

```
In [1]: import pandas as pd
import numpy as np

from IPython.display import display, Markdown

from urllib.request import urlopen
import requests

from collections import Counter
from itertools import chain

import matplotlib.pyplot as plt

import re
import nltk
from bs4 import BeautifulSoup
from pymorphy2 import MorphAnalyzer
from wordcloud import WordCloud

import json

from pathlib import Path
import os

from ngutils import *
from tqdm import tqdm

DATA_DIR = Path('data')
os.listdir(DATA_DIR)
```

```
Out[1]: ['areas + districts.json',
 'areas.csv',
 'datasets.json',
 'dataset_news_1.xlsx',
 'dataset_news_1_mod.xlsx',
 'dst-off-lct10.py',
 'news.json',
 'normal_forms.csv',
 'organizations.csv',
 'result_task3.csv',
 'Материалы для презентации ЛЦТ21',
 'Материалы для презентации ЛЦТ21-20211018T195925Z-001.zip',
```

'ТЗ\_Задача 10. Рекомендательная система новостей для пользователей mos.ru и приложения Моя Москва.pdf',  
'Шаблон презентации конкурса Лидеры Цифровой трансформации 2021.pptx']

Финальный этап (4-7 ноября):

Члены комиссии по присуждению премий Мэра Москвы «Лидеры цифровой трансформации» голосуют по шкале от 0 до 5 баллов с шагом 1 балл согласно следующим критериям:

- Подход коллектива к решению задачи (идея решения задачи, оригинальность, способ реализации);
- Техническая проработка решения (обоснованность выбранных методов для разработки модели, алгоритма, сервиса);
- Эффективность решения в рамках поставленной задачи (точность прогнозирования/распознавания/рекомендаций, количество учтенных факторов и др. в зависимости от выбранной задачи);
- Выступление коллектива на питч-сессии.

In [2]:

```
# Загрузка организаций. Источник - mos.ru/api, выявлен путем изучения сайта mos.ru
if not (DATA_DIR/'organizations.csv').exists():
    list_organizations = []
    href = 'https://www.mos.ru/api/structure/v1/frontend/json/ru/institutions?per-page=50'
    while True:
        json_page = json.load(urlopen(href))
        list_organizations += json_page['items']
        print(f"Загружено {len(list_organizations)} организаций" + " "*10, end='\r')
        if href != json_page['_links'][['last']]['href']:
            href = json_page['_links'][['next']]['href']
        else:
            break
    # Список организаций/департаменов сохраняем в таблицу df_organizations
    df_organizations = (
        pd.DataFrame(list_organizations)
        .drop(columns=['yandex_metrika_id','edo_id','rich_snippet','external_card_title','reception_description','icon'])
        .set_index('id')
    )
    df_organizations.lead_institution_id = df_organizations.lead_institution_id.fillna(0).astype(int)
    df_organizations.to_csv(DATA_DIR/'organizations.csv')
else:
    df_organizations = pd.read_csv(DATA_DIR/'organizations.csv', index_col='id')

view_types(df_organizations, display_force=True)
df_organizations.head(3)
```

	<b>int</b>	<b>str</b>	<b>NaN</b>		<b>(min)</b>		<b>(max)</b>	<b>(unique)</b>
<b>type_id</b>	2905	0	0		1691		14691	13
<b>name</b>	0	2905	0	Акционерное общество "Концерн "Радио-центр"	Юридическое управление префектуры Северо-Восто...		2842	
<b>description</b>	0	57	2848	Антитеррористическая комиссия города Москвы ко...	Управление делами Мэра и Правительства Москвы ...		58	
<b>pgu_link</b>	0	30	2875	/pgu/ru/departments/7700000000163323287/	https://www.mos.ru/pgu/ru/departments/77000000...		31	
<b>has_reception</b>	2905	0	0		0		1	2
<b>code</b>	0	51	2854		apr		zags	52
<b>short_name</b>	0	238	2667		AHO "ЦРСВ"		Ясенево	239
<b>lead_institution_id</b>	2905	0	0		0		103564090	127
<b>external_card_url</b>	0	160	2745	http://adm-moskovsky.ru/		https://zyablikovo.mos.ru/		161

2905 rows x 9 columns

Out[2]:		<b>type_id</b>	<b>name</b>	<b>description</b>	<b>pgu_link</b>	<b>has_reception</b>	<b>code</b>	<b>short_name</b>
		<b>id</b>						
		9224090	6691	Комитет города Москвы по обеспечению реализаци...	Москомстройинвест контролирует соблюдение зако...	https://www.mos.ru/pgu/ru/departments/77000000...	1	invest
		9238090	1691	Департамент предпринимательства и инновационно...	Основными направлениями работы Департамента яв...	NaN	1	dpir
		9463090	1691	Департамент внешнеэкономических и международны...	Департамент разрабатывает и реализует политику...	NaN	1	dvms

In [3]:

```
# Загрузка округов и районов. Источник - mos.ru/api, ссылка извлечена из файла 'areas + districts.json',
# предоставленного организаторами
if not (DATA_DIR/'areas.csv').exists():
```

```

href = 'https://www.mos.ru/api/directories/v2/frontend/json/territory/districts?expand=areas&per-page=50&page=1'
json_page = json.load(urlopen(href))
s_areas_districts = pd.DataFrame(json_page['items']).set_index(['id', 'title']).areas
df_areas = (
    s_areas_districts[s_areas_districts.str.len() > 0]
    .explode()
    .transform({'area_id': lambda x: x['id'], 'area_title': lambda x: x['title']})
    .reset_index()
    .set_index('area_id')
    .rename(columns={'id': 'district_id', 'title': 'district_title'})
)
df_areas.to_csv(DATA_DIR/'areas.csv')
else:
    df_areas = pd.read_csv(DATA_DIR/'areas.csv', index_col='area_id')

districts = dict(df_areas[['district_id', 'district_title']].values)
view_types(df_areas, display_force=True)
print(f"\nОкруга: \n{districts}")
df_areas.sample(3)

```

	int	str	(min)	(max)	(unique)
<b>district_id</b>	146	0	1500	11500	11
<b>district_title</b>	0	146	Восточный	Южный	11
<b>area_title</b>	0	146	Академический	Ясенево	146

146 rows x 3 columns

Округа:

{1500: 'Центральный', 2500: 'Южный', 3500: 'Северный', 4500: 'Юго-Западный', 5500: 'Северо-Восточный', 6500: 'Западный', 7500: 'Восточный', 8500: 'Северо-Западный', 9500: 'Юго-Восточный', 10500: 'Зеленоградский', 11500: 'Троицкий и Новомосковский'}

Out[3]:

	district_id	district_title	area_title
	area_id		
<b>34501</b>	3500	Северный	Западное Дегунино
<b>87501</b>	7500	Восточный	Восточный
<b>79501</b>	6500	Западный	Проспект Вернадского

In [4]:

```
print(f"Округ id = 1500: {districts[1500]}")
```

```
print(f"Район id = 16501: {df_areas.loc[16501].area_title}; "
      f"округ id = {df_areas.loc[16501].district_id}: {df_areas.loc[16501].district_title}")
```

Округ id = 1500: Центральный  
Район id = 16501: Зябликово; округ id = 2500: Южный

In [5]:

```
# Загрузка новостей
df_json = pd.read_json(DATA_DIR/'news.json')
```

In [6]:

```
# Переформатирование таблицы новостей
df_json['tag_ids'] = df_json.tags.apply(lambda x:[item['id'] for item in x])

# Тэги
df_tags = pd.DataFrame(chain.from_iterable(df_json.tags)).drop_duplicates().set_index('id').rename_axis(None)
df_tags['qty'] = pd.Series(Counter(chain.from_iterable(df_json.tag_ids)))
print('Топ-10 тэгов')
df_tags = df_tags.sort_values('qty', ascending=False)
df_tags.head(10)
```

Топ-10 тэгов

Out[6]:

		title	created_at	qty
4019217		Сергей Собянин	2016-01-26 13:27:14	744
36217		строительство	2015-12-28 00:16:28	414
47247217		коронавирус	2020-03-03 12:40:22	343
62217		парки	2015-12-28 00:16:28	310
25217		благоустройство	2015-12-28 00:16:28	294
5433217		Владимир Ефимов	2016-04-18 12:11:11	266
3217		транспорт	2015-12-28 00:16:28	258
47576217		COVID-19	2020-03-16 11:46:31	221
16312217	программа реновации		2017-03-23 16:26:46	200
19925217	Алексей Фурсин		2017-07-19 17:50:16	188

In [7]:

```
# Темы
```

```

df_themes = pd.DataFrame(chain.from_iterable(df_json.themes)).drop_duplicates().set_index('id').rename_axis(None)
df_themes['qty'] = pd.Series(Counter(chain.from_iterable(df_json.theme_ids)))
print('Топ-10 тем')
df_themes = df_themes.sort_values('qty', ascending=False)
df_themes.head(10)

```

Топ-10 тем

Out[7]:

		title	created_at	updated_at	icon_id	url	qty
157287	Строительство и благоустройство	2019-11-06 11:08:33	2021-07-02 17:43:19	4061.0	/news/maintheme/157287/	348	
115287	Интересная Москва	2019-01-15 12:02:21	2021-07-06 20:04:25	8061.0	/news/maintheme/115287/	237	
27287	Развитие метро	2017-11-03 22:23:24	2021-07-02 17:47:10	3061.0	/news/maintheme/27287/	99	
2287	Планируйте маршрут	2017-11-03 22:23:24	2021-07-07 10:17:22	NaN	/news/maintheme/2287/	80	
99287	Музейные истории	2018-06-07 15:19:55	2021-07-06 20:04:25	5061.0	/news/maintheme/99287/	77	
213287	Здоровье и технологии	2021-05-19 12:06:26	2021-07-02 17:49:57	19061.0	/news/maintheme/213287/	63	
117287	Семейные выходные	2019-04-08 13:04:35	2021-07-02 16:09:06	11061.0	/news/maintheme/117287/	48	
116287	Советы библиотекаря	2019-03-18 12:23:01	2021-07-02 16:09:06	NaN	/news/maintheme/116287/	43	
60287	Люди города	2017-11-03 22:23:24	2021-04-08 20:36:38	NaN	/news/maintheme/60287/	39	
210287	Капремонт	2021-03-12 14:09:52	2021-06-18 18:29:01	NaN	/news/maintheme/210287/	37	

In [8]:

```

# Сфераы
df_json['sphere_id'] = df_json.sphere.apply(lambda x:x['id'])
df_json['sphere_ids'] = df_json.spheres.apply(lambda x:[item['id'] for item in x])
df_spheres = pd.DataFrame(chain.from_iterable(df_json.spheres)).drop_duplicates().set_index('id').rename_axis(None)
df_spheres['qty'] = pd.Series(Counter(chain.from_iterable(df_json.sphere_ids)))
print('Топ-10 сфер')
df_spheres = df_spheres.sort_values('qty', ascending=False)
df_spheres.head(10)

```

Топ-10 сфер

Out[8]:

		title	special	activated	priority	qty
231299		Мой район	0	1	410	1416
4299		Строительство и реконструкция	0	1	490	1095

			title	special	activated	priority	qty
<b>3299</b>			Культура	0	1	360	986
<b>12299</b>	Экономика и предпринимательство			0	1	370	847
<b>1299</b>			Социальная сфера	0	1	400	717
<b>5299</b>			Городское хозяйство	0	1	460	651
<b>18299</b>			Здравоохранение	0	1	500	606
<b>183299</b>			Технологии	0	1	440	602
<b>2299</b>			Транспорт	0	1	480	593
<b>15299</b>			Образование	0	1	470	455

In [9]:

```
# Организации
df_json['void'] = ''
df_json void = df_json void.str.split()
df_json organizations mask(df_json organizations.isna(), df_json void, inplace=True)
df_json['organization_ids'] = df_json organizations
df_organizations['qty'] = pd.Series(Counter(chain.from_iterable(df_json organization_ids)))
print('Топ-10 организаций')
df_organizations = df_organizations.sort_values('qty', ascending=False)[
    ['name', 'has_reception', 'code', 'short_name', 'lead_institution_id', 'qty']
]
df_organizations.head(10)
```

Топ-10 организаций

Out[9]:

			name	has_reception	code	short_name	lead_institution_id	qty
			id					
<b>20614090</b>	Департамент информационных технологий города М...			1	dit	ДИТ	11491090	319.0
<b>9238090</b>	Департамент предпринимательства и инновационно...			1	dpir	ДПИР	11491090	309.0
<b>19180090</b>	Департамент градостроительной политики города ...			1	dgp	ДГП	11491090	190.0
<b>19889090</b>	Департамент жилищно-коммунального хозяйства го...			1	dgkh	ДЖХ	11491090	187.0
<b>12585090</b>	Департамент культуры города Москвы			1	kultura	Депкульт	11491090	186.0

<b>id</b>			<b>name</b>	<b>has_reception</b>	<b>code</b>	<b>short_name</b>	<b>lead_institution_id</b>	<b>qty</b>
<b>20488090</b>	Департамент культурного наследия города Москвы			1	dkn	ДКН	11491090	159.0
<b>103367090</b>	Департамент инвестиционной и промышленной политики			1	dipp	ДИПП	11491090	157.0
<b>20882090</b>	Департамент транспорта и развития дорожно-транс...			1	dt	Дептранс	11491090	153.0
<b>20703090</b>	Департамент природопользования и охраны окружающей среды			1	eco	ДПиООС	11491090	146.0
<b>9479090</b>	Департамент здравоохранения города Москвы			1	dzdrav	Депздрав	11491090	143.0

In [10]:

```
# Заполняем пропуски превью и текста, подставляя замещающие столбцы
df_json.preview_text.mask(df_json.preview_text.isna(), df_json.preview, inplace=True)
df_json.full_text.mask(df_json.full_text.isna(), df_json.text, inplace=True)

df_json.theme_id = df_json.theme_id.fillna(0).astype(int)
df_json.territory_area_id = df_json.territory_area_id.fillna(0).astype(int)
df_json.territory_district_id = df_json.territory_district_id.fillna(0).astype(int)
df_json.oiv_id = df_json.oiv_id.fillna(0).astype(int)
df_json.label = df_json.label.fillna('')
```

