Сравнение лексики различных СМИ по нескольким акторам

Дербенёва Лиза, Фирсова Настя, Гришанова Аня

Куратор: Антон Леонов

Репозиторий проекта на GitHub

Предварительный план

1. Составление корпуса
2. Написание кода для обработки текста
3. Написание кода для анализа обращений и биграмм
4. Создание словаря частотности, визуализация для каждой статьи
5. Написание кода для определения окраски статьи
6. Написание кода для сравнения статей по тематике
7. Доработка программы до user-friendly состояния

Цель и гипотезы

Цель: автоматизировать сравнение содержания различных статей СМИ

Гипотезы:

1. Статьи будут различаться по типу окраски в зависимости от темы, на которую написана статья.
2. В похожих по стилю журналах об акторах будут говорить на одни и те же темы, а именно:
    a. Daily Mail и The Evening Standard будут наиболее похожими
    b. Hello! Magazine и Sky News будут наиболее сильно отличаться
3. Похожие по результатам автоматического анализа лексики статьи будут иметь одинаковую окраску и схожие тематики, а непохожие - разные.

Daily Mail

Factual Reporting
Very High
High
Mostly Factual
Mixed
LOW
Very Low

QUESTIONABLE SOURCE

London Evening Standard

Factual Reporting
Very High
High
MOSTLY FACTUAL
Mixed
Low
Very Low

RIGHT-CENTER BIAS

Данные с сайта https://mediabiasfactcheck.com

В журналах отражаются схожие политические взгляды, степень достоверности фактов примерно одинакова.

Sky News UK

Данные с сайта
https://mediabiasfactcheck.com

Factual Reporting
Very High
HIGH
Mostly Factual
Mixed
Low
Very Low

LEAST BIASED

SkyNews и Hello! Magazine сильно различаются:

- SkyNews является достаточно достоверным источником со статьями разных тематик
- Hello! Magazine - журнал сплетен о знаменитостях

1. Составление корпуса

Материалы

Были выбраны 4 журнала, соответствующих критериям:

- наличие поиска по статьям на сайте
- популярность издания
- констраст между изданиями

www.dailymail.co.uk

www.hellomagazine.com

www.standard.co.uk

www.news.sky.com

Пример кода для скачивания с www.hellomagazine.com

```
import requests
from bs4 import BeautifulSoup as bs
import re
import time
import wget
```

```
file_out = open('out.txt', 'w', encoding='utf-8')
root_url = 'https://www.hellomagazine.com/tags/kanye-west/'
root = 'https://www.hellomagazine.com/'
resp = requests.get(root_url)
page = bs(resp.content, 'lxml')
time.sleep(3)
page_results = []
for div in page.find_all('div'):
    for link in div.find_all('a'):
        if 'kanye' in str(link).lower():
            link_result = re.search(r'(?<=href=").*?(?=")', str(link)).group()
            #print(link_result,file=file_out)
            if link_result not in page_results:
                if 'tags' not in link_result:
                    page_results.append(link_result)
#for page in page_results:
    #print(page,file=file_out)
path = 'C:\\Users\\Ann\\PycharmProjects\\py@21project\\' + re.findall(r'[A-z\.]+'
#print(page_results[0])
#wget.download(page_results[0], path + f'\\{0}.html')
for i in range(len(page_results)):
    wget.download(page_results[i], path + f'\\{i}.html')
```

2. Написание кода для обработки текста

Код для обработки

Весь код с комментариями есть в репозитории на github.

Сначала был написан код для обработки одной статьи. В результате его работы пользователь получает список из 50 наиболее частотных лемм и облако из них же.

Далее код был усовершенствован для обработки корпуса.

