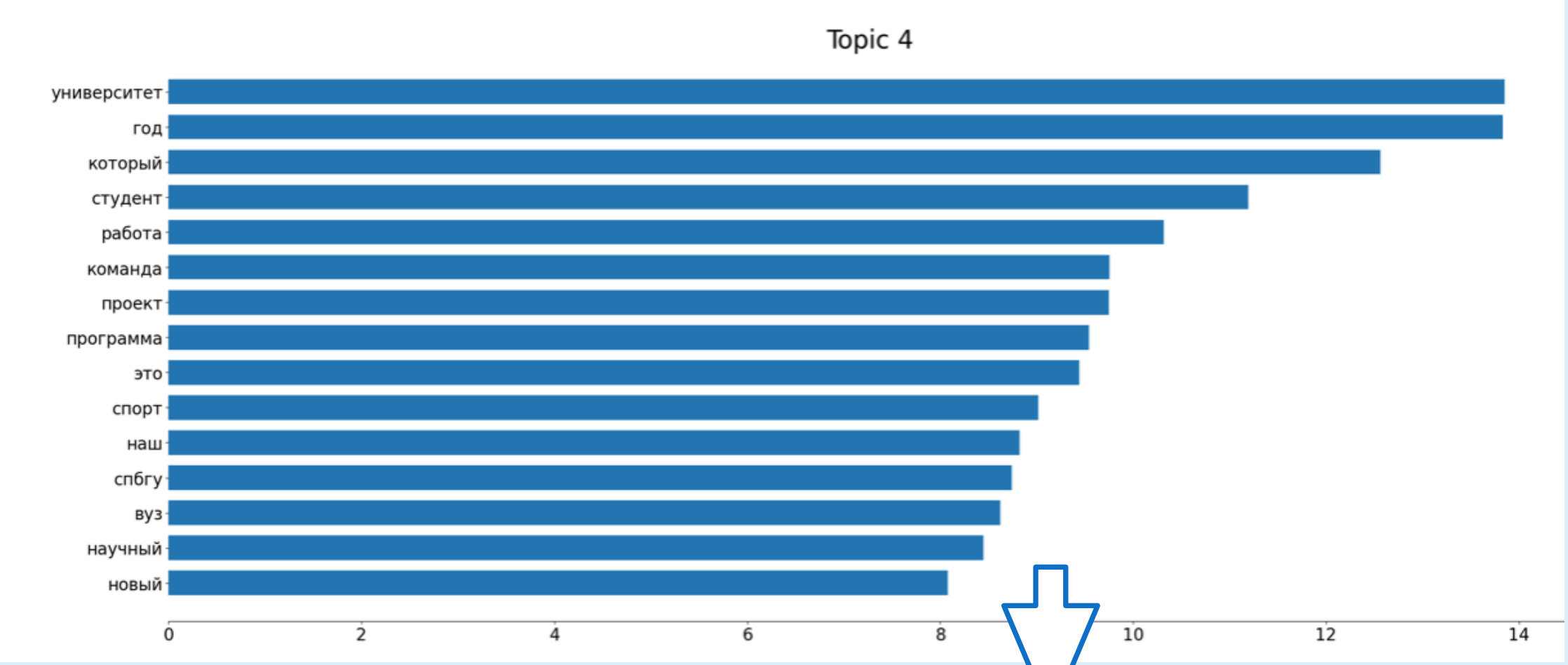
```
def plot_top_words(model, feature_names, n_top_words, title):

    fig, axes = plt.subplots(7, 1, figsize=(30, 100)) # параметры отображения # 7строки по 1 столбцов
    axes = axes.flatten()
    all_features = {} # словарь для сохранения ключевых слов для тем

    for topic_idx, topic in enumerate(model.components_):
        top_features_ind = topic.argsort()[:-n_top_words - 1:-1]
        top_features = [feature_names[i] for i in top_features_ind]
        # строка для сохранения темы и слов в словарь

        weights = topic[top_features_ind]

        ax = axes[topic_idx]
        ax.barh(top_features, weights, height=0.7)
        ax.set_title(f'Topic {topic_idx +1}',
                     fontdict={'fontsize': 30})
        ax.invert_yaxis()
        ax.tick_params(axis='both', which='major', labelsize=20)
        for i in 'top right left'.split():
            ax.spines[i].set_visible(False)
        fig.suptitle(title, fontsize=40)

    plt.show()
```