In [11]:

```
# Оставляем только нужные колонки
cols = ['id', 'title', 'url', 'published_at',
         'label', 'tag_ids',
         'theme_id', 'theme_ids',
         'sphere_id', 'sphere_ids',
         'territory_area_id', 'territory_district_id',
         'preview_text', 'full_text',
         'oiv_id', 'organization_ids']
```

In [12]:

```
df_news = df_json[cols].set_index('id')
view_types(df_news)
```

	<b>str</b>	<b>Timestamp</b>	<b>list</b>	<b>int</b>	<b>(min)</b>	<b>(max)</b>	<b>(unique)</b>
<b>title</b>	6554	0	0	0	#Доброосень: на фестивале «Золотая осень» стар...	Советская классика на больших экранах: «Моски...	6485

	str	Timestamp	list	int	(min)	(max)	(unique)
<b>url</b>	6554	0	0	0	/mayor/themes/10299/3239050/	/news/item/9921073/	6554
<b>published_at</b>	0	6554	0	0	2011-08-29 19:42:00	2021-08-31 18:47:56	6453
<b>label</b>	6554	0	0	0		Юбилей фильма	57
<b>tag_ids</b>	0	0	6554	0	[10004217, 13858217, 33662217, 44745217]	[]	6099
<b>theme_id</b>	0	0	0	6554	0	213287	41
<b>theme_ids</b>	0	0	6554	0	[100287, 115287, 116287, 117287]	[]	77
<b>sphere_id</b>	0	0	0	6554	1299	352299	62
<b>sphere_ids</b>	0	0	6554	0	1299	352299	1408
<b>territory_area_id</b>	0	0	0	6554	0	146501	146
<b>territory_district_id</b>	0	0	0	6554	0	12500	13
<b>preview_text</b>	6554	0	0	0	Ярмарки нового формата становятся площадками д...	6425	
<b>full_text</b>	6554	0	0	0	<blockquote>&laquo;ВДНХ меняется на глазах. Еж...	Мэр Москвы принял участие в торжественной...	6554
<b>oiv_id</b>	0	0	0	6554	0	103466090	45
<b>organization_ids</b>	0	0	6554	0	[100173090]	[]	161

6554 rows x 15 columns

In [13]:

```
morph = MorphAnalyzer()
stop = nltk.corpus.stopwords.words('russian')
stop.extend(['как', 'что', 'где', 'какой', 'для', 'более', 'чем', 'ещё', 'быть', 'стать', 'мой', 'это', 'можно', 'всё',
            'почти', 'парка', 'после', 'первый', 'четыре', 'пять', 'шесть', 'семь', 'тысяча', 'самый', 'конец', 'год',
            'который', 'также', 'весь', 'свой', 'сергей', 'число', 'около', 'имя', 'кроме', 'каждый', 'корп', 'москва',
            ])
str(stop)
# стоп-слова с добавлениями
```

Out[13]: "[и", "в", "во", "не", "что", "он", "на", "я", "с", "со", "как", "а", "то", "все", "она", "так", "его", "но", "да", "ты", "к",
 "у", "же", "вы", "за", "бы", "по", "только", "ее", "мне", "было", "вот", "от", "меня", "еще", "нет", "о", "из", "ему", "теперь",
 "когда", "даже", "ну", "вдруг", "ли", "если", "уже", "или", "ни", "быть", "был", "него", "до", "vas", "нибудь", "опять", "уж", "ва
 м", "ведь", "там", "потом", "себя", "ничего", "ей", "может", "они", "тут", "где", "есть", "надо", "ней", "для", "мы", "тебя", "и

```
'х', 'чем', 'была', 'сам', 'чтоб', 'без', 'будто', 'чего', 'раз', 'тоже', 'себе', 'под', 'будет', 'ж', 'тогда', 'кто', 'этот', 'того', 'потому', 'этого', 'какой', 'совсем', 'ним', 'здесь', 'этом', 'один', 'почти', 'мой', 'тем', 'чтобы', 'неё', 'сейчас', 'были', 'куда', 'зачем', 'всех', 'никогда', 'можно', 'при', 'наконец', 'два', 'об', 'другой', 'хоть', 'после', 'над', 'больше', 'тот', 'чerez', 'эти', 'нас', 'про', 'всего', 'них', 'какая', 'много', 'разве', 'три', 'эту', 'моя', 'впрочем', 'хорошо', 'свою', 'этой', 'перед', 'иногда', 'лучше', 'чуть', 'том', 'нельзя', 'такой', 'им', 'более', 'всегда', 'конечно', 'всю', 'между', 'как', 'что', 'где', 'какой', 'для', 'более', 'чем', 'ещё', 'быть', 'стать', 'мой', 'это', 'можно', 'всё', 'почти', 'парка', 'после', 'первый', 'четыре', 'пять', 'шесть', 'семь', 'тысяча', 'самый', 'конец', 'год', 'который', 'также', 'весь', 'свой', 'сергей', 'число', 'около', 'имя', 'кроме', 'каждый', 'корп', 'москва']"
```

In [14]:

```
# Загрузка словаря нормальных форм слов для ускорения работы
if (DATA_DIR/'normal_forms.csv').exists():
    dict_nf = dict(pd.read_csv(DATA_DIR/'normal_forms.csv', index_col=0, squeeze=True))
else:
    dict_nf = dict()

# Функция для преобразование html-кода в текст. Текст пришлось дополнительно чистить от блоков ссылок
bt = lambda x:BeautifulSoup(re.sub(r'<p class="mceNonEditable inline_question_history">.*?</p>', '', x)).get_text()

# Функция для получения нормальной формы списка слов с буферным словарем для ускорения работы
text_normalize = lambda x: list(
    dict_nf[word] if word in dict_nf
    else dict_nf.setdefault(word, morph.normal_forms(word)[0].upper())
    for word in x
)

# Функция для удаления стоп-слов
del_stop = lambda x: [word for word in x if word.lower() not in stop]
```

In [15]:

```
# Основная функция лемматизации и нормализации корпуса - создаем BOW (bag of words)
def text_to_BOW(text_input):
    return del_stop(text_normalize(re.findall(r'\b(?<!-) [а-я][а-я-]+[а-я](?!-)\b', bt(text_input).lower())))

# Формирование списков нормализованных слов для заголовка и текста каждой новости
df_news['words_title'] = df_news.title.apply(text_to_BOW)
df_news['words_text'] = df_news.full_text.apply(text_to_BOW)

# Сохраняем словарь нормальных форм слов
pd.Series(dict_nf).to_csv(DATA_DIR/'normal_forms.csv')
```

In [16]:

```
# 100 самых часто-встречающихся слов в корпусе за август
w = pd.Series(chain.from_iterable(df_news.words_text))
w.str.lower().value_counts().head(100).index
```

```
Out[16]: Index(['дом', 'работа', 'новый', 'город', 'проект', 'московский', 'центр',  
    'время', 'улица', 'человек', 'мочь', 'программа', 'станция',  
    'городской', 'место', 'столица', 'здание', 'получить', 'территория',  
    'объект', 'день', 'большой', 'работать', 'ребёнок', 'школа', 'площадка',  
    'развитие', 'процент', 'рассказать', 'департамент', 'помощь', 'район',  
    'строительство', 'площадь', 'система', 'метр', 'участник', 'специалист',  
    'житель', 'детский', 'миллион', 'пройти', 'появиться', 'парк',  
    'участок', 'москвич', 'россия', 'услуга', 'социальный', 'музей',  
    'сделать', 'столичный', 'комплекс', 'создать', 'современный', 'рубль',  
    'линия', 'организация', 'электронный', 'отметить', 'общий', 'часть',  
    'построить', 'участие', 'медицинский', 'смочь', 'сервис', 'метро',  
    'мэр', 'вид', 'вопрос', 'возможность', 'хороший', 'начало',  
    'находиться', 'технология', 'мероприятие', 'принять', 'портал', 'жизнь',  
    'руководитель', 'век', 'зона', 'главный', 'проводести', 'установить',  
    'например', 'история', 'квадратный', 'компания', 'проходить',  
    'несколько', 'государственный', 'документ', 'наш', 'пациент', 'сегодня',  
    'спортивный', 'корпус', 'решение'],  
   dtype='object')
```

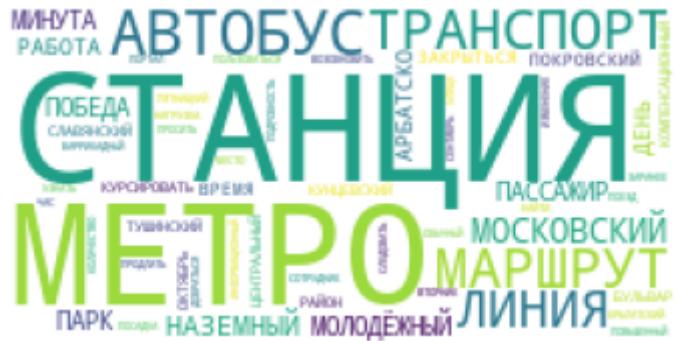
```
In [17]: # Функции для рисования качества словарного корпуса  
# Очень впечатляет и сразу дает представление об объекте. Размер слов имеет значение!  
def pic_profile(words, random_state=10):  
    wordcloud = WordCloud(width=800, height=400, background_color="white", random_state=random_state).generate(  
        ' '.join(pd.Series(words).sample(frac=1)))  
    plt.figure(figsize=(18, 18))  
    plt.imshow(wordcloud)  
    plt.axis("off")  
    plt.show()  
  
def ava_profile(words, random_state=None):  
    wordcloud = WordCloud(width=240, height=120, background_color="white", random_state=random_state).generate(  
        ' '.join(pd.Series(words).sample(frac=1)))  
    return wordcloud
```

```
In [18]: pic_profile(w)  
  
# Полный профиль всего новостного корпуса за август 2021
```



In [19]:

```
plt.imshow(ava_profile(chain.from_iterable(df_news.words_text.iloc[1:2])))
plt.axis("off")
plt.show()
```



```
In [20]: fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})

for ax, title, words_text in zip(axs.flat, df_news.title.iloc[:20], df_news.words_text.iloc[:20]):
    ax.imshow(ava_profile(words_text))
    ax.axis("off")
    ax.set_title(title[:30]+'...')

plt.tight_layout()
plt.show()

# Новости. Картина дня, например
```

Открыта запись на электронное ..  
PORTAL ПОЛУЧИТЬ ЛИЧНЫЙ  
Голосование  
РОССИЯ ОНЛАЙН УЧАСТИК  
ЭЛЕКТРОННЫЙ БЮЛЛЕТЕНЬ ИЮНЬ  
УЧАСТИЕ НЕОБХОДИМО  
ПРОГЛОСОВАТЬ

«По масштабам Вселенной 90 лет..  
КОСМОНАВТ БОЛЬШОЙ ЗВЕЗДА  
ПЛОЩАДКА ТЕАТР НЕБО УВИДЕТЬ  
МУЗЕЙ ЗВЁЗДНЫЙ МЕСТО  
НАШ ЧЕЛОВЕК СЕГОДНЯ  
ПЛАНЕТАРИЙ ТЕХНОЛОГИЯ НАГРУЗКА  
ФИЛЬМ АСТРОНОМИЧЕСКИЙ НАСТОЯЩИЙ

«Зимний дрифт»: как прошли общ..  
МОТОКРЮСС СОРЕВНОВАНИЕ ДРИФТ  
ВИД БУДУЩЕЕ СПОРТ ПАРК ТРАССА  
САМОПЕМЕТ ГОРОД МЕСТО ПИЛОТ

Сыграть в «Мафию» с Вием и пос..  
БИБЛИОТЕКА УЛИЦА ПИСАТЕЛЬ ДОМ СМОЧЬ  
ПРОЙТИ ГОГОЛЬ БИЛЕТ ДЕНЬ  
ГОСТЬ АПРЕЛЬ НИКОЛАЙ НИКОЛАЙ

На портале «Узнай Москву» появ..  
МАРШРУТ ИНТЕРАКТИВНЫЙ ИЗМАЙЛОВСКИЙ  
ПРОЕКТ ПАРОВОЙ ТЕХНОЛОГИЯ  
ПРУД ИНТЕРЕСНЫЙ СУХОЙ  
ДЕПАРТАМЕНТ БАШНИНСКИЙ

Для пассажиров закрытого участ..  
СВИАЖСКИЙ АВТОБУС ТРАНСПОРТ  
СТАНЦИЯ МЕТРО МОСКОВСКИЙ  
ЛИНИЯ РАЙОН ПАССАЖИР  
ПАРК МАРШРУТ МИНУТА  
УЧАСТИЕ НЕОБХОДИМО  
ПРОГЛОСОВАТЬ

Фастфуд XVIII века и другие со..  
ДАННЫЕ РУБАТЬ ВЕК ПОЭТ  
ГОРОД ОПИСАНИЕ СКОЛЬКО  
КАЛАШНИКOV НАЧАТЬ ИМПЕРАТОРСКИЙ  
ПУТЕВОДИТЕЛЬ ПОСЛЕДНИЙ  
НАЗВАНИЕ ДЕЛЕНИЕ АИЗБА-  
АДМИНИСТРАТИВНАЯ РИА ОРГАНЫ АДМИНИСТРАЦИИ

Число высокоточных МРТ-исследо..  
МРТ ПАЦИЕНТ БОЛЬНИЦА СТАЦИОНАР  
ДОСТУПНЫЙ ЕДИНЫЙ АППАРАТ  
МЕДИЦИНСКИЙ ВОЗМОЖНОСТЬ  
ОБОРУДОВАНИЕ БЕЗОГРАНИЧЕННО  
ИССЛЕДОВАНИЕ РАБОТАТЬ  
АНАЛИТИЧЕСКИЙ РЕЗОНАНСНЫЙ

Объявлены победители междунаро..  
КОНЦЕПЦИЯ КОМПЛЕКС  
ПРОГРАММА КВАРТАЛ СТОЛИЦА  
КОНКУРС ОБЛIMK СРЕДА  
АРХИТЕКТУРНЫЙ КАЧЕСТВОВАНО  
ПОЛУЧИТЬ РЕНОВАЦИЯ  
ИНИЦИАТИВА РЕНОВАЦИЯ  
РЕНОВАЦИЯ ЭКСПЕРТ

