

PANORAMA BOT

Ася Линич, Тарас Андрушко, Нелля Джубаева

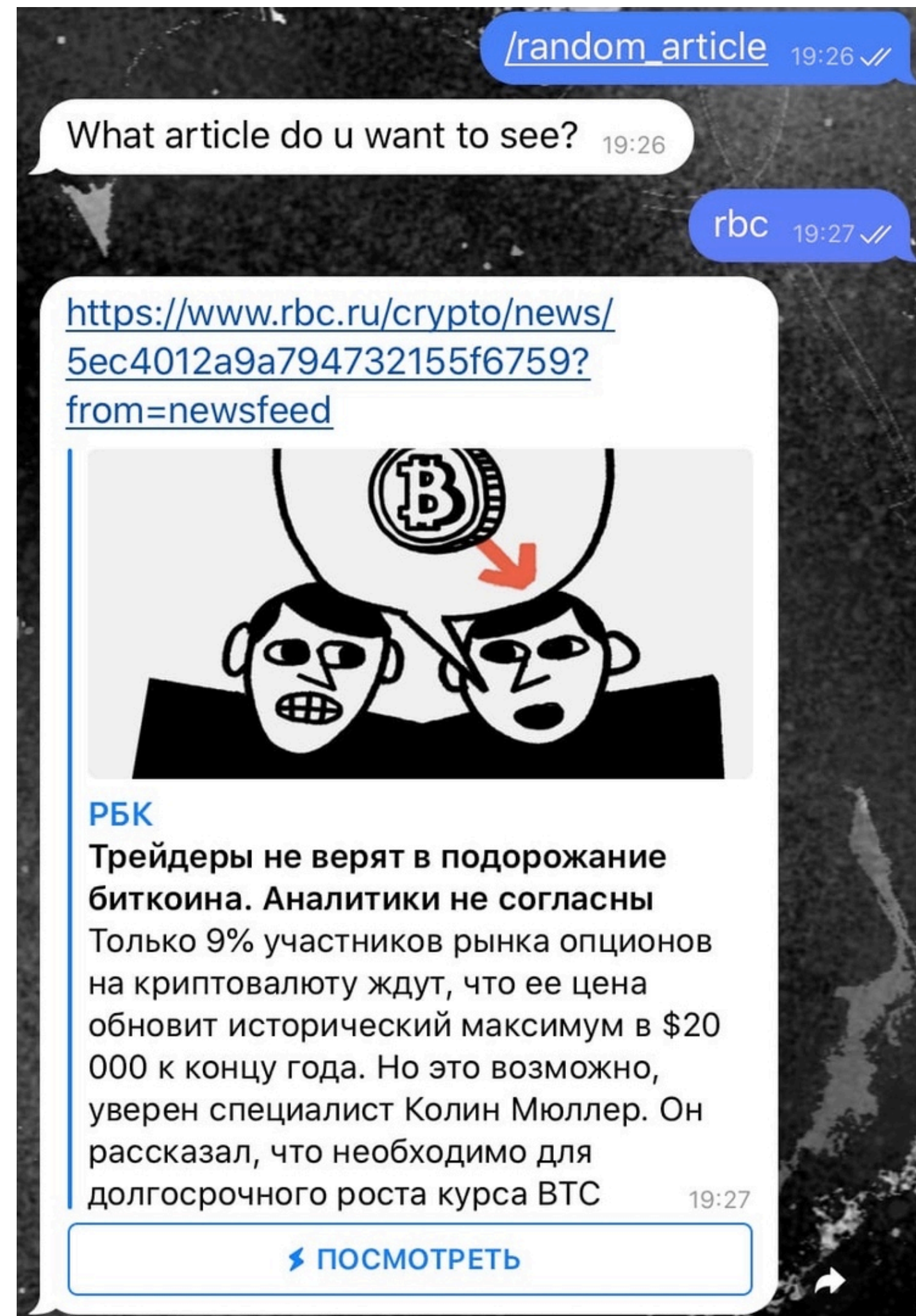
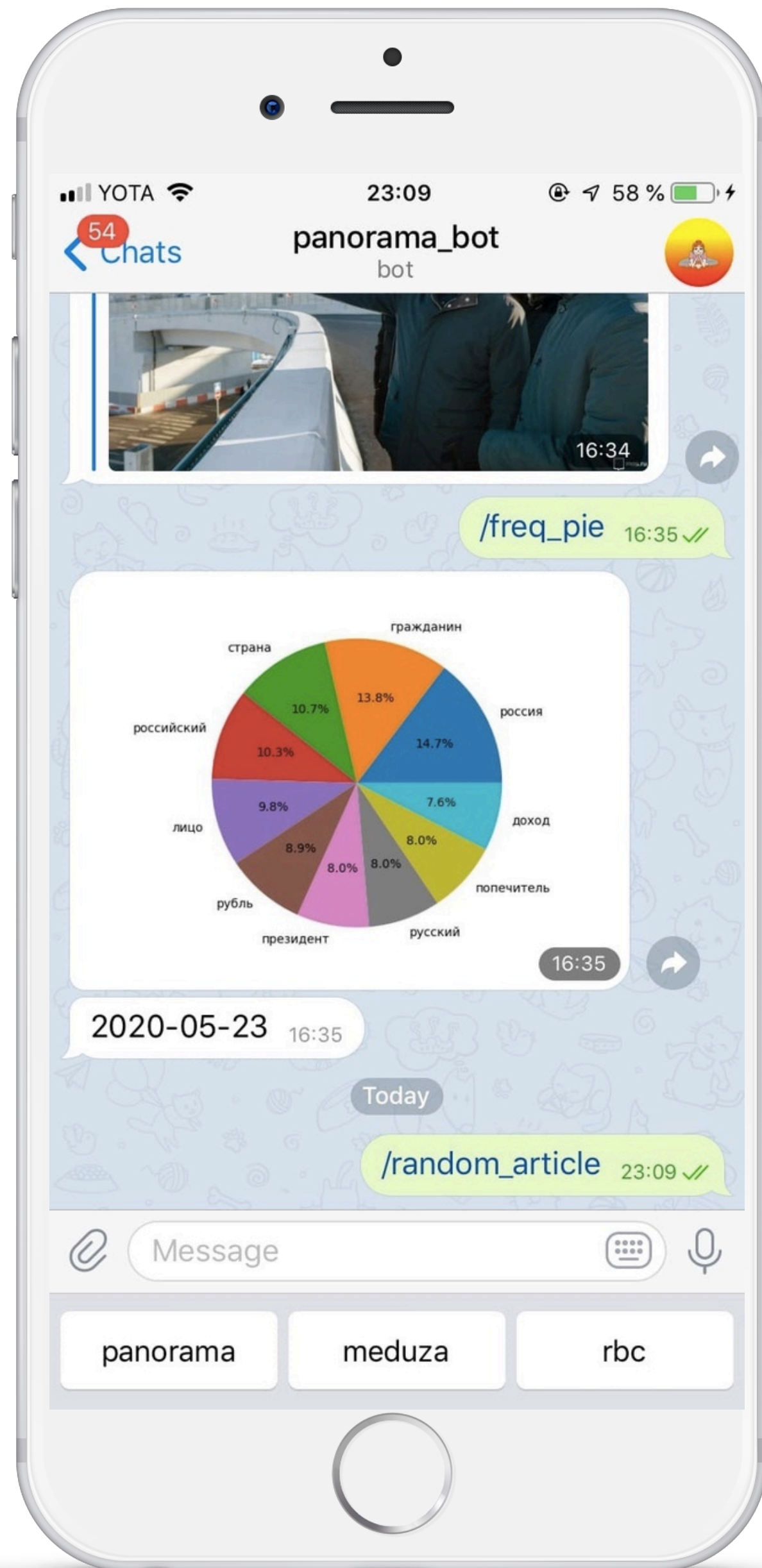
Что умеет бот?