Создание корпуса

Использовались библиотеки **BeautifulSoup** и **NLTK**.

```
# удаление html-тэгов

    file = open(filename, "r", encoding="utf-8")
    contents = file.read()
    file.close()
    soup = BeautifulSoup(contents, 'lxml')

    html_free = soup.get_text('\n', strip='True')
    text = html_free.lower()
```

Создание корпуса

Помимо удаления лишних символов, была проведена токенизация текста и были удалены стоп-слова.

```
# удаление стоп-слов


from nltk.corpus import stopwords
#nltk.download('stopwords')
english_stopwords = stopwords.words("english")
no_stopwords = ''
for w in text_tokens:
    if w not in english_stopwords:
        no_stopwords = no_stopwords + w + ' '
```

Создание корпуса

После текст лемматизировался с помощью библиотеки **SpaCy** и создавался список наиболее частотных лемм.

Конченый корпус представлял из себя словарь в файле формата json с лемматизированными текстами статей, в которых имя актора упоминается 3 и более раз.

```
import spacy

nlp = spacy.load('en_core_web_sm')
doc = nlp(no_stopwords)
lemmatization = []
for token in doc:
    lemmatization.append(token.lemma_)

text_lemma = nltk.Text(lemmatization)

from nltk.probability import FreqDist
fdist = FreqDist(text_lemma)
```

Облака частотных слов

Облака создавались с помощью библиотек **wordcloud** и **matplotlib.**

```
# чтобы исключить имя актора из частотных слов

stops=str(input('Напишите фамилию и формы имени актора, которые могут часто встречаться, с маленькой буквы через пробел: '))
stops_list=stops.split()

   freq_dict = dict()
   for t in tg_dict:
       freq = prepared(tg_dict[t])
       freq_dict.update({ t : freq[0] })
       text_raw = " ".join(freq[1])
       # указываем размеры и цвета изображения, а также список стоп-слов
       wordcloud = WordCloud(width=1000, height=1000, stopwords = stops_list,
                             background_color = "#fff5ee", colormap = "tab10").generate(text_raw)
       cloudname = t + '.png'
       wordcloud.to_file(cloudname)
```

3. Написание кода для анализа биграмм

Код для вывода биграмм

```
words = []
for key in data.keys():
    text = data[key].split('.')
    for sentence in text:
        bi_grams = list(ngrams(sentence.split(), 2))
        for gram in bi_grams:
            word = gram[0] + '_' + gram[1]
            words.append(word)
    for word in words:
        dict_magazines[key] += word + ' '
```

Биграммы были преобразованы в вид:

(‘two’, ‘words’) -> ‘two_words’

для корректного построения и отображения облака слов.

Далее облако строилось с помощью библиотеки **wordcloud**

Код для вывода биграмм

В far-right газете Daily Mail самой частотной биграммой оказалась: trans_activists

В остальных газетах самой частотной была биграмма harry_potter.

4. Написание кода для определения окраски статьи

Код для определения эмоциональной окраски

Dataset с веб-сайта IMDB (by Andrew Maas)

- Значительно больше данных, чем в других эталонных наборах данных - 50000 отзывов
- 25000 крайне полярных обзоров фильмов для обучения и 25 000 для тестирования
- Равное количество позитивных и негативных комментариев
- Большинство отзывов состоит из 200-350 слов

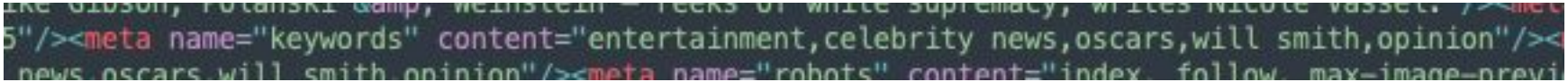
Код для определения эмоциональной окраски

Очистка, обработка и лемматизация набора данных, на основе которого будет происходить определение эмоциональной окраски статьи ➔ Создание списков из слов, которые используются в положительных и в отрицательных отзывах ➔ Определение количества совпадений между словами, использованными в статье/газете/корпусе, и “положительными”/“отрицательными” словами