Город начал разбирать объекты ..  
МЕТРОВЫЙ АДРЕС ДЕМОНТИРОВАТЬ КВАДРАТНЫЙ  
САМА РУБЛЬ БЕЛЫЙ КОМПЛЕКС  
СОБСТВЕННИК КОМПЕТЕНЦИЯ СОГЛАСИТЬСЯ  
ПОМОТЬ ОКТЕБРЕЙСКИЙ УЛИЦА  
ОБЪЕКТ ДЕМОНТАЖ СНОС  
ПРУД ВЛАДЕЛЕЦ УДАЧА  
ДЕПАРТАМЕНТ ВЛАДЕЛЕЦ СПИСОК

Москвичка Арина Аверина выигра..  
УПРАЖНЕНИЕ БАЛЛ ЧЕМПИОН  
ЗАВЕДЕВАТЬ МАРИЯ ДИН СБОРНАЯ  
ДАРЫ МАРГАРITA АРИН МЕСТО  
ОБРУЧЫ ИГРАТЬ АНАСТАСИЯ ЗАНЯТЬ ЕВРОПА  
СОРЕВНОВАНИЕ ПАПЛОВ РЕЗУЛЬТАТ

Главные стройки Москвы: какие ..  
ОБЪЕКТ ПРОЕКТ ПЛОЩАДЬ  
ЗДАНИЕ ПОСТРОИТЬ ЦЕНТР  
ГЛАВНЫЙ ХОРОШИЙ ПОБЕДИТЬ  
ПОСТРОИТЬ СТИЛЬ  
ПОСТРОИТЬ ДОМЕЦ  
ПОСТРОИТЬ ВОССЕ  
ПОСТРОИТЬ КОМПЛЕКС

Москва окажет финансовую подде..  
МИЛЛИОН РУБЛЬ  
СУММА ЖИВОПИСЬ  
ГРАНТ АНИМАЦИОННЫЙ  
АНИМАЦИОННЫЙ РУБЛЬ

Больше миллиона многолетников ..  
ПЛОЩАДЬ МНОГОЛЕТНИК  
МНОГОЛЕТНИК КОМПОЗИЦИЯ  
РАСТЕНИЕ ЦВЕТ  
МНОГОЛЕТНИЙ ОСЕНЬ  
МНОГОЛЕТНИЙ СЛОВА  
МНОГОЛЕТНИЙ МАССА

Многофункциональный комплекс ..  
КАФЕ ДЕПАРТАМЕНТ АПАРТАМЕНТ  
ДУБРОВКА ТРАНСПОРТНЫЙ  
СРЕДСТВО СОЗДАНИЕ  
КОМПЛЕКС ИМПЕРІАЛЬНО  
ОБЪЕКТ СПОРТИВНЫЙ  
РАЗМЕСТИТЬСЯ СПУСК  
ПРИЛОЖЕНИЕ  
ПРИЛОЖЕНИЕ

В Текстильщиках демонтирована ..  
УЧАСТОК ЭТАЖ ПОМЕЩЕНИЕ  
ДОКУМЕНТАЦИЯ ВОЗВЕСТИ  
УЛИЦА ЗЕМЕЛЬНЫЙ РАЗРЕШИТЕЛЬНЫЙ  
ПРИСТРОЙКА

Планируйте маршрут: на Савелов..  
МОСКВА БЕССЫКОВОЙ УЧАСТОК  
ПУТЬ УКЛАДКА  
РАСПИСАНИЕ  
МОСКОВСКИЙ РЕВЕРС

Реконструкцию Старо-Рублевског..  
ПУТЕПРОВОД НАПРАВЛЕНИЕ  
УЛИЦА ПУТЬ  
ДОРОГА СОЕДИНИТЬ  
МЖД СТРАТЕГИЧЕСКИЙ  
ПУТЬ ПЕРЕСЕЧЕНИЕ  
СОЕДИНИТЬ КОМПЛЕКС  
СЕКЦИЯ СЧЁТ  
КРУЖКА УСЛУГА  
ПОЛЬЗОВАТЕЛЬ УКАЗАТЬ  
ЗАПИСЬ СОДЕЙСТВИЕ

In [21]:

```
# Функция для расчета idf для всех слов корпуса, inverse document frequency – обратная частота документа
def calc_idf(corpus):
    return (np.log10(len(corpus)))
```

```
-np.log10(pd.Series(corpus).explode().reset_index().drop_duplicates().iloc[:,1].value_counts()))

# Функция для расчета tf для всех слов текста документа, term frequency – частота слова
def calc_tf(text):
    return pd.Series(Counter(text))/len(text)

# Функция для расчета dw для всех документов, document weight – удельный вес документа в корпусе
def calc_dw(corpus):
    return pd.Series(corpus.str.len())/len(corpus)
```

In [22]:

```
# Пример calc_idf
calc_idf(df_news.words_text)
```

Out[22]:

```
НОВЫЙ          0.245963
ГОРОД          0.246197
РАБОТА          0.251722
ВРЕМЯ          0.287718
МОСКОВСКИЙ     0.333347
...
КУЗНЯ          3.816506
НЕМОЧНЫЙ       3.816506
СИРЫЙ          3.816506
МУЗШКОЛА        3.816506
БЛАГОПРИЯТСТВОВАТЬ 3.816506
Name: words_text, Length: 54215, dtype: float64
```

In [23]:

```
# Пример calc_tf
calc_tf(df_news.words_text.iloc[0]).nlargest()
```

Out[23]:

```
ГОЛОСОВАНИЕ    0.043393
ЭЛЕКТРОННЫЙ   0.031558
УЧАСТИЕ         0.017751
ЗАЯВКА          0.017751
ИЮНЬ            0.017751
dtype: float64
```

In [24]:

```
# Пример calc_dw
calc_dw(df_news.words_text).nlargest()
```

Out[24]:

```
id
7465050      0.631218
4851050      0.607568
```

```
72877073    0.601160
6629050     0.553860
72684073    0.486878
Name: words_text, dtype: float64
```

```
In [25]: # Расчет tf_idf для всех слов во всех документах
tfs = pd.concat([calc_tf(x) for x in tqdm(df_news.words_text)], keys=df_news.words_text.index, names=['id', 'words'])
tf_idfs = tfs.swaplevel(0, 1).mul(calc_idf(df_news.words_text), level='words').sort_index()
tf_idfs
```

100% | ██████████ | 6554/6554 [00:05<00:00, 1285.82it/s]

```
Out[25]: words      id
ЁЖ        4174050  0.004932
          15535073  0.016490
          61031073  0.004008
          72007073  0.001808
          84196073  0.004744
          ...
ЯШИК      95258073  0.002354
ЯШИК-КОРПУС 91851073  0.004322
ЯЩИЧЕК     76852073  0.008774
ЯЩИЧЕК-ДОМИК 95204073  0.004148
ЯЩИЧНЫЙ    76852073  0.008774
Length: 1283315, dtype: float64
```

```
In [26]: # Функция оценки схожести текста с документами корпуса
def score_tdidf(text_BOF, tf_idfs_corpus):
    # если слово в тексте встречается несколько раз, учитываем это, рассчитав мультипликатор
    mult = pd.Series(text_BOF).value_counts()
    return tf_idfs_corpus.loc[mult.index].mul(mult, level='words').groupby(level='id').sum()
```

```
In [27]: text = df_news.words_text.loc[6751050]
str(text[:100])
```

```
Out[27]: "[ 'СЕНТЯБРЬ', 'СОБЯНИН', 'ПОСЕТИТЬ', 'КОРПУС', 'ШКОЛА', 'СТАРШЕКЛАССНИК', 'ШКОЛА', 'ПОЗДРАВИТЬ', 'УЧИТЬСЯ', 'ПЕДАГОГ', 'ДЕНЬ', 'ЗНАНИЕ', 'СЕГОДНЯ', 'УДИВИТЕЛЬНЫЙ', 'ДЕНЬ', 'СЕНТЯБРЬ', 'ДЕНЬ', 'ЗНАНИЕ', 'ПРИВЫКНУТЬ', 'ПРИХОДИТЬ', 'ШКОЛА', 'СЕГОДНЯ', 'ВДВОЙНЕ', 'ВОЛНІТЕЛЬНЫЙ', 'ПРАЗДНИК', 'МНОГІЕ', 'РЕБЁНОК', 'ПОЛГОДА', 'ВІДЕТЬ', 'ДРУГ', 'ДРУГ', 'КЛАСС', 'ПЕДАГОГ', 'ВОЛНЕННІЕ', 'ПРИМЕШИВАТЬСЯ', 'ТРЕВОГА', 'ОРГАНИЗОВАТЬ', 'ПРОЦЕСС', 'РИСК', 'СНОВА', 'ШКОЛА', 'СОЖАЛЕНИЕ', 'ЭПІДЕМІЯ', 'ЗАКОНЧИТЬСЯ', 'СКАЗАТЬ', 'СОБЯНИН', 'МЕНЕЕ', 'СЛОВО', 'МЭР', 'ДЕТСКІЙ', 'ДОШКОЛЬНИЙ', 'УЧРЕЖДЕНИЕ', 'ШКОЛА', 'ВУЗ', 'ГОРОД', 'ПРИСТУПИТЬ', 'РАБОТА', 'ДЕНЬ', 'ХОТЕТЬ', 'ПОЖЕЛАТЬ', 'ЗДОРОВЬЕ', 'НОВЫЙ', 'ЗНАНИЕ', 'УСПЕХ', 'ДОБРЫЙ', 'НАСТРОЕНИЕ', 'УЧИТЬСЯ', 'ПЕДАГОГ', 'ДОБАВИТЬ', 'СОБЯНИН', 'МОСКОВСКИЙ', 'ШКОЛА', 'ЮЖНЫЙ', 'ТУШИНО', 'КРУПНЫЙ', 'ОБРАЗОВАТЕЛЬНЫЙ', 'КОМПЛЕКС', 'СОСТОЯТЬ', 'ЗДАНИЕ', 'УЧИТЬСЯ', 'РЕБЁНОК', 'РАБОТАТЬ',
```

```
'СОТРУДНИК', 'УЧИТЕЛЬ', 'ВОСПИТАТЕЛЬ', 'ПЕДАГОГИЧЕСКИЙ', 'РАБОТНИК', 'СОТРУДНИК', 'ШКОЛА', 'УДОСТОИТЬ', 'ЗВАНИЕ', 'ЗАСЛУЖИТЬ', 'УЧИТЕЛЬ', 'РОССИЙСКИЙ', 'ФЕДЕРАЦИЯ', 'ПРЕПОДАВАТЕЛЬ', 'ЯВЛЯТЬСЯ', 'ПОЧЁТНЫЙ', 'РАБОТНИК', 'ОБРАЗОВАНИЕ']"
```

```
In [28]: score_tdidf(text, tf_idfs).nlargest(10)
```

```
Out[28]: id
6751050    4.070813
88732073    3.168164
4960050    2.733516
87414073    2.691670
5906050    2.644962
49617073    2.485554
95018073    2.479252
5479050    2.434351
43323073    2.421520
91270073    2.400217
dtype: float64
```

```
In [29]: # Максимальная схожесть на исходной новости, на высоких позициях похожие новости
# Получилось то что надо и работает быстро. Для удобства дальнейшей работы соберем созданные функции в класс
```

### Описание класса TfIdf\_model()

Решили самостоятельно написать быструю векторную реализацию TF-IDF, чтобы уложится в требования по скорости работы модели (< 1 сек.)

---

Существует несколько методов оценки соответствия документа  $D$  из корпуса текстов запросу  $Q$  на основе TF-IDF.

---

Классический метод оценки:

$$score(D, Q) = \sum_{i=1}^n \text{TF}(q_i, D) \cdot \text{IDF}(q_i),$$

где  $Q$  — запрос, состоящий из слов  $q_1, \dots, q_n$ ,

$\text{TF}(q_i, D) = \frac{qty(q_i, D)}{|D|}$  — частота слова (term frequency)  $q_i$  в документе  $D$ ,

$qty(q_i, D)$  — количество слов  $q_i$  в документе  $D$ , а  $|D|$  — общее количество слов в этом документе,

$\text{IDF}(q_i)$  — обратная частота документа (inverse document frequency) для слова  $q_i$ .

---

Также реализовали ряд методов из семейства BM (best match), а именно - BM25, BM15, BM11, используемые поисковыми системами для упорядочивания документов по их релевантности поисковому запросу. Наиболее распространенным из них является метод ранжирования BM25, который часто называют «Okapi BM25» по названию поисковой системы Okapi ([https://ru.wikipedia.org/wiki/Okapi\\_BM25](https://ru.wikipedia.org/wiki/Okapi_BM25)).

$$\text{score}_{BM25}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{3 \cdot \text{TF}(q_i, D)}{\text{TF}(q_i, D) + 0.5 + 1.5 \cdot \frac{|D|}{\text{avgdl}}},$$

$$\text{score}_{BM15}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{3 \cdot \text{TF}(q_i, D)}{\text{TF}(q_i, D) + 2},$$

$$\text{score}_{BM11}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{3 \cdot \text{TF}(q_i, D)}{\text{TF}(q_i, D) + 2 \cdot \frac{|D|}{\text{avgdl}}},$$

где  $\text{avgdl}$  — средняя длина документа в корпусе.

---

Классическая формула  $\text{IDF}(q_i)$ :

$$\text{IDF}(q_i) = \log \frac{N}{n(q_i)},$$

где  $N$  — общее количество документов в корпусе, а  $n(q_i)$  — количество документов, содержащих  $q_i$ .

Также применяется «сглаженный» вариант этой формулы:

$$\text{IDF}_{smooth}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

Эта формула имеет недостаток — для часто-встречающихся слов значение  $\text{IDF}_{smooth}$  отрицательно. Таким образом, при наличии двух почти идентичных документов, в одном из которых есть слово, а в другом — нет, второй может получить завышенную оценку. Устраним недостаток, обнулив отрицательные значения.

Отметим, что в библиотеке sklearn ([https://scikit-learn.org/stable/modules/feature\\_extraction.html#tfidf-term-weighting](https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting)) используются другие два варианта формулы IDF, которые отличаются от классического и «сглаженного».

$$\text{IDF}_{\text{sklearn}}(q_i) = \log \frac{N}{n(q_i)} + 1$$

$$\text{IDF}_{\text{sklearn-smooth}}(q_i) = \log \frac{1 + N}{1 + n(q_i)}$$

В классе `Tfidf_model()` мы реализовали все описанные выше варианты.