Пример ключевых слов и топиков

Темы 1, 2, 3, 5, 6 - невозможно объединить в одну тему



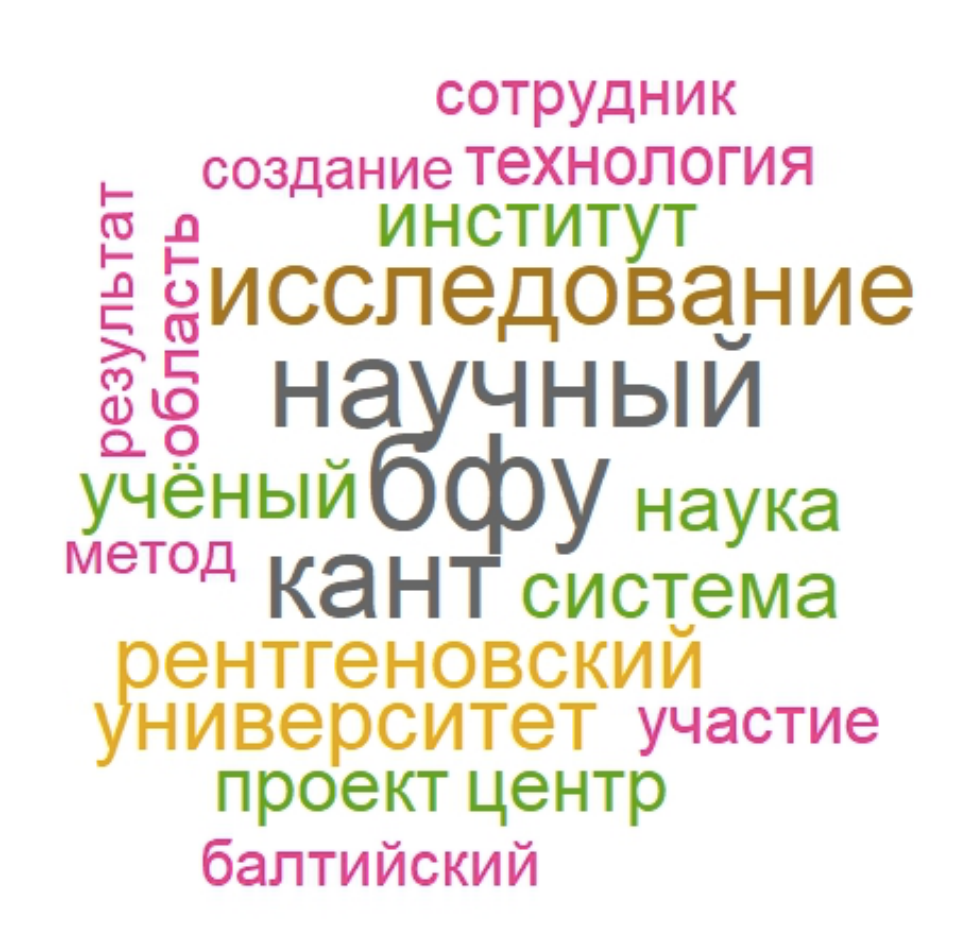
Но! Темы 4, 7



Индивидуализация ВУЗов

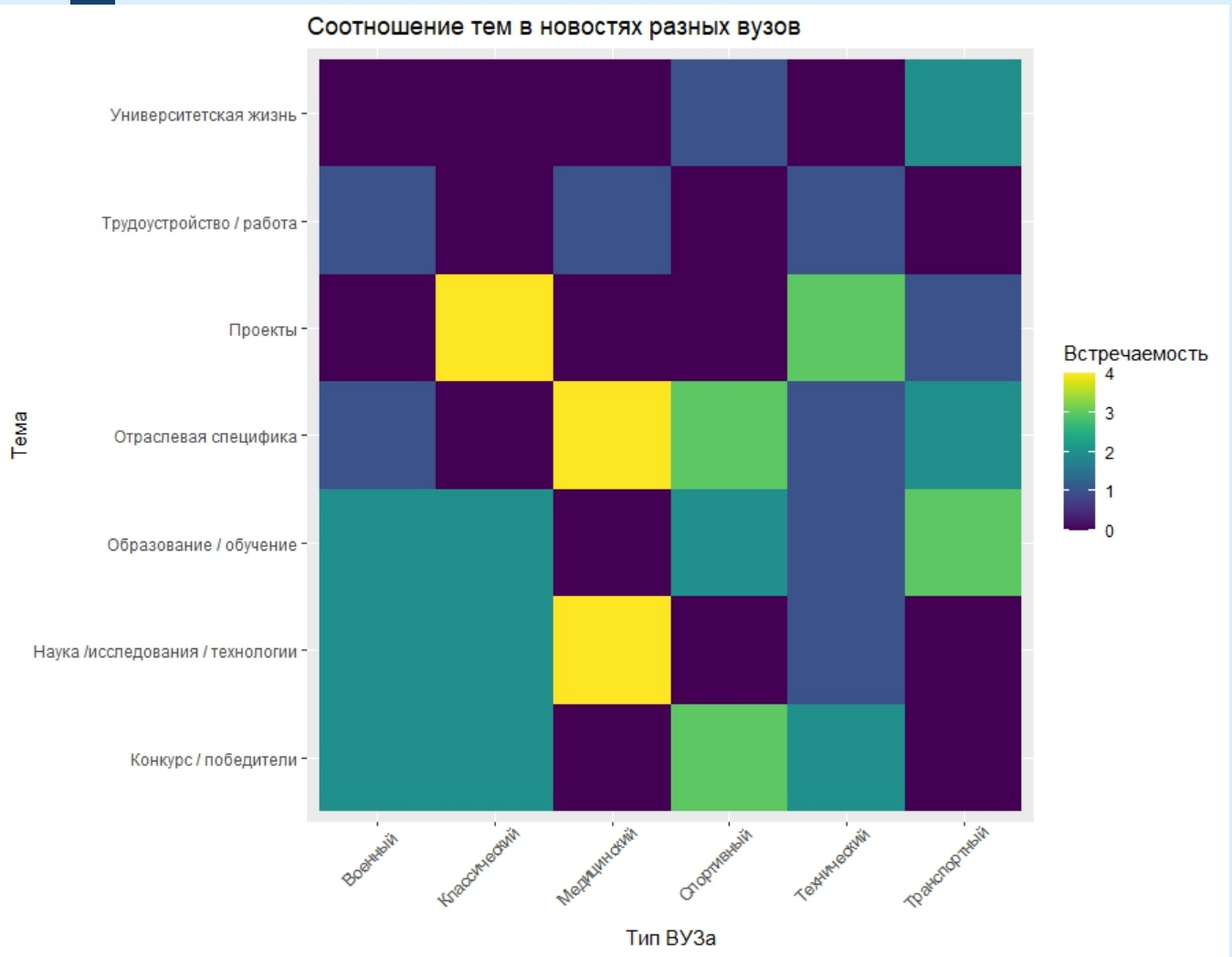
Визуализация в R (1)

Пример облака ключевых слов
(Отдельные ВУЗы)



Пример облака ключевых слов
(Все классические ВУЗы)

Визуализация в R (2)



Выводы, трудности, перспективы

Для качественного анализа текстов с целью определения идентичности ВУЗа необходимо работать конкретно с каждым ВУЗом, так как определяющими факторами являются - территориальный компонент, федеральный/нефедеральный, классический/отраслевой. Все эти факторы формируют идентичность. Индивидуализированный подход, не коллективный.

- Анализ контента из vk - отрицательный результат
- В отличие от Rake, Yake LDA дает более интерпретируемые результаты
- TfIdf для разных текстов одного ВУЗА- самый хороший результат
- 5 слов для анализа мало, 10 достаточно, но сложно выявить темы, 15-20 слов дает хороший результат
- Ngram = 1, 2 или Ngram = 4 или Ngram = 4, причем рассмотрение отдельно (LDA неинтерпретируема)
- Наиболее удачный вариант для LDA - 7 кластеров по 15 ключевых слов (n-gram = 1)
- Для TfIdf - n-gram = 1, 2
- После удаления ключевых слов в R результат лучше интерпретируется

Продолжение исследования - работа с конкордансом на корпусе с синтаксической разметкой для установления связей с именными сущностями (названия ВУЗов, университет, политех, кантиана…)

Схожие проекты

- Смирнова В.Д. Автоматическое определение тем, ассоциированных с пандемией covid-19, в русскоязычном корпусе социальных медиа (ВКР)
- Седова А.Г. Тематическое моделирование русскоязычных текстов с опорой на леммы и лексические конструкции, 2017 (ВКР)
- Sherstinova T. et al. Topic modeling of the Russian short stories of 1900–1930s: the most frequent topics and their dynamics
- Чижик А.В. Исследование динамики общественного настроения в социальных сетях с использованием методов тематического моделирования International Journal of Open Information Technologies ISSN: 2307-8162 vol. 9, no. 12, 2021
- Апишев М. Анализ текстов. Предобработка и выделение признаков (лекция).
- Чечнева Н.С. Исследование оценочной лексики потребительских отзывов в системе Яндекс.Маркет
- Митрофанова О.А., Гаврилик Д. Извлечение ключевых слов в научных текстах

СПАСИБО ЗА ВНИМАНИЕ