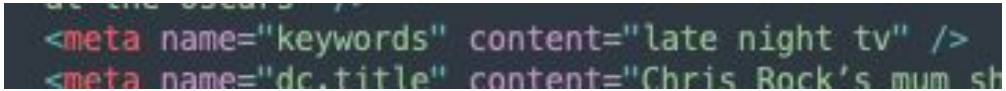
5. Написание кода для определения тематики статьи

Определение тематики статьи

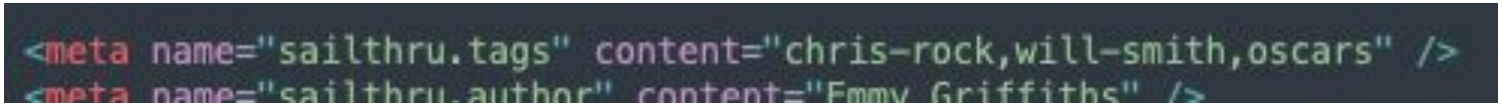
На сайте *www.glamourmagazine.co.uk* tags были обозначены как *keywords*:



На сайте *www.hellomagazine.com* под *keywords* подразумевалось другое:



А tags были обозначены как *sailthru.tags*:



→ В разных журналах использовались разные tags

Определение тематики статьи

С помощью облаков частотных слов

+ можно точно определить, о какой конкретно теме, затрагивающей актора, идёт речь (но не всегда)
- трудно делать обобщения и статистику
- для каждого текста отдельно - трудоемко и времязатратно

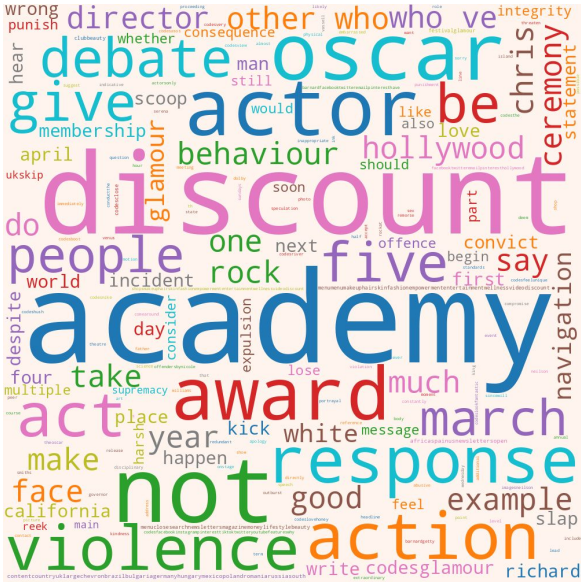
С помощью сайта https://ru.megaindex.com/a/tcategories

+ определение общих тем (например, Society/People), на данных можно делать какие-либо сравнения и статистические подсчеты
- нельзя узнать конкретную тему и точно определить, как она связана с актором
- для каждого текста отдельно - трудоемко и времязатратно