In [30]:

```
class Tfidf_model():
    """
    Быстрая векторная реализация TF-IDF
    """

    def __init__(self, method=None, method_idf=None):
        """
        Параметры:

        method = None | 'BM25' | 'BM15' | 'BM11'
        method_idf = None | 'smooth' | 'sklearn' | 'sklearn-smooth'

        Описание алгоритмов
        https://ru.wikipedia.org/wiki/Okapi_BM25
        https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting
        """
        self.method = method
        self.method_idf = method_idf

    # При обучении модели рассчитываем tf_idf для всех слов во всех документах
    def fit(self, corpus_BOW):

        # Количество документов, содержащий слово для каждого слова корпуса
        qnts = pd.Series(corpus_BOW).explode().reset_index().drop_duplicates().iloc[:,1].value_counts()

        # Расчет idf для всех слов корпуса, inverse document frequency – обратная частота документа
```

```

if self.method_idf == 'smooth':
    # считаем слаженную формулу IDF, отрицательные значения приравниваем нулю
    idfs = np.maximum(np.log(len(corpus_BOW) - qnts + 0.5) - np.log(qnts + 0.5), 0)
elif self.method_idf == 'sklearn':
    # считаем формулу IDF из sklearn
    idfs = np.log(len(corpus_BOW)) - np.log(qnts) + 1
elif self.method_idf == 'sklearn-smooth':
    # считаем формулу IDF из sklearn слаженную
    idfs = np.log(len(corpus_BOW) + 1) - np.log(qnts + 1)
else:
    # считаем классическую формулу IDF
    idfs = np.log(len(corpus_BOW)) - np.log(qnts)

# Расчет tf для всех слов текста документа, term frequency – частота слова
tfs = pd.concat([pd.Series(Counter(x)).div(len(x)) for x in corpus_BOW],
                 keys=corpus_BOW.index, names=['id', 'words'])

if self.method in ['BM25', 'BM11']:
    # Расчет dw для всех документов, document weight – удельный вес документа в корпусе
    dws = corpus_BOW.str.len()/np.mean(corpus_BOW.str.len())

# Расчет tf_idf для всех слов во всех документах
if self.method == 'BM25':
    # реализуем формулу BM25 в векторизованном виде
    self.tf_idfs = (
        tfs.mul(3).div(tfs.add(0.5).add(dws.mul(1.5), level='id'))
        ).swaplevel(0, 1).mul(idfs, level='words').sort_index()
elif self.method == 'BM15':
    # реализуем формулу BM15 в векторизованном виде
    self.tf_idfs = tfs.mul(3).div(tfs.add(2)).swaplevel(0, 1).mul(idfs, level='words').sort_index()
elif self.method == 'BM11':
    # реализуем формулу BM11 в векторизованном виде
    self.tf_idfs = (
        tfs.mul(3).div(tfs.add(dws.mul(2), level='id'))
        ).swaplevel(0, 1).mul(idfs, level='words').sort_index()
else:
    # реализуем формулу tf-idf в векторизованном виде
    self.tf_idfs = tfs.swaplevel(0, 1).mul(idfs, level='words').sort_index()

def get_tf_idfs(self):
    return self.tf_idfs

# Расчет рекомендательных оценок для документа X для документов с индексами ids
def predict(self, text_BOW, ids=None):

```

```
"""
ФУНКЦИЯ ОЦЕНКИ СХОЖЕСТИ ТЕКСТА С ДОКУМЕНТАМИ КОРПУСА
"""

mult = pd.Series(text_BOW).value_counts()
assert len(mult) > 1

return (
    self.tf_idfs.loc[mult.index]
    .mul(mult, level='words')
    .groupby(level='id').sum()
    .sort_values(ascending=False)
)
```

```
In [31]: model = Tfidf_model(method='BM15', method_idf='smooth')
model.fit(df_news.words_text)
```

```
In [32]: model.predict(text).head(20)
```

```
Out[32]: id
6751050    12.637878
88732073    9.419154
4960050     8.294909
87414073    8.200679
5906050     7.934871
95018073    7.345497
49617073    7.326384
91270073    7.071555
43323073    7.004632
5479050     6.918741
79177073    6.894715
6863050     6.862873
5825050     6.842533
4231050     6.711075
37885073    6.610411
7073050     6.534970
95230073    6.345946
6748050     6.324044
7614050     6.299830
30028073    6.266190
dtype: float64
```

```
In [33]: df_news_tags = df_news[['words_text', 'tag_ids']].explode('tag_ids')
```

```

df_tags['words_text'] = pd.Series([
    list(chain.from_iterable(df_news_tags.words_text[df_news_tags.tag_ids==x])) for x in df_tags.index
], df_tags.index)
df_tags.head(10)

```

Out[33]:

		title	created_at	qty	words_text
4019217	Сергей Собянин	2016-01-26 13:27:14	744	[ГРАДОСТРОИТЕЛЬНО-ЗЕМЕЛЬНЫЙ, КОМИССИЯ, ГОРОД, ...	
36217	строительство	2015-12-28 00:16:28	414	[ПРОГРАММА, РЕНОВАЦИЯ, ПОСТРОИТЬ, СОЦИАЛЬНЫЙ, ...	
47247217	коронавирус	2020-03-03 12:40:22	343	[ПРОШЕДШЕЕ, СУТКИ, ВЫЗДОРОВЕТЬ, ПАЦИЕНТ, ПРОХО...	
62217	парки	2015-12-28 00:16:28	310	[ПРОГУЛКА, ИЗМАЙЛОВСКИЙ, БАБУШКИНСКИЙ, ПОЗНАВА...	
25217	благоустройство	2015-12-28 00:16:28	294	[РЕКОНСТРУКЦИЯ, СЕВЕРО-ЗАПАД, СТОЛИЦА, ОТКРЫТЬ...	
5433217	Владимир Ефимов	2016-04-18 12:11:11	266	[ОДОБРИТЬ, ЗАЯВКА, ПРОМЫШЛЕННЫЙ, ПРЕДПРИЯТИЕ, ...	
3217	транспорт	2015-12-28 00:16:28	258	[РАСПИСАНИЕ, ПРИГОРОДНЫЙ, ПОЕЗД, САВЁЛОВСКИЙ, ...	
47576217	COVID-19	2020-03-16 11:46:31	221	[СВЯЗЬ, РОСТ, ЗАБОЛЕВАЕМОСТЬ, МОСКВИЧ, ПРИЗЫВА...	
16312217	программа реновации	2017-03-23 16:26:46	200	[ОБЪЯВИТЬ, ПОБЕДИТЕЛЬ, МЕЖДУНАРОДНЫЙ, КОНКУРС,...	
19925217	Алексей Фурсин	2017-07-19 17:50:16	188	[ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНЫЙ, КОМПЛЕКС, ТЕХНО...	

In [34]:

```

model = Tfidf_model(method='BM15', method_idf='smooth')
model.fit(df_tags.words_text)

```

In [35]:

```

text = df_news.loc[80375073].words_text

```

In [36]:

```

df_tags[['title']].assign(score = model.predict(text)).nlargest(10, 'score')

```

Out[36]:

	title	score
25831217	Арбатско-Покровская линия	6.836759
14244217	изменение маршрута	4.652652
4790217	компенсационные автобусы	4.189350

	title	score
<b>4677217</b>	ночные маршруты	3.918032
<b>32221217</b>	маршруты автобусов	3.918032
<b>5114217</b>	изменения расписания	3.554374
<b>5660217</b>	изменения маршрутов	3.431856
<b>34240217</b>	Савеловская	3.322157
<b>4806217</b>	компенсационные маршруты	3.122524
<b>4290217</b>	Волжская	3.016110

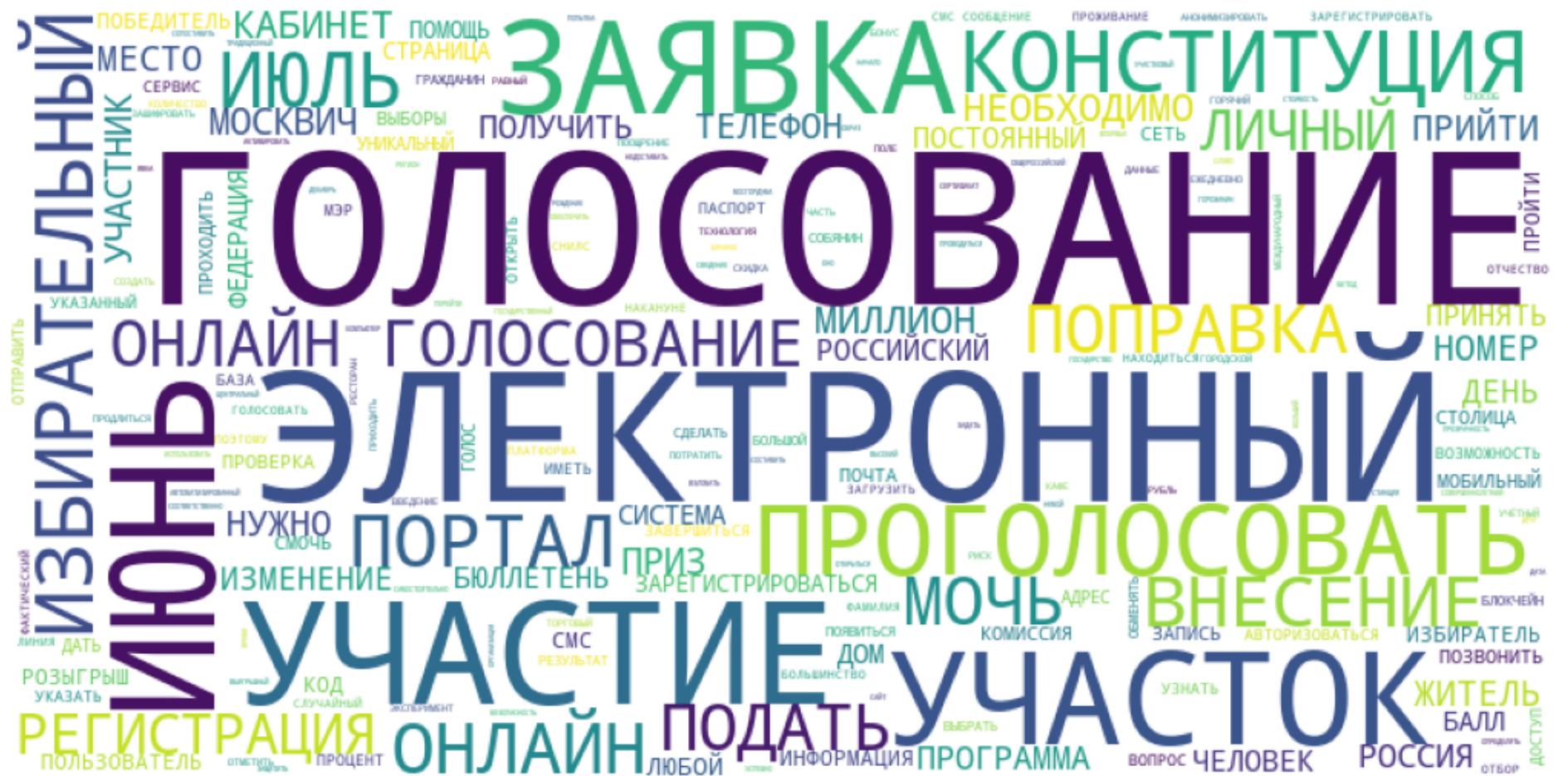
```
In [37]: df_tags[['title']].loc[df_news.loc[80375073].tag_ids]
```

```
Out[37]:
```

	title
<b>10217</b>	метро
<b>462217</b>	ограничения
<b>4790217</b>	компенсационные автобусы
<b>25641217</b>	Большая кольцевая линия
<b>25831217</b>	Арбатско-Покровская линия

```
In [38]: print(df_tags.loc[170217].title)
pic_profile(df_tags.loc[170217].words_text)
```

Конституция



In [39]:

```
print(df_tags.loc[57138217].title)
pic_profile(df_tags.loc[57138217].words_text)
```

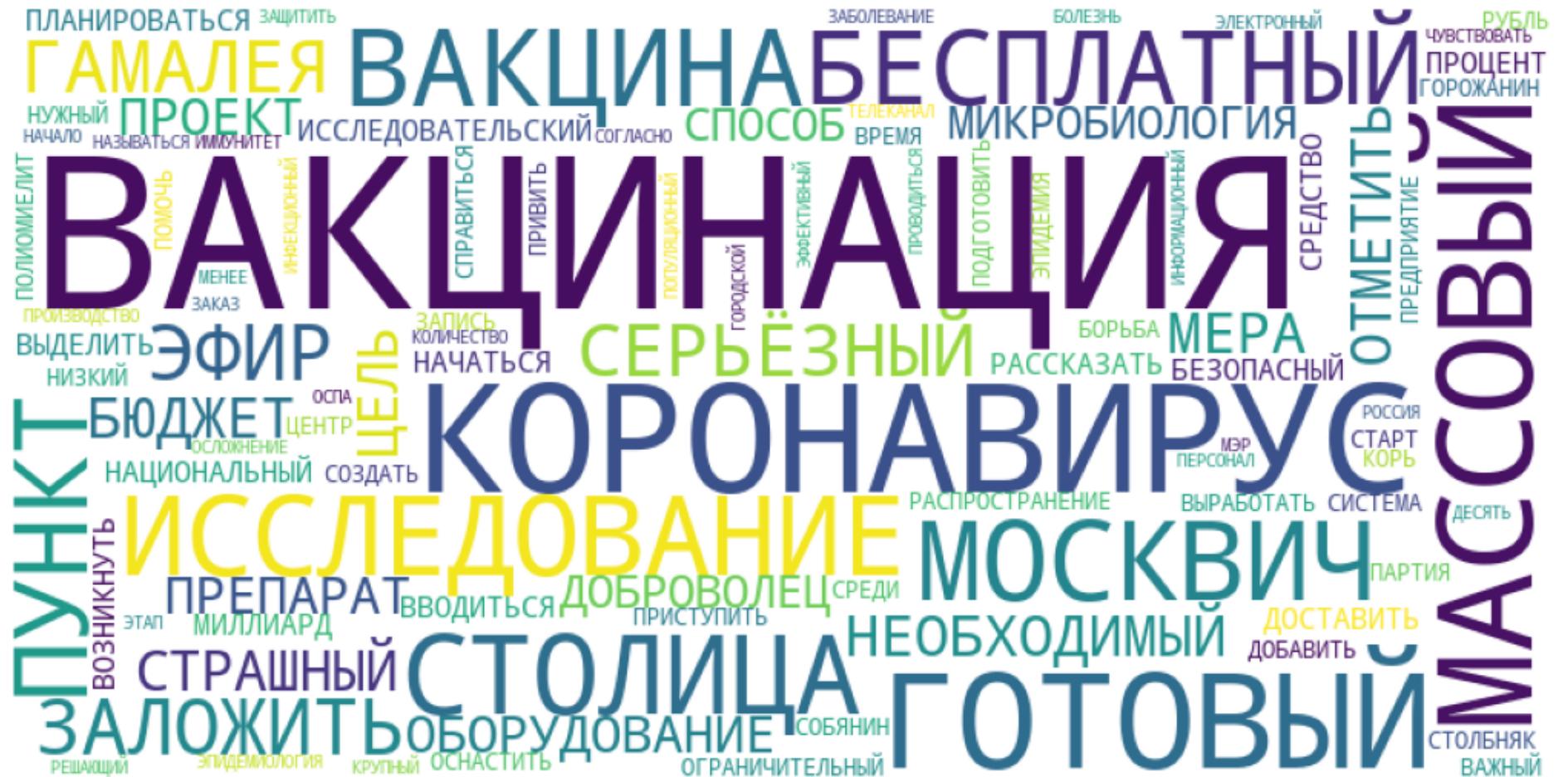
японские макаки



In [40]:

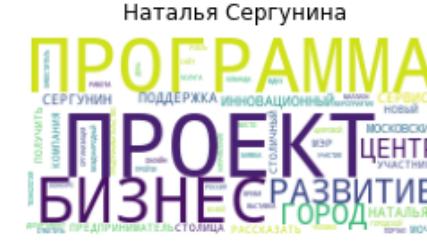
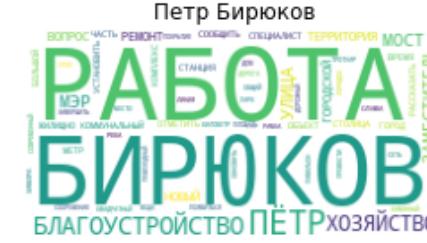
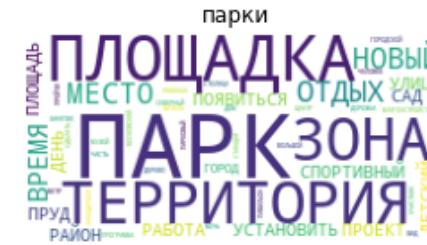
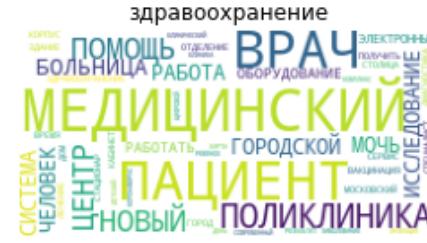
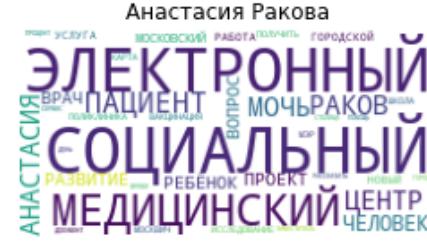
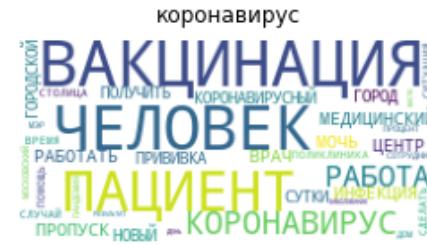
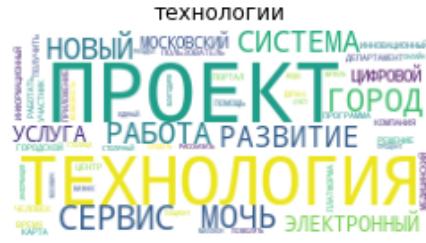
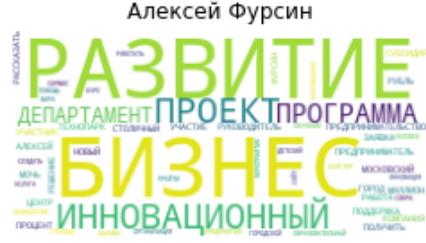
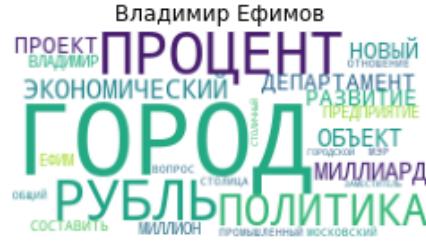
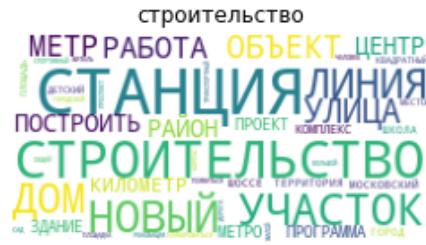
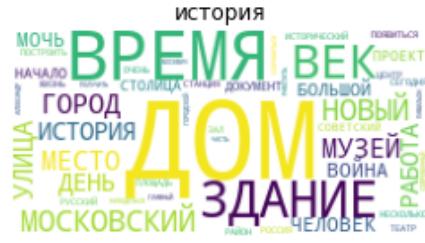
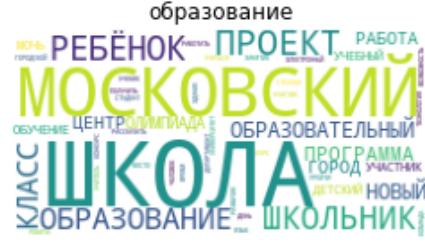
```
print(df_tags.loc[23373217].title)
pic_profile(df_tags.loc[23373217].words_text)
```

вакцина



In [41]:

```
fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})  
  
for ax, title, words_text in zip(axs.flat, df_tags.title.iloc[:20], df_tags.words_text.iloc[:20]):  
    ax.imshow(ava_profile(words_text))  
    ax.axis("off")  
    ax.set_title(title[:30]+('...' if len(title)>30 else ''))  
  
plt.tight_layout()  
plt.show()  
  
# Тэги
```



In [42]:

```
df_news_organizations = df_news[['words_text', 'organization_ids']].explode('organization_ids')
df_organizations['words_text'] = pd.Series([
    list(chain.from_iterable(df_news_organizations.words_text[df_news_organizations.organization_ids==x]))
```

```

for x in df_organizations.index
], df_organizations.index)
df_organizations.head(10)

```

Out[42]:

		name	has_reception	code	short_name	lead_institution_id	qty	words_text
	id							
20614090	Департамент информационных технологий города М...		1	dit	ДИТ	11491090	319.0	[ПРОГУЛКА, ИЗМАЙЛОВСКИЙ, БАБУШКИНСКИЙ, ПОЗНАВА...
9238090	Департамент предпринимательства и инновационно...		1	dpir	ДПИР	11491090	309.0	[ПРАВИТЕЛЬСТВО, УЧРЕДИТЬ, НОВЫЙ, ГРАНТ, ОБЩИЙ...
19180090	Департамент градостроительной политики города ...		1	dgp	ДГП	11491090	190.0	[ЮГО-ВОСТОК, РАЙОН, ЮЖНОПОРТОВЫЙ, СЧЁТ, СРЕДСТ...
19889090	Департамент жилищно-коммунального хозяйства го...		1	dgkh	ДЖХ	11491090	187.0	[ЗАВЕРШАТЬ, СЕЗОН, ВЫСАДКА, МНОГОЛЕТНИЙ, РАСТЕ...
12585090	Департамент культуры города Москвы		1	kultura	Депкульт	11491090	186.0	[ОНЛАЙН-ЗАЯВКА, ЗАЧИСЛЕНИЕ, КРУЖКА, СЕКЦИЯ, ПО...
20488090	Департамент культурного наследия города Москвы		1	dkn	ДКН	11491090	159.0	[КАНУН, ДЕНЬ, ВЛЮБИТЬ, СТОЛИЧНЫЙ, ДЕПАРТАМЕНТ...
103367090	Департамент инвестиционной и промышленной поли...		1	dipp	ДИПП	11491090	157.0	[СТОЛИЦА, НАЧИНАТЬСЯ, ПЯТЫЙ, СЕЗОН, ПРОЕКТ, ОТ...
20882090	Департамент транспорта и развития дорожно-тран...		1	dt	Дептранс	11491090	153.0	[СЕНТЯБРЬ, ОКТЯБРЬ, ЗАКРЫТЬСЯ, УЧАСТОК, АРБАТС...
20703090	Департамент природопользования и охраны окружа...		1	eco	ДПИООС	11491090	146.0	[СПЕЦИАЛИСТ, МОСПРИРОДА, СООБЩАТЬ, СКОРО, ПРИ...
9479090	Департамент здравоохранения города Москвы		1	dzdrav	Депздрав	11491090	143.0	[ГОРОДСКОЙ, КЛИНИЧЕСКИЙ, БОЛЬНИЦА, КОНЧАЛОВСКИ...

In [43]:

```

model = TfidfModel(method='BM15', method_idf='smooth')
model.fit(df_organizations.words_text)

```

<ipython-input-30-a38e4130b54b>:41: DeprecationWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.  
tfs = pd.concat([pd.Series(Counter(x)).div(len(x)) for x in corpus\_BOW],

```
In [44]: df_organizations[['name']].assign(score = model.predict(text)).nlargest(10, 'score')
```

Out[44]:

		name	score
	id		
<b>100297090</b>	Государственное природоохранное бюджетное учреждение города Москвы	4.465652	
<b>100173090</b>	Государственное бюджетное учреждение города Москвы	4.335641	
<b>100305090</b>	ГУП «Мосгортранс»	3.970252	
<b>20882090</b>	Департамент транспорта и развития дорожно-транс...	3.553694	
<b>20795090</b>	Департамент средств массовой информации и рекл...	1.985786	
<b>19267090</b>	Департамент строительства города Москвы	1.702489	
<b>19399090</b>	Комитет по архитектуре и градостроительству го...	1.611432	
<b>103466090</b>	Антитеррористическая комиссия города Москвы	1.327872	
<b>11491090</b>	Правительство Москвы	1.292867	
<b>19535090</b>	Департамент развития новых территорий города М...	1.232689	

```
In [45]: df_organizations[['name']].loc[df_news.loc[80375073].organization_ids]
```

Out[45]:

	name
	id
<b>20882090</b>	Департамент транспорта и развития дорожно-тран...

```
In [46]: print('Организация', df_organizations.loc[20882090, 'name'])
pic_profile(df_organizations.loc[20882090].words_text)
```

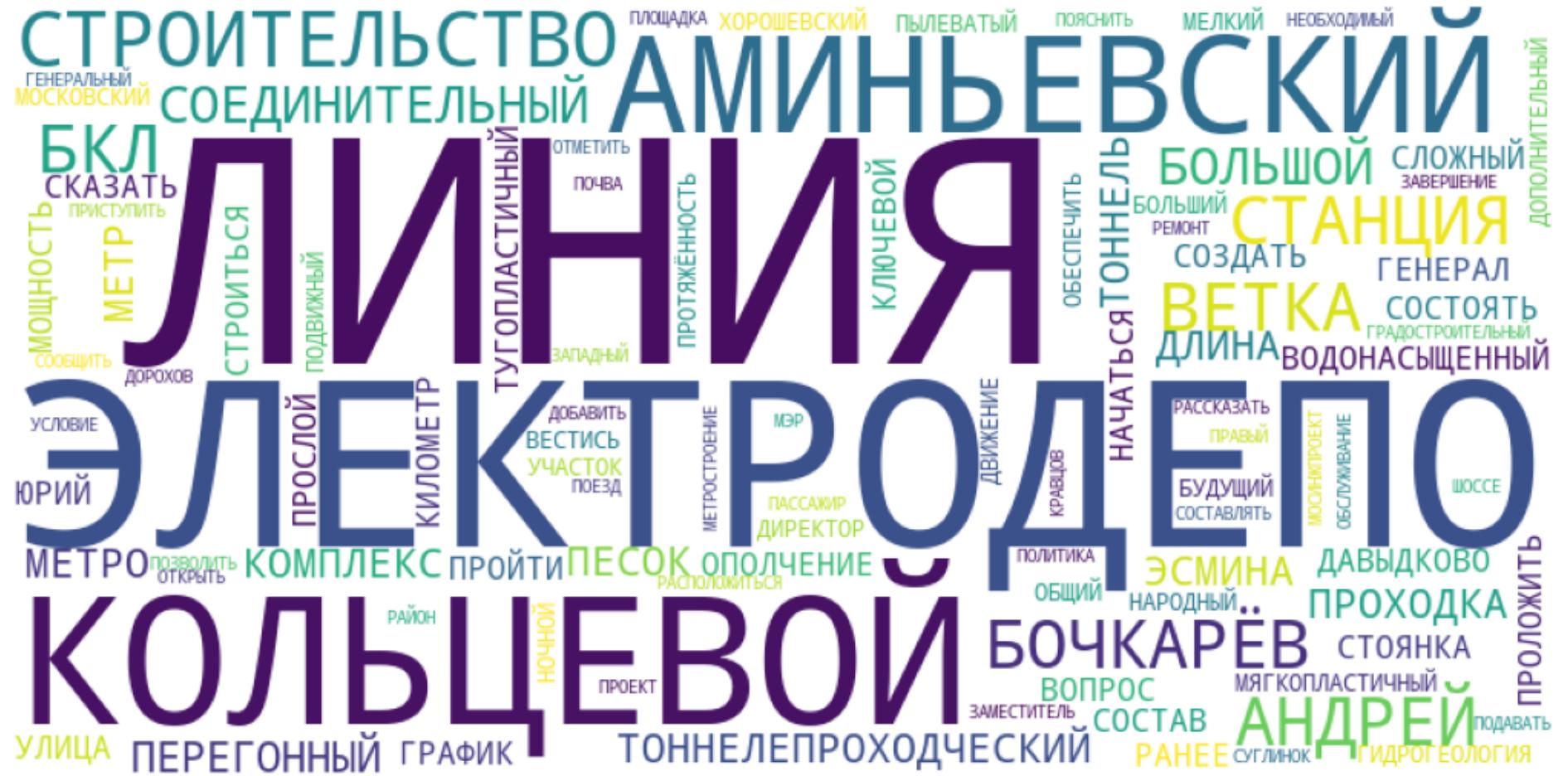
Организация Департамент транспорта и развития дорожно-транспортной инфраструктуры города Москвы