Выдавать рандомные статьи

У нас есть парсер сайтов медузы и рбк, который выдает рандомные статьи с главных страниц, а также штука, которая генерирует рандомные статьи на панораме.

Строить графики

Ася написала штуку, которая делает диаграммы частотности слов.



КОМАНДЫ БОТА



/random_article

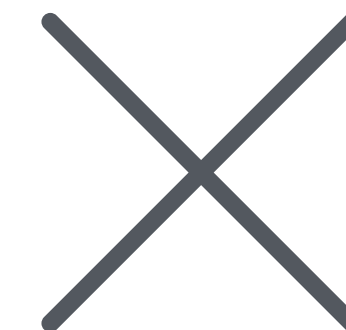
отправляет randomную статью, есть кнопочки для выбора ресурса, из которого нужна статья



/freq_pie

делает классные графики!

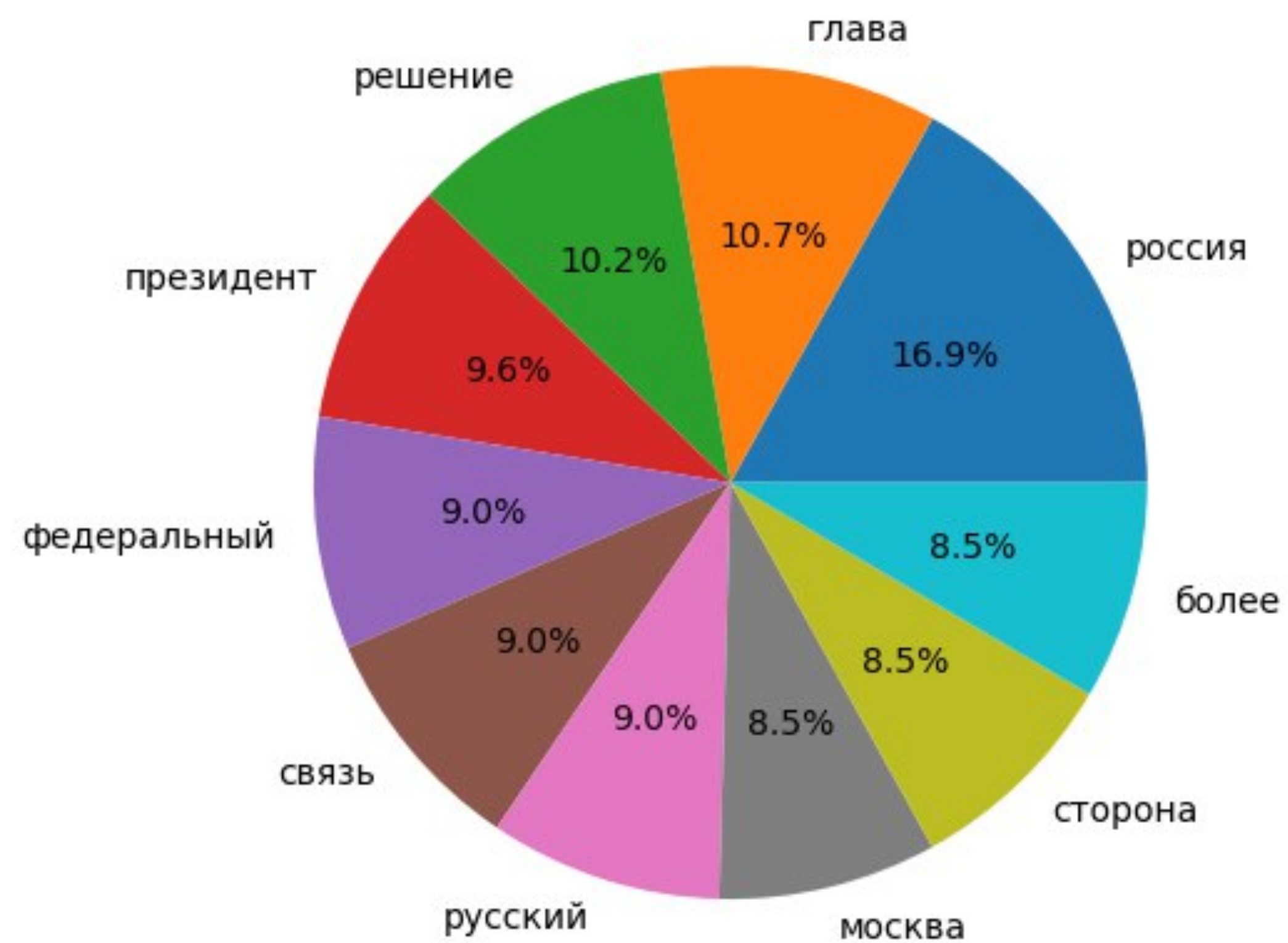
ГРАФИКИ



```
def normal(item):
    lemma = ''
    for i in item:
        if (i >= "а" and i <= "я") or (i >= "А" and i <= "Я") or i == "ё" or i == "Ё" or (i == '-' and lemma != ''):
            lemma += i
    lemma_info = Morph_Anal.parse(lemma)[0]
    lemma = lemma_info.normal_form
    if lemma not in freq_list and (lemma_info.tag.POS not in ['NPRO', 'PREP', 'CONJ', 'PRCL', 'INTJ', 'COMP', None]):
        return lemma
```

```
def freq(text):
    all_words = {}
    for item in text:
        item = normal(item)
        if item != None and item in all_words.keys():
            all_words[item] += 1
        elif item != None:
            all_words[item] = 1
    n = 10 #number of pieces of pie
    maximum = [['', 0]] * n
    items = all_words.keys()
    for key in items:
        for i in range(n):
            if all_words[key] > maximum[i][1]:
                rshift = maximum[i]
                for j in range(i + 1, n):
                    maximum[j], rshift = rshift, maximum[j]
                maximum[i] = [key, all_words[key]]
                break
    return maximum
```

Для их создания использовалась библиотека matplotlib вместе с rymorphy2. Функция построения графика запускается, когда пользователь вводит /freq_pie



Вот так выглядит
график
Он очень красивый

RBC & MEDUZA Парсеры

Что использовалось?

requests, bs4, random (takeRandomElement), typing

```
def get_rbc_articles(url='https://tv.rbc.ru/'): #Returns list of url-articles from rbc main page
    soup = __get_page_soup(url)
    articles = soup.find_all('a', id=re.compile('id_newsfeed'))
    return list(map(lambda page: page['href'],
                    filter(lambda page: 'rbc' in page['href'],
                           articles)))
```

```
def get_meduza_articles(url="https://meduza.io/api/w5/screens/news"): #Returns list of url-articles from meduza API
    content = __get_page_content(url)
    articles = re.findall('{"key": "(.*?)"', content)

    def contains_any(url): #Returns true if url contains any element of allowed
        allowed = ['shapito', 'cards', 'news', 'feature', 'episodes']
        return bool(sum(list(
            map(lambda name: name in url,
                allowed))))

    return list(map(lambda url: "https://meduza.io/" + url,
                    filter(contains_any,
                           articles)))
```

[код]



https://github.com/letitself/project_newsbot