Определение тематики статьи

Статья про трансофобный твит Дж. Роулинг

Статья про новый альбом Канье

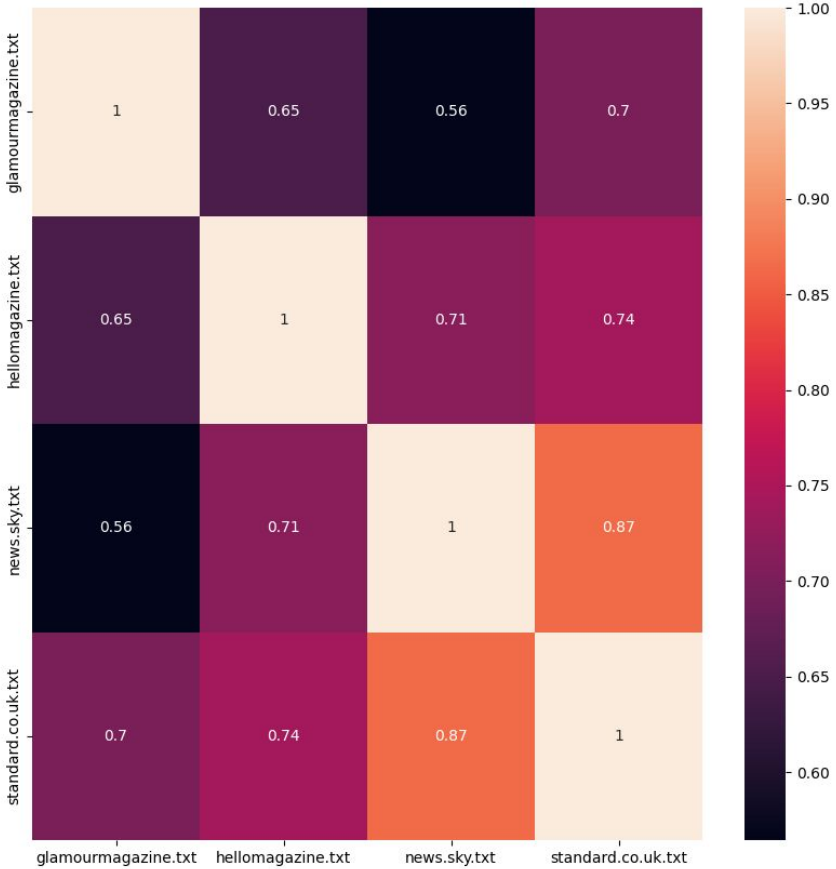


Скандал при участии Уилла Смита на премии Оскар

6. BERT. Сравнение лексики статей и журналов.

Сравнение текстов.

Для определения сходства между содержанием статей и журналов мы использовали предварительно обученную модель SBERT. Программа получала на вход файл с корпусом в формате json, а результатом её работы была таблица сходства статей/журналов.



6. Доработка программы до user-friendly состояния

Для дальнейшего использования другими

По ряду причин сделать программу полностью user-friendly не удалось, поэтому помимо наличия диалога с пользователем в некоторых программах (например, в предварительной обработке корпуса) были добавлены:

1. Подробные комментарии к коду на github + пояснения в readme.md файлах.
2. Инструкция (папка final на github).

Анализ полученных результатов

Положительная и отрицательная окраска статей

Результат анализа корпусов:

- *jk-rowling.json*: скорее положительная окраска
- *kanye-west.json*: скорее положительная окраска
- *will-smith.json*: скорее положительная окраска

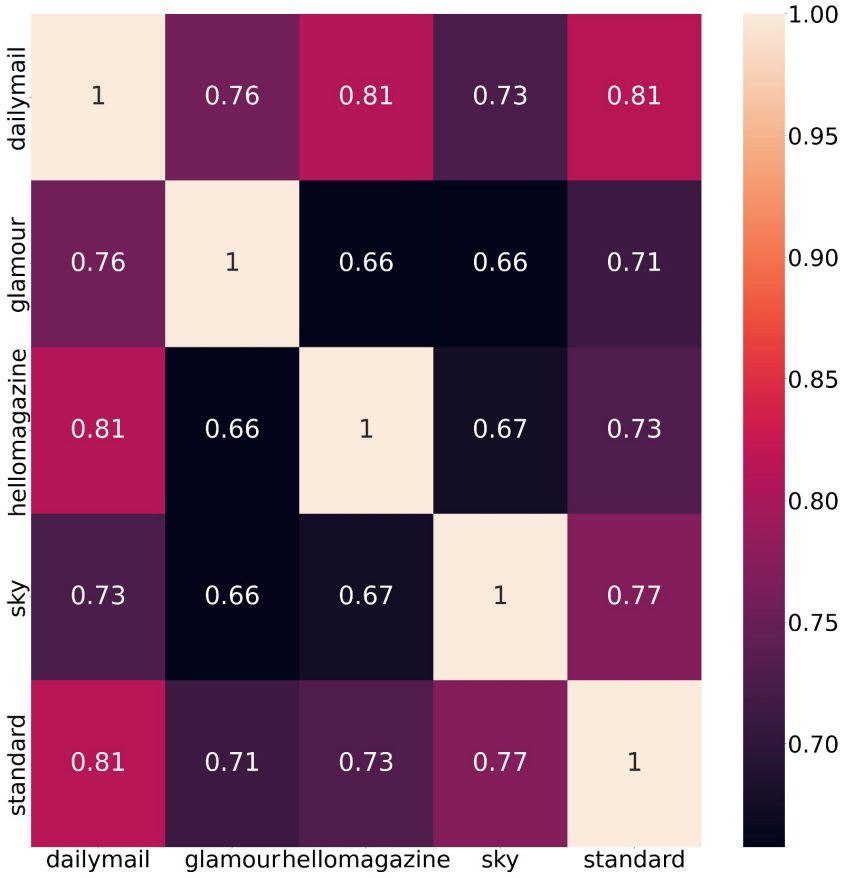
Некоторые наблюдения:

- Почти все статьи имеют положительную окраску, предположительно, потому что издательства из-за нормативно-правового регулирования цензуры.
- Те статьи, которые имеют отрицательную окраску, часто содержат описывающие жестокость слова (например, статья 8 из корпуса *will-smith-cut.json* считается негативной, так как содержит лемму ‘slap’ целых 12 раз)
- Статьи будут различаться по типу окраски в зависимости от темы, на которую написана статья, иногда даже самым неожиданным образом. (Подтвердилась гипотеза №1)

Сравнение текстов.

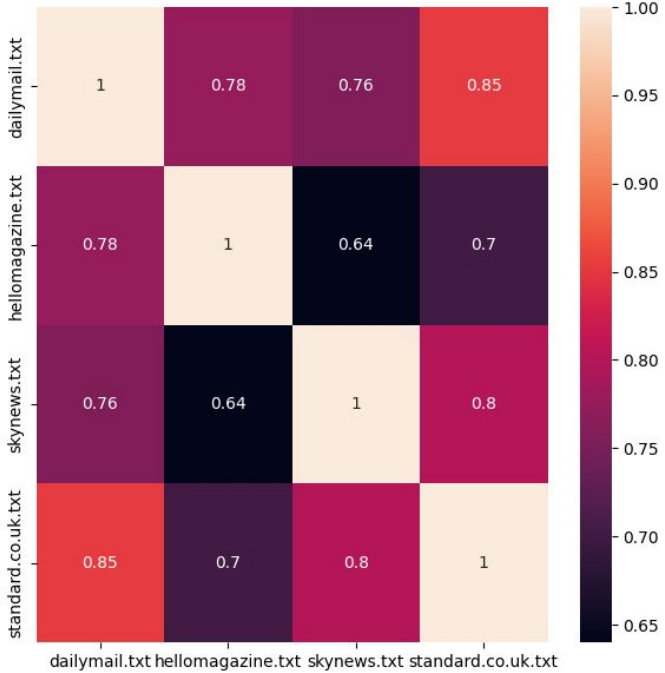
При сравнении объединённых по источнику статей (один журнал - статьи по трём акторам) наиболее похожими оказались:

- Daily Mail и Hellomagazine
- Daily Mail и Standard

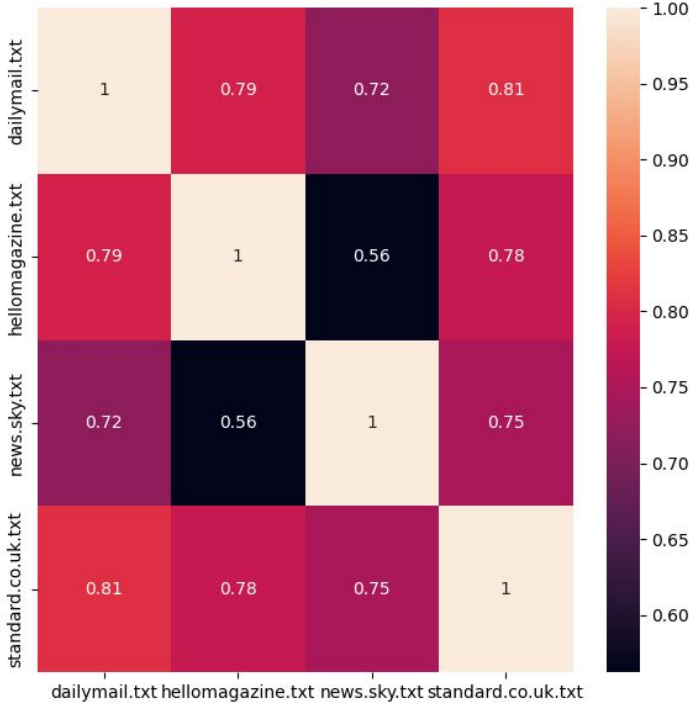


Сравнение текстов

Далее мы проанализировали корпуса журналов, содержащие статьи по каждому актору отдельно.



Роулинг



Канье

Сравнение текстов: подтверждение гипотезы 2

Очень похожими были результаты сравнения корпусов Канье и Дж. Роулинг. Первое и последнее место по сходству занимали одни и те же журналы (Daily Mail и Standard, skynews и hellomagazine соответственно). Кроме того, практически одинаковое процентное сходство имела и пара Daily Mail + HelloMagazine.

топ	Роулинг			Канье		
	журнал 1	журнал 2	похожесть	журнал 1	журнал 2	похожесть
1	standard	dailymail	0,85	standard	dailymail	0,81
…	hellomagazine	dailymail	0,78	hellomagazine	dailymail	0,79
6	skynews	hellomagazine	0,64	hellomagazine	news.sky	0,56

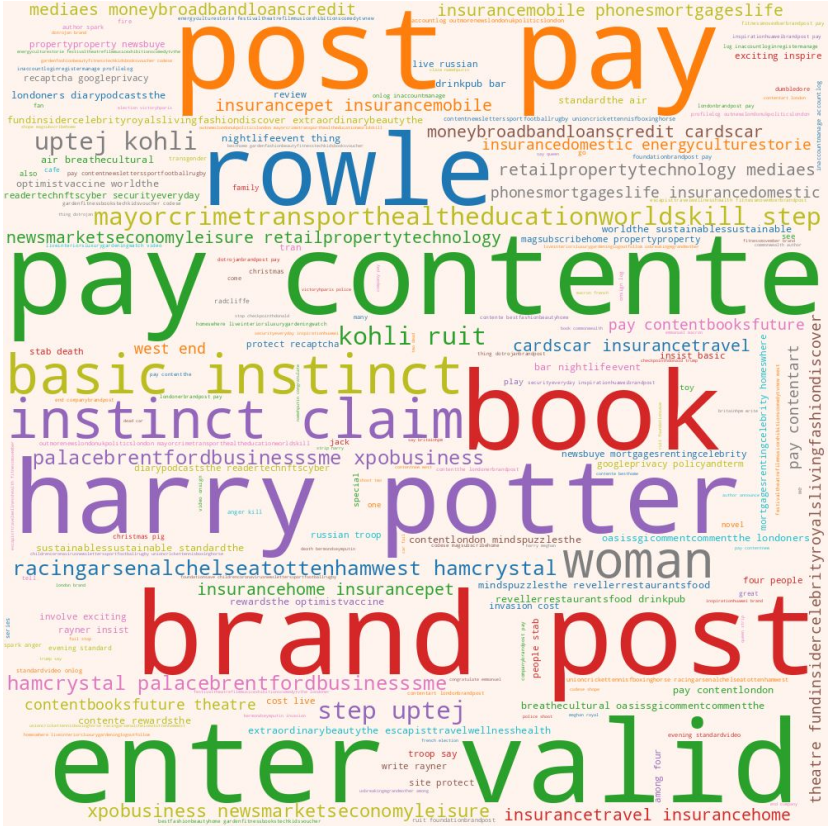
Сравнение текстов

Мы попытались определить темы, на которые писали похожие журналы, примитивным способом - с помощью облак наиболее частотных слов.