In [47]:

```
print('Организация', df_organizations.loc[100173090, 'name'])  
pic_profile(df_organizations.loc[100173090].words_text)
```

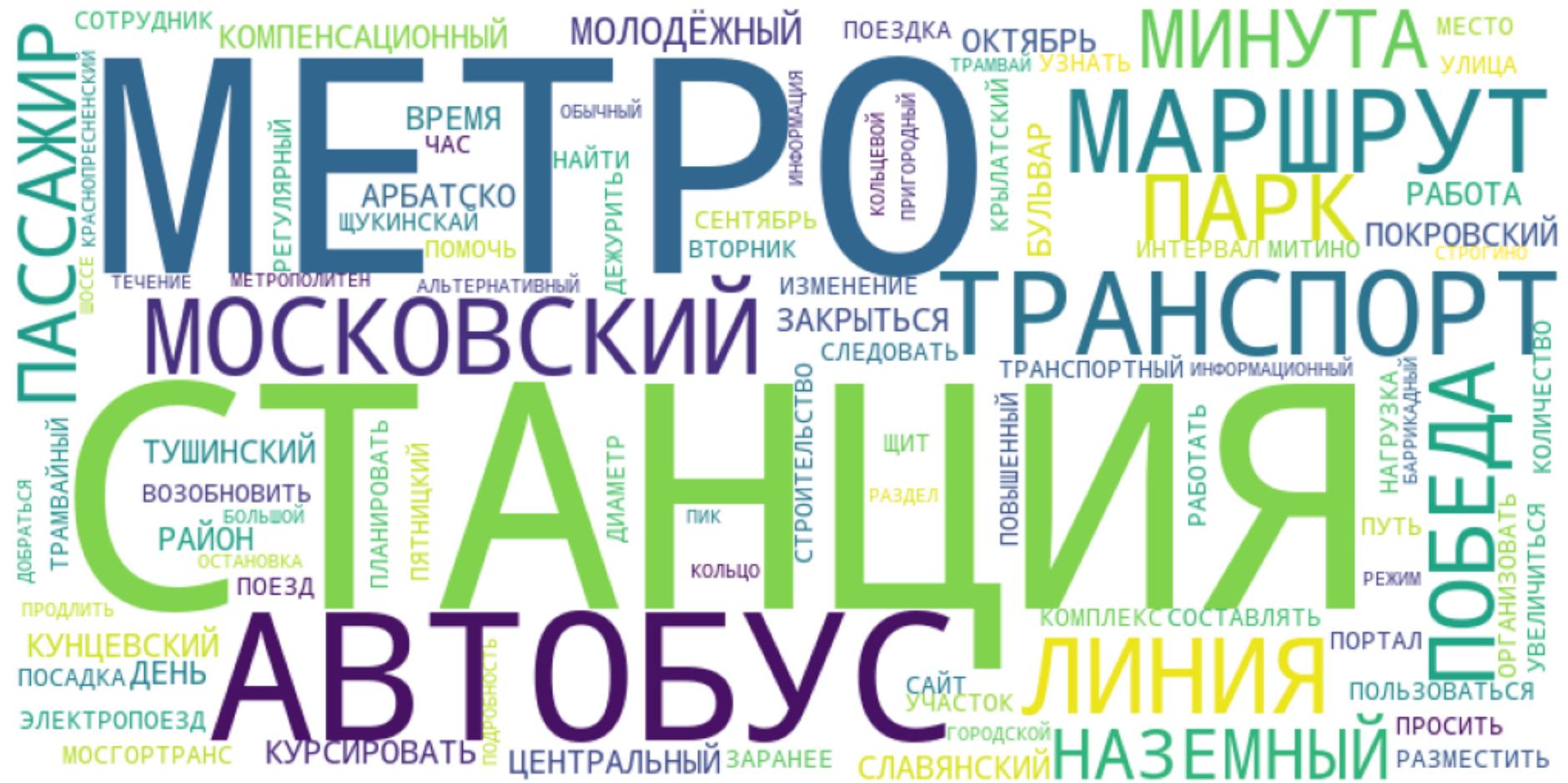
Организация Государственное бюджетное учреждение города Москвы "Информационно-аналитический центр Комплекса градостроительной политики и строительства города Москвы "Мосстройинформ"



In [48]:

```
print('Новость', df_news.loc[80375073, 'title'])  
pic_profile(df_news.loc[80375073].words_text)
```

Новость Для пассажиров закрытого участка Арбатско-Покровской линии пустили бесплатные автобусы КМ



In [49]:

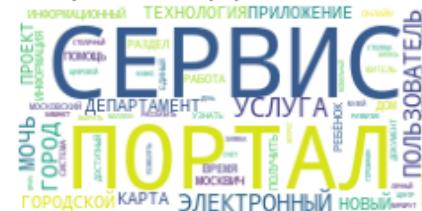
```
fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})

for ax, title, words_text in zip(axs.flat, df_organizations.name.iloc[:20], df_organizations.words_text.iloc[:20]):
    ax.imshow(ava_profile(words_text))
    ax.axis("off")
    ax.set_title(title[:30]+('...' if len(title)>30 else ''))

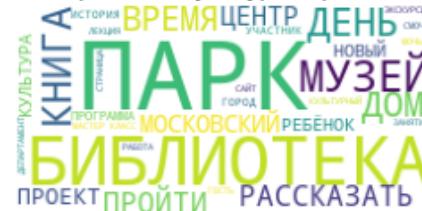
plt.tight_layout()
plt.show()

# Организации
```

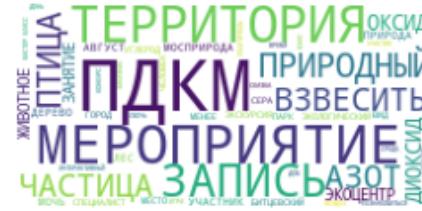
Департамент информационных тех..



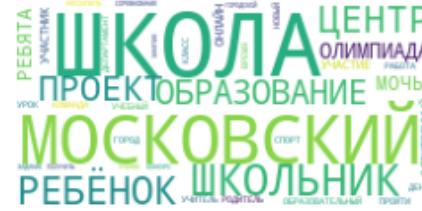
Департамент культуры города Мо..



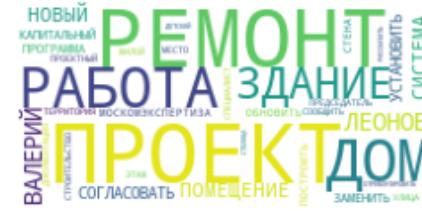
Департамент природопользования..



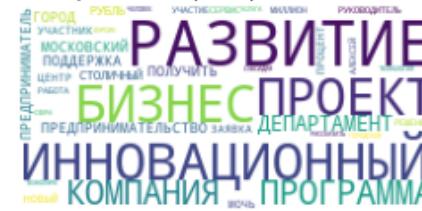
Департамент образования и наук..



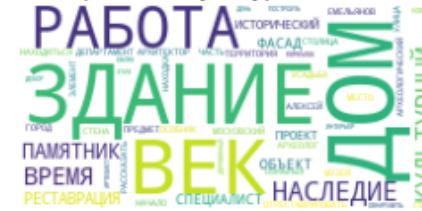
Комитет города Москвы по ценов..



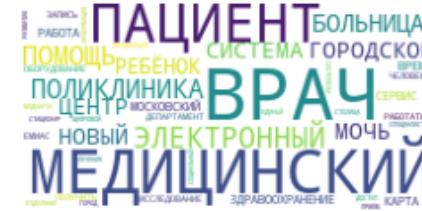
Департамент предпринимательств..



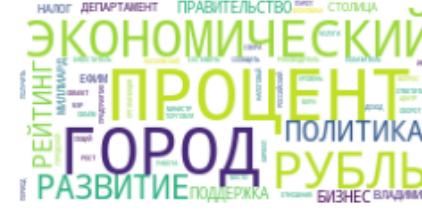
Департамент культурного наслед..



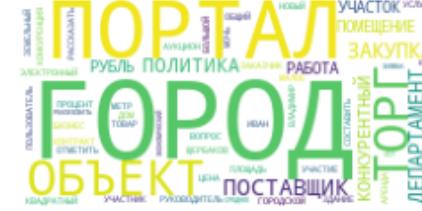
Департамент здравоохранения го..



Департамент экономической поли..



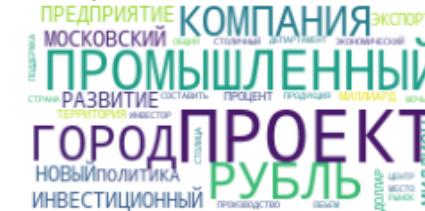
Департамент города Москвы по к..



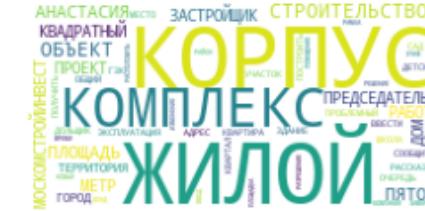
Департамент градостроительной ..



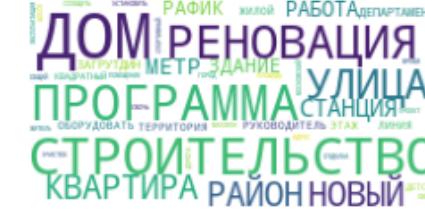
Департамент инвестиционной и п..



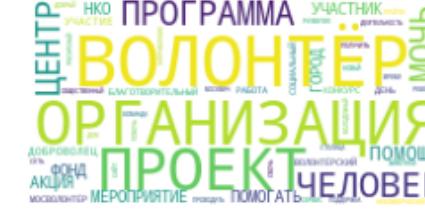
Комитет города Москвы по обесп..



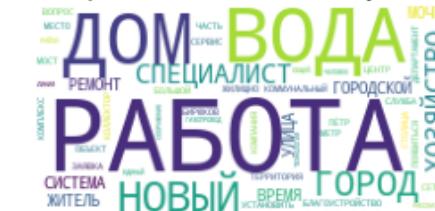
Департамент строительства горо..



Комитет общественных связей и ..



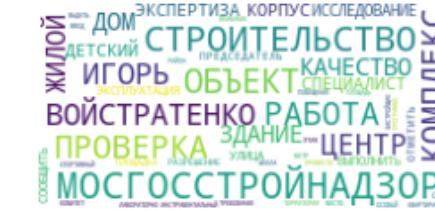
Департамент жилищно-коммунальн..



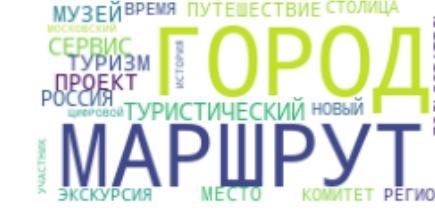
Департамент транспорта и разви..



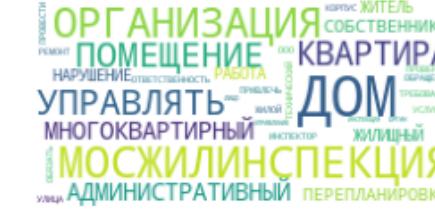
Комитет государственного строи..



Комитет по туризму города Моск..



Государственная жилищная инспе..



In [50]:

```
df_news_themes = df_news[['words_text', 'theme_ids']].explode('theme_ids')
df_themes['words_text'] = pd.Series([
    list(chain.from_iterable(df_news_themes.words_text[df_news_themes.theme_ids==x]))
```

```

for x in df_themes.index
], df_themes.index)
df_themes.head(10)

```

Out[50]:

		<b>title</b>	<b>created_at</b>	<b>updated_at</b>	<b>icon_id</b>	<b>url</b>	<b>qty</b>	<b>words_text</b>
<b>157287</b>		Строительство и благоустройство	2019-11-06 11:08:33	2021-07-02 17:43:19	4061.0	/news/maintheme/157287/	348	[МИТИН, ВВЕСТИ, ЭКСПЛУАТАЦИЯ, СПОРТИВНЫЙ, КОМП...
<b>115287</b>		Интересная Москва	2019-01-15 12:02:21	2021-07-06 20:04:25	8061.0	/news/maintheme/115287/	237	[НОЯБРЬ, ОТКРЫТЬСЯ, СТРАНА, МИР, ПЛАНЕТАРИЙ, С...
<b>27287</b>		Развитие метро	2017-11-03 22:23:24	2021-07-02 17:47:10	3061.0	/news/maintheme/27287/	99	[ПРОДЛЕНИЕ, ТРОИЦКИЙ, ЛИНИЯ, МЕТРО, СТАНЦИЯ, С...
<b>2287</b>		Планируйте маршрут	2017-11-03 22:23:24	2021-07-07 10:17:22	NaN	/news/maintheme/2287/	80	[РАСПИСАНИЕ, ПРИГОРОДНЫЙ, ПОЕЗД, САВЁЛОВСКИЙ, ...]
<b>99287</b>		Музейные истории	2018-06-07 15:19:55	2021-07-06 20:04:25	5061.0	/news/maintheme/99287/	77	[НОВЫЙ, ВЫПУСК, ИСТОРИЯ, ВЕЩЬ, РАССКАЗАТЬ, ПУТ...
<b>213287</b>		Здоровье и технологии	2021-05-19 12:06:26	2021-07-02 17:49:57	19061.0	/news/maintheme/213287/	63	[МОСКОВСКИЙ, ПОЛИКЛИНИКА, БОЛЬШОЙ, СТАНОВИТЬСЯ...]
<b>117287</b>		Семейные выходные	2019-04-08 13:04:35	2021-07-02 16:09:06	11061.0	/news/maintheme/117287/	48	[ВЫХОДНОЙ, СОТРУДНИК, МОСКОВСКИЙ, БИБЛИОТЕКА, ...]
<b>116287</b>		Советы библиотекаря	2019-03-18 12:23:01	2021-07-02 16:09:06	NaN	/news/maintheme/116287/	43	[ВЫХОДНОЙ, СОТРУДНИК, МОСКОВСКИЙ, БИБЛИОТЕКА, ...]
<b>60287</b>		Люди города	2017-11-03 22:23:24	2021-04-08 20:36:38	NaN	/news/maintheme/60287/	39	[НОЯБРЬ, ОТКРЫТЬСЯ, СТРАНА, МИР, ПЛАНЕТАРИЙ, С...
<b>210287</b>		Капремонт	2021-03-12 14:09:52	2021-06-18 18:29:01	NaN	/news/maintheme/210287/	37	[ВДНХ, ОТРЕМОНТИРОВАТЬ, СКЛАДСКОЙ, ПОМЕЩЕНИЕ, ...]