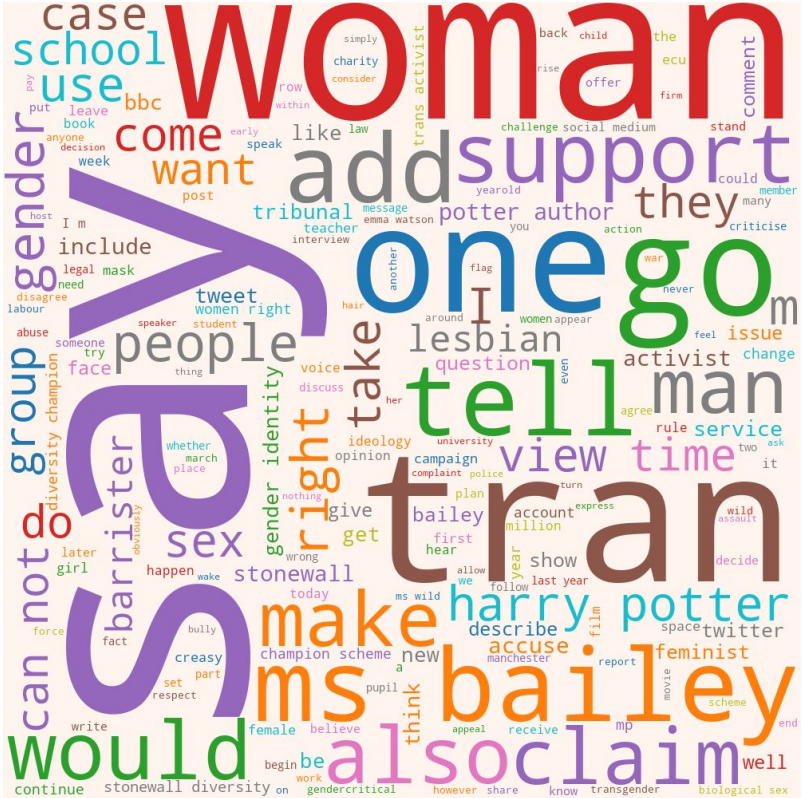
В корпусе Роулинг были наиболее примечательные результаты, поэтому далее представлены результаты по нему.

Можно заметить, что в обоих похожих журналах речь идёт про Гарри Поттера и транософобию примерно в равных соотношениях.