In [51]:

```

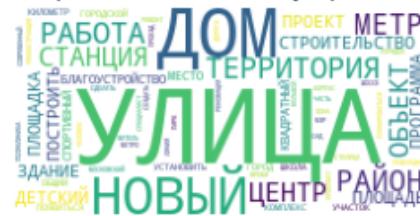
fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})

for ax, title, words_text in zip(axs.flat, df_themes.title.iloc[:20], df_themes.words_text.iloc[:20]):
    ax.imshow(ava_profile(words_text))
    ax.axis("off")
    ax.set_title(title[:30]+('...' if len(title)>30 else ''))

plt.tight_layout()
plt.show()

```

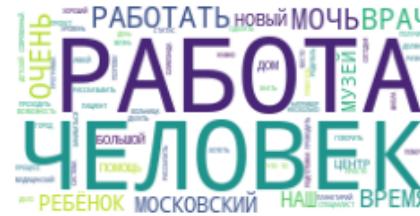
Строительство и благоустройств..



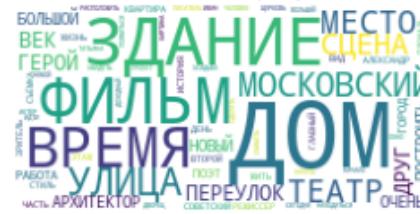
Музейные истории



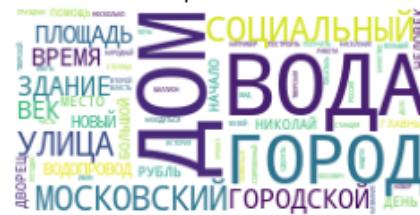
Люди города



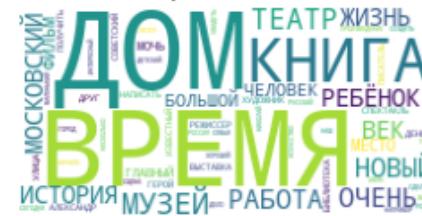
Московские прогулки



История Москвы



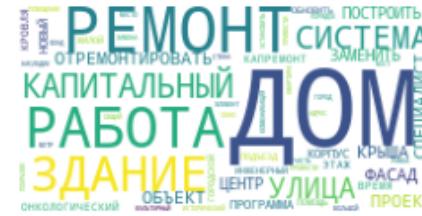
Интересная Москва



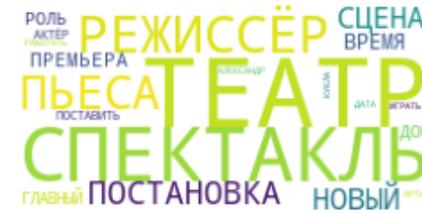
Здоровье и технологии



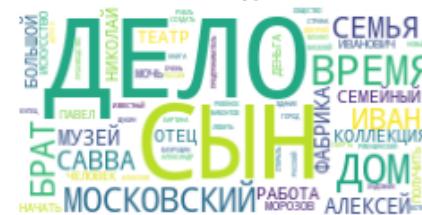
Капремонт



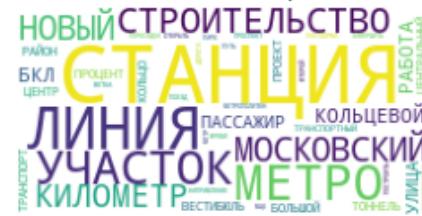
Жизнь театров



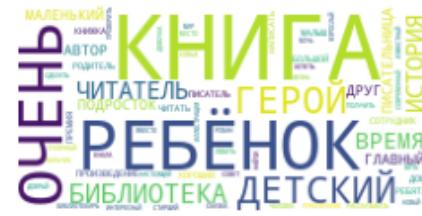
Московские династии



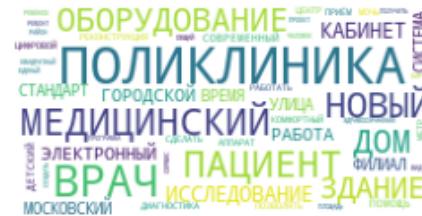
Развитие метро



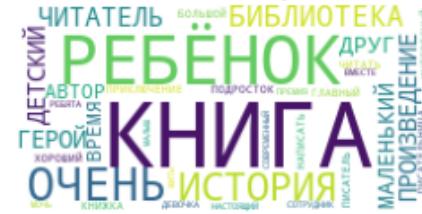
Семейные выходные



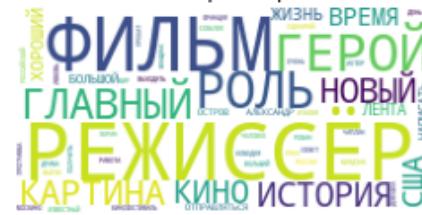
Ваша новая поликлиника



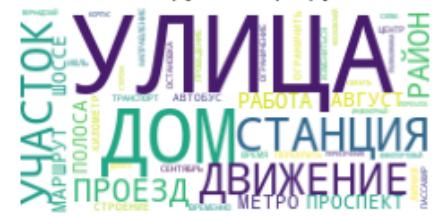
Книжный клуб



Кинопremьеры



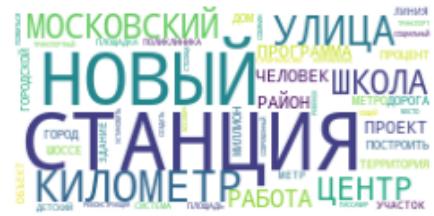
Планируйте маршрут



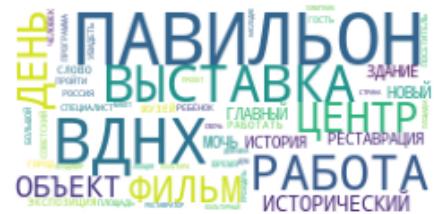
Советы библиотекаря



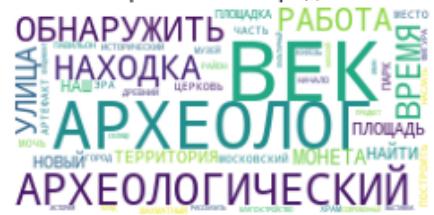
Реализация нацпроектов



ВДНХ



Археология города



```
In [52]: df_news_spheres = df_news[['words_text', 'sphere_ids']].explode('sphere_ids')
df_spheres['words_text'] = pd.Series([
    list(chain.from_iterable(df_news_spheres.words_text[df_news_spheres.sphere_ids==x]))
    for x in df_spheres.index
], df_spheres.index)
df_spheres.head(10)
```

Out[52]:

		title	special	activated	priority	qty	words_text
231299		Мой район	0	1	410	1416	[СЕНТЯБРЬ, ОКТЯБРЬ, ЗАКРЫТЬСЯ, УЧАСТОК, АРБАТС...
4299		Строительство и реконструкция	0	1	490	1095	[ЮГО-ВОСТОК, РАЙОН, ЮЖНОПОРТОВЫЙ, СЧЁТ, СРЕДСТ...
3299		Культура	0	1	360	986	[НОЯБРЬ, ОТКРЫТЬСЯ, СТРАНА, МИР, ПЛАНЕТАРИЙ, С...
12299		Экономика и предпринимательство	0	1	370	847	[ПРАВИТЕЛЬСТВО, УЧРЕДИТЬ, НОВЫЙ, ГРАНТ, ОБЩИЙ,...
1299		Социальная сфера	0	1	400	717	[ПОРТАЛ, ОТКРЫТЬСЯ, ЗАПИСЬ, УЧАСТИЕ, ОНЛАЙН-ГО...
5299		Городское хозяйство	0	1	460	651	[ЗАВЕРШАТЬ, СЕЗОН, ВЫСАДКА, МНОГОЛЕТНИЙ, РАСТЕ...
18299		Здравоохранение	0	1	500	606	[ГОРОДСКОЙ, КЛИНИЧЕСКИЙ, БОЛЬНИЦА, КОНЧАЛОВСКИ...
183299		Технологии	0	1	440	602	[ГОРОДСКОЙ, КЛИНИЧЕСКИЙ, БОЛЬНИЦА, КОНЧАЛОВСКИ...
2299		Транспорт	0	1	480	593	[СЕНТЯБРЬ, ОКТЯБРЬ, ЗАКРЫТЬСЯ, УЧАСТОК, АРБАТС...
15299		Образование	0	1	470	455	[ОНЛАЙН-ЗАЯВКА, ЗАЧИСЛЕНИЕ, КРУЖКА, СЕКЦИЯ, ПО...

```
In [53]: fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})

for ax, title, words_text in zip(axs.flat, df_spheres.title.iloc[:20], df_spheres.words_text.iloc[:20]):
    ax.imshow(ava_profile(words_text))
    ax.axis("off")
    ax.set_title(title[:30]+('...' if len(title)>30 else ''))

plt.tight_layout()
plt.show()

# Сфера
```

## Мой район

УЛИЦА  
РАБОТА  
НОВЫЙ ЦЕНТР  
ДОМ

## Социальная сфера

ЧЕЛОВЕК  
ЦЕНТР  
ПРОЕКТ  
РАБОТА  
УЛИЦА  
ДОМ  
ТЕРРИТОРИЯ

## Транспорт

ЛИНИЯ  
УЛИЦА  
ЧАСТОК  
СТАНЦИЯ  
МЕТРО  
РАБОТА  
УЛИЦА  
ЧАСТОК  
МОСКОВСКИЙ  
СТАНЦИЯ  
МЕТРО  
РАБОТА  
УЛИЦА  
ЧАСТОК  
МОСКОВСКИЙ  
СТАНЦИЯ  
МЕТРО

## Парки и пешеходные зоны

МЕСТО  
ЗОНА  
ПЛОЩАДЬ  
УЛИЦА  
ПАРК  
ПЛОЩАДКА  
ОТДЫХ  
СТАНЦИЯ  
МЕТРО  
РАБОТА  
УЛИЦА  
ЧАСТОК  
МОСКОВСКИЙ  
СТАНЦИЯ  
МЕТРО  
РАБОТА  
УЛИЦА  
ЧАСТОК  
МОСКОВСКИЙ  
СТАНЦИЯ  
МЕТРО

## Уникальность: Знаковые культур..

РАБОТА  
ВЕК  
ДОМ  
ЗДАНИЕ  
ВРЕМЯ  
РАБОТА  
УЛИЦА  
ИСТОРИЧЕСКИЙ  
ЧАСТЬ  
ФАСАД  
БОЛЬШОЙ  
ПРОЕКТ  
РЕСТАВРАЦИЯ  
ВРЕМЯ  
НАСЛЕДИЕ

## Строительство и реконструкция

ОБЪЕКТ  
ЗДАНИЕ  
УЛИЦА  
СТРОИТЕЛЬСТВО  
НОВЫЙ  
РАБОТА  
ЦЕНТР  
МЕТР  
ПРОЕКТ  
УЧАСТОК

## Городское хозяйство

ВОДА  
НОВЫЙ  
РАБОТА  
УЛИЦА  
ДОМ  
ТЕРРИТОРИЯ

## Образование

ОБРАЗОВАНИЕ  
ШКОЛНИК  
МОСКОВСКИЙ  
ШКОЛА  
ПРОЕКТ  
РЕБЁНОК

## Комфорт: Транспортная инфрастр..

СТАНЦИЯ  
МЕТРО  
ЛИНИЯ  
УЛИЦА  
РАБОТА  
УЧАСТОК  
ПРОЕКТ  
НОВЫЙ

## Качество: Благоустройство

УЛИЦА  
НОВЫЙ  
РАБОТА  
ДОМ  
УЧАСТОК  
ПРОЕКТ  
БОЛЬШОЙ  
ЗДАНИЕ  
ВРЕМЯ  
ПЛОЩАДКА  
ПАРК  
СТАНЦИЯ  
ПОЯВИТЬСЯ

## Культура

БИБЛИОТЕКА  
РЕБЁНОК  
ИСТОРИЯ  
НОВЫЙ МУЗЕЙ  
ДОМ ВРЕМЯ  
РАБОТА  
УЧАСТОК

## Здравоохранение

МЕДИЦИНСКИЙ  
ПОЛИКЛИНИКА  
ПАЦИЕНТ  
ЧЕЛОВЕК  
ЦЕНТР  
ПОМОЩЬ

## Комфорт

ДОМ  
НОВЫЙ  
СТАНЦИЯ  
УЧАСТОК  
УЛИЦА

ТЕАТР  
МУЗЕЙ  
КНИГА ВРЕМЯ  
НОВЫЙ

## Экология

ТЕРРИТОРИЯ  
ЖИВОТОВЕ  
ПТИЦА  
ПДКМ  
ГРОДО  
МЕРОПРИЯТИЕ

## Экономика и предпринимательс..

РАЗВИТИЕ РУБЛЬ  
ГОРОД  
ПРОЕКТ  
МОСКОВСКИЙ  
КОМПАНИЯ  
БИЗНЕС

## Технологии

ПРОЕКТ  
ГОРОД  
ЭЛЕКТРОННЫЙ  
СЕРВИС

ВАКЦИНАЦИЯ  
ПАЦИЕНТ  
ЧЕЛОВЕК  
КОРОНАВИРУС

СТРОИТЕЛЬСТВО  
НОВЫЙ УЛИЦА  
РАЙОН МЕТР  
ДОМ ПЛОЩАДКА  
МЕДИА  
ПРОГРАММА  
РЕНОВАЦИЯ

РЕНОВАЦИЯ ПЛОЩАДКА  
КВАРТИРА СТРОИТЕЛЬСТВО ДОМ  
ПРОГРАММА НОВЫЙ УЛИЦА  
ЖИТЕЛЬЕ РАЙОН

In [54]:

```
df_areas['words_text'] = pd.Series([
    list(chain.from_iterable(df_news.words_text[df_news.territory_area_id==x]))
    for x in df_areas.index
])
```

```
], df_areas.index)
df_areas.head(10)
```

Out[54]:

area_id	district_id	district_title	area_title	words_text
1501	1500	Центральный	Арбат	[НАСЛАДИТЬСЯ, РОМАНС, НИКОЛАЙ, РИМСКИЙ-КОРСАКО...
2501	1500	Центральный	Басманnyй	[СТОЛИЧНЫЙ, БИБЛИОТЕКА, ПОДГОТОВИТЬ, ПРАЗДНИЧН...
3501	1500	Центральный	Замоскворечье	[КАНУН, ДЕНЬ, ВЛЮБИТЬ, СТОЛИЧНЫЙ, ДЕПАРТАМЕНТ,...
4501	1500	Центральный	Красносельский	[МЕДИАПЛАТФОРМА, МОСКВАСТОБОЙ, ЦИФРОВОЙ, ТУРИС...
5501	1500	Центральный	Мещанский	[СЕРЕДИНА, ИЮНЬ, ГОРОД, УКРАСИТЬ, МИЛЛИОН, ОДН...
6501	1500	Центральный	Пресненский	[НОЯБРЬ, ОТКРЫТЬСЯ, СТРАНА, МИР, ПЛАНЕТАРИЙ, С...
7501	1500	Центральный	Таганский	[АРЕНДОВАТЬ, САМОКАТ, ВЕЛОСИПЕД, СКИДКА, ПОДВЕ...
8501	1500	Центральный	Тверской	[НАЗВАТЬ, ПОБЕДИТЕЛЬ, ЕЖЕГОДНЫЙ, КОНКУРС, ХОРО...
9501	1500	Центральный	Хамовники	[НОВЫЙ, ВЫПУСК, ИСТОРИЯ, ВЕШЬ, РАССКАЗАТЬ, ПУТ...
10501	1500	Центральный	Якиманка	[ГРАДОСТРОИТЕЛЬНО-ЗЕМЕЛЬНЫЙ, КОМИССИЯ, ГОРОД, ...

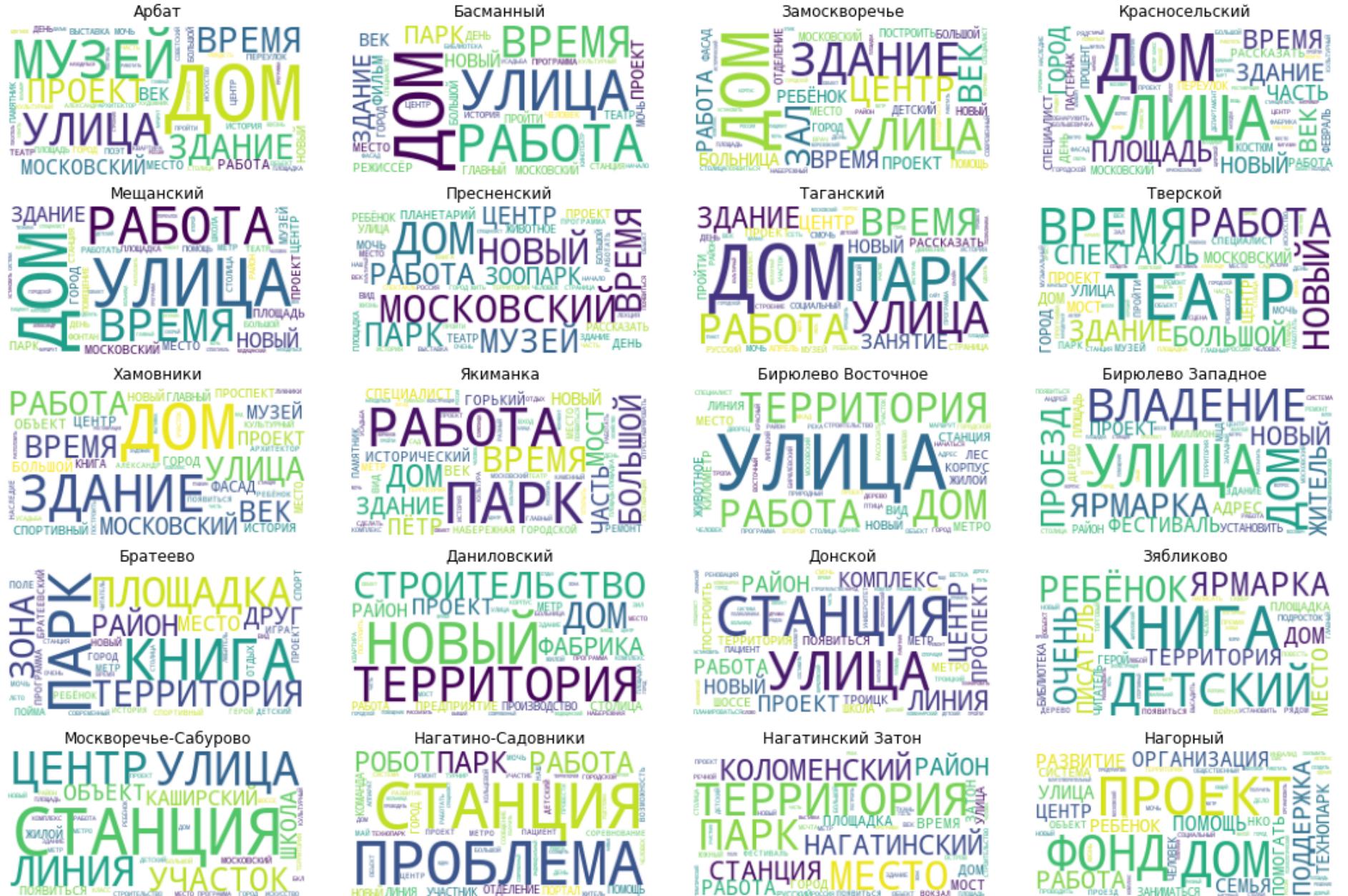
In [55]:

```
fig, axs = plt.subplots(nrows=5, ncols=4, figsize=(15, 10), subplot_kw={'xticks': [], 'yticks': []})

for ax, title, words_text in zip(axs.flat, df_areas.area_title.iloc[:20], df_areas.words_text.iloc[:20]):
    ax.imshow(ava_profile(words_text))
    ax.axis("off")
    ax.set_title(title[:30]+('...' if len(title)>30 else ''))

plt.tight_layout()
plt.show()

# Городские районы
```



In [56]:

# Протестируем работу NER-алгоритма на это новости'

```
txt = bt(df_news.loc[49617073,'full_text'])
print('Новость', df_news.loc[49617073,'title'], '\n\n', txt)
```

Новость Где лучшие химики, физики и математики: названы победители олимпиады «Учитель школы большого города»

27 декабря Департамент образования и науки подвел итоги первой олимпиады «Учитель школы большого города». В профессиональных соревнованиях участвовали более двух тысяч преподавателей математики, физики и химии. Педагоги прошли четыре конкурсных этапа – дистанционный, очный, практический и финальный. По словам самих участников, больше всего им понравился лабораторный практикум, на котором нужно было ставить опыты и экспериментировать.

Главная цель олимпиады «Учитель школы большого города» – поддержка и поощрение педагогов, которые обладают глубокими знаниями по своему предмету.

Как прошел финал олимпиады

В финал олимпиады прошли 19 участников. 21 декабря в Московском центре технологической модернизации образования они выступили перед коллегами с авторскими мастер-классами.

По итогам олимпиады определились семь победителей. Лучшими преподавателями химии стали Владимир Головнёр (школа № 1259) и Евгений Трубицын (школа № 218). Сильнейшие среди математиков – Дмитрий Невидимый из школы № 1530, Игорь Эльман из школы № 218 и Дмитрий Мухин из школы № 179, а среди физиков – Филипп Шапошников из школы № 1553 и Варвара Копьева из школы № 1383. У всех победителей олимпиады есть возможность участвовать в профессиональном конкурсе «Учитель года Москвы».

In [57]:

```
# NER можно применять, работает. Как дополнение к другим инструментам - хорошо.
from natasha import (
    Segmenter,
    MorphVocab,

    NewsEmbedding,
    NewsMorphTagger,
    NewsSyntaxParser,
    NewsNERTagger,

    Doc
)

emb = NewsEmbedding()
segmenter = Segmenter()
morph_vocab = MorphVocab()
ner_tagger = NewsNERTagger(emb)

doc = Doc(txt)
doc.segment(segmenter)
doc.tag_ner(ner_tagger)

for span in doc.spans:
    span.normalize(morph_vocab)
```

```

NER_extract = dict(pd.DataFrame([(x.type, x.normal) for x in doc.spans], columns=['type', 'title']).groupby('type').title.unique())

if 'ORG' in NER_extract:
    print('Найдены организации: ', ', '.join(NER_extract['ORG']))
if 'LOC' in NER_extract:
    print('\nНайдены места: ', ', '.join(NER_extract['LOC']))
if 'PER' in NER_extract:
    print('\nНайдены люди: ', ', '.join(NER_extract['PER']))

```

Найдены организации: Департамент образования и науки, Московском центре

Найдены люди: Владимир Головнер, Евгений Трубицын, Дмитрий Невидимый, Игорь Эльман, Дмитрий Мухин, Филипп Шапошников, Варвара Копьевা

In [ ]:

In [58]:

```

# Посмотрел датасеты в открытых данных
# Датасеты в открытых данных не обновлялись уже давно, решил не использовать
json_datasets = json.loads(DATA_DIR/'datasets.json').read_text(encoding='utf-8')
df_datasets = pd.DataFrame(
    ((item['Id'], item['Caption'], item['LastUpdateDate'], item['IsArchive']) for item in json_datasets['Items']),
    columns=['Id', 'Caption', 'LastUpdateDate', 'IsArchive']
).set_index('Id').rename_axis(None)
df_datasets.LastUpdateDate = pd.to_datetime(df_datasets.LastUpdateDate)
df_datasets[~df_datasets.IsArchive].sort_values('LastUpdateDate', ascending=False).head(40)

```

Out[58]:

		Caption	LastUpdateDate	IsArchive
62381	График закрытия акушерских стационаров медицин...	2021-11-01	False	
1461	Перечень приватизируемых объектов в разрезе эт...	2021-10-25	False	
62701	Ежемесячный рейтинг станций Московского метроп...	2021-09-23	False	
2682	Мобильные прививочные пункты	2021-09-20	False	
62662	Выделенные полосы на улично-дорожной сети	2021-09-16	False	
62681	Дорожные знаки	2021-09-09	False	
62101	Запланированные дорожные ремонтные работы	2021-07-28	False	

	Caption	LastUpdateDate	IsArchive
<b>62603</b>	Карта улиц с односторонним движением в городе ...	2021-07-22	False
<b>62601</b>	Карта улиц с уменьшенной скоростью движения «Б...	2021-07-22	False
<b>62059</b>	Светофорные объекты на улично-дорожной сети	2021-07-22	False
<b>62581</b>	Работы по капитальному ремонту и благоустройст...	2021-07-20	False
<b>62207</b>	Входы и выходы станций Московских центральных ...	2021-07-14	False
<b>62201</b>	Железнодорожные вокзалы Москвы	2021-07-14	False
<b>62441</b>	Текущие (локальные) ремонтные работы, проводим...	2021-06-04	False
<b>918</b>	Станции проката велосипедов	2021-06-04	False
<b>62421</b>	Московский реестр участников и неучастников бю...	2021-06-03	False
<b>62541</b>	Карта «школьных зон» города Москвы	2021-05-24	False
<b>62525</b>	Карта среднемесячной загруженности дорог с инд...	2021-05-13	False
<b>62523</b>	Годовой пассажиропоток по всем видам обществен...	2021-05-13	False
<b>62521</b>	Месячный пассажиропоток по всем видам обществен...	2021-05-13	False
<b>62747</b>	Табло отображения информации на улично-дорожнно...	2021-05-10	False
<b>62745</b>	Точность выполнения расписания на линиях Моско...	2021-05-10	False
<b>62743</b>	Пассажиропоток по станциям Московского метропо...	2021-05-10	False
<b>62741</b>	Информация о количестве станций, вагонов и про...	2021-05-10	False
<b>744</b>	Приюты для бродячих животных	2021-05-05	False
<b>2317</b>	Перечень дорожных специализированных предприят...	2021-05-04	False
<b>62503</b>	Реабилитационные и Реабилитационно-образовател...	2021-04-30	False
<b>62501</b>	Данные о действующих зарегистрированных уведом...	2021-04-29	False
<b>916</b>	Велосипедные парковки	2021-04-27	False
<b>2855</b>	Данные о действующих ордерах на производство р...	2021-04-27	False
<b>62481</b>	Туристский событийный календарь города Москвы	2021-04-23	False

	Caption	LastUpdateDate	IsArchive
3295	Мероприятия по установке подъемных платформ дл...	2021-04-06	False
1343	Справочник органов исполнительной власти город...	2021-03-26	False
897	Дорожки велосипедные	2021-03-25	False
1046	Ответственные балансодержатели межрельсового п...	2021-03-25	False
624	Входы и выходы вестибюлей станций Московского ...	2021-03-09	False
1982	Динамика изменения величины прожиточного миним...	2021-02-09	False
62461	Данные о действующих зарегистрированных уведом...	2021-02-08	False
620	Ярмарки выходного дня	2021-02-04	False
62409	Ежегодное количество культурных мероприятий в ...	2021-01-21	False