Standard



Daily Mail



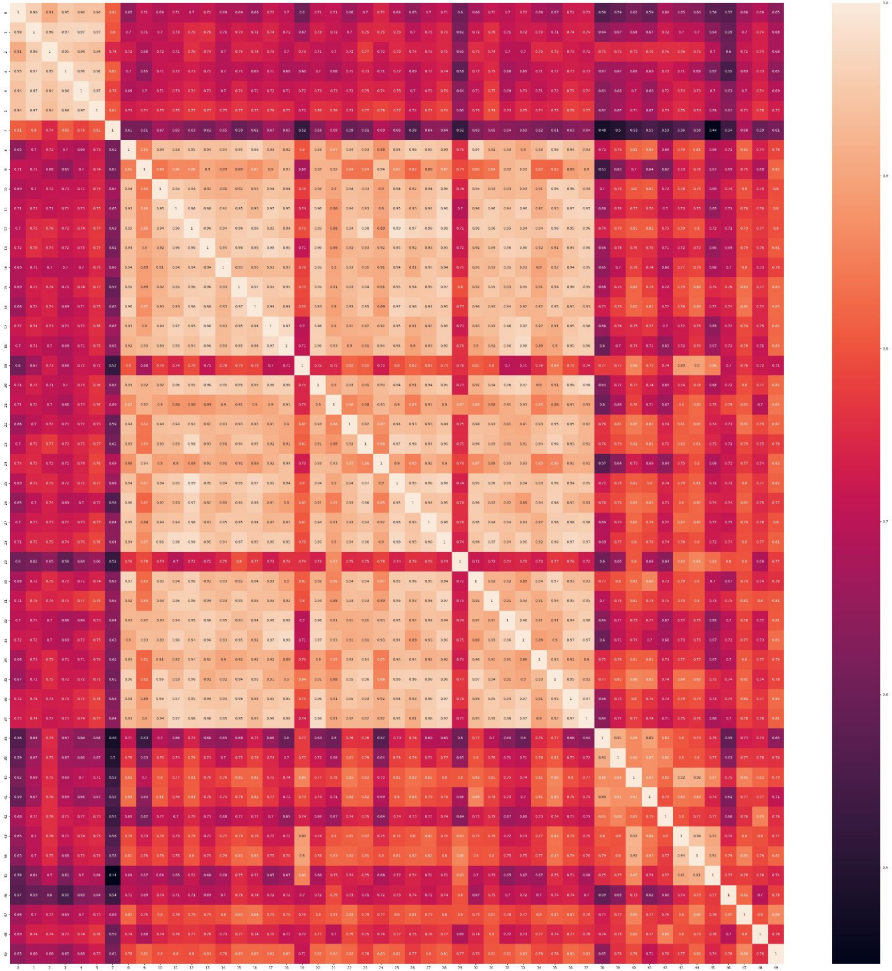
В наиболее непохожих журналах наблюдается различие тем: в первом речь идёт про скандал, связанный с трансофобным твитом, во втором, видимо, про “Гарри Поттер 20 лет спустя: возвращение в Хогвартс”

SkyNews

HelloMagazine

Сравнение статей

Было выполнено сравнение сокращенных корпусов (50 статей) для 3 акторов, и были выделены самые похожие/непохожие статьи



Сравнение самых похожих статей: подтверждение гипотезы 3

Похожие по результатам БЕРТа статьи имеют одинаковую окраску и тематику.

Самые похожие статьи			
номер статьи	Положительная / отрицательная	Тематика	Сходство
Дж Роулинг			
82	Положительная	Society/People (81,8%), Arts/Movies (77,1%)	
111	Положительная	Arts/Movies (89,9%), Society/People (86,3%)	0,99
Канье			
3	Положительная	Society/People	
29	Положительная	Society/People	0,98
Уилл Смит			
12	Положительная	Society/People	
25	Положительная	Society/People	0,99

Сравнение самых НЕпохожих статей: частичное подтверждение гипотезы 3

В непохожих статьях не наблюдается различие в окраске, но во всех случаях есть различия в тематике.

Самые НЕпохожие статьи			
номер статьи	Положительная / отрицательная	Тематика	Сходство
Дж Роулинг			
124	Положительная	Arts/Literature	
127	Положительная	Society/Politics	0,53
Канье			
18	Положительная	Socity/Politics	
41	Положительная	Home/Family	0,58
Уилл Смит			
7	Положительная	Health/Diseases	
45	Положительная	Society/People	0,44

Обязанности

Настя Фирсова

- создание корпуса (Уилл Смит)
- написание кода для определения окраски текстов
- определение тематики текстов
- анализ полученных данных

Аня Гришанова

- создание корпуса (Канье Уэст)
- написание кода и анализ биграмм
- исправление ошибок в лемматизации и исключение посторонних слов, не относящихся к содержанию статей, из текстов корпуса (папка final на github)
- написание инструкции ко всем кодам
- анализ полученных данных

Лиза Дербенева

- создание корпуса (Дж. Роулинг)
- написание кода для предварительной обработки статей, создания облаков частотных слов
- BERT
- подготовка материалов для анализа (папка “результаты анализа трёх корпусов” на github)
- анализ полученных данных