

Розробка агрегатора новин з різних ресурсів для месенджера Telegram



Студент: Соломко Р.Т.

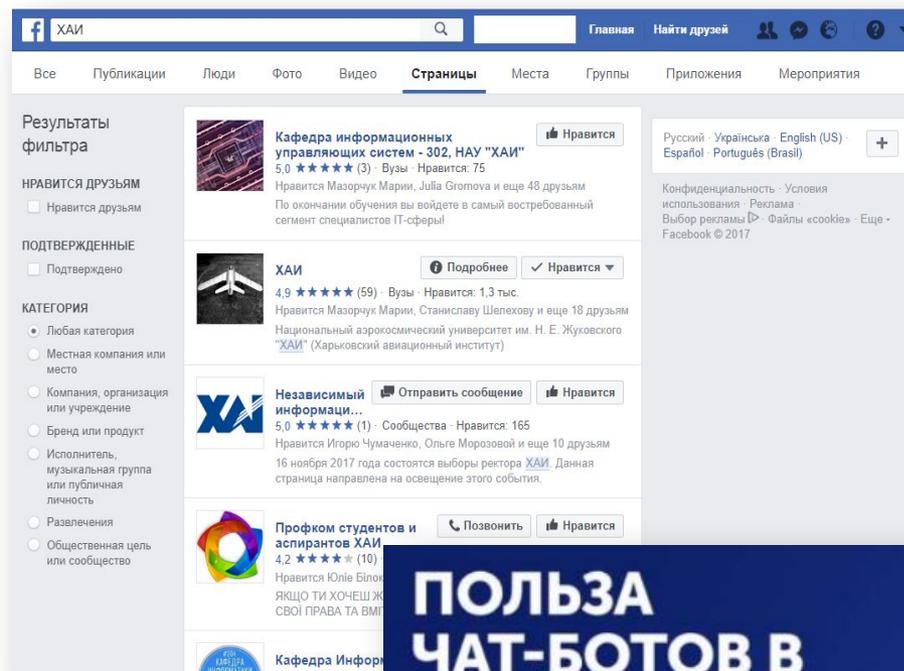
Научный руководитель: Миланов М.В.

Новостной чат-бот для ХАИ

Результаты поиска «ХАИ» в сети Facebook

- Это сервис, агрегирующий информацию из различных ресурсов:
 - новостных групп в социальных сетях ВКонтакте и Facebook,
 - освещающим новостные поводы, относящиеся к Национальному аэрокосмическому университету им. Н.Е. Жуковского ХАИ,
 - а также с официального веб-сайта университета,

производящего контекстный анализ сообщений, их категоризацию и предоставляющего его пользователям персонализированные рассылки сообщений как в реальном времени (в момент публикации на том или ином ресурсе) и (или) же в качестве подборки лучших один раз в день



ПОЛЬЗА ЧАТ-БОТОВ В МЕССЕНДЖЕРАХ

- Одно приложение на все случаи жизни;
- Нужная информация в один клик;
- Оптимизация процесса работы;
- Интерактивность и скорость работы.

Цель и задачи исследования

Цель: разработка сервиса, который выполняет функцию агрегирования новостей о ХАИ из разных ресурсов и выполняет их рассылку пользователям

Задачи:

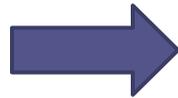
- анализ средств разработки,
- разработка механизмов сбора новостей из социальных сетей и RSS-каналов,
- разработка алгоритмов подсчета популярности и агрегации лучших сообщений,
- разработка структуры базы данных,
- развертывание проекта на сервере

Объект: новостной чат-бот

Предмет: агрегирования новостей о ХАИ из разных ресурсов для новостного чат-бота в мессенджере Telegram

Telegram Bot API

СОЗДАНИЕ НОВОГО БОТА



Команды для @BotFather

- /setname
- /setdescription
- /setabouttext
- /setuserpic
- /setcommands
- /setjoingroups
- /setprivacy
- /deletebot

МЕХАНИКА ВЗАИМОДЕЙСТВИЯ С БОТАМИ



- Запрос в URL
- application/x-www-form-urlencoded
- application/json (не подходит для загрузки файлов)
- multipart/form-data (для загрузки файлов)



Все запросы к API Телеграммы Bot должны быть поданы через HTTPS и должны быть представлены в следующем виде:

```
https://api.telegram.org/bot<token>/METHOD_NAME
```

Например:

```
https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v  
1u123ew11/getMe
```

Выбор средств разработки

ЯЗЫКИ РАЗРАБОТКИ БОТОВ

JavaScript

Python

PHP

C#

Rust

Ruby

БИБЛИОТЕКИ ДЛЯ СОЗДАНИЯ TELEGRAM БОТОВ

• pyTelegramBotAPI

• Telegram-bot-api

• Telegame

• Telepot

• pytgbot

• tgclient

• wspan.telegram

• tapioca-telegram

• Twx, Twx.botapi, Twx-botapi

• python-telegram-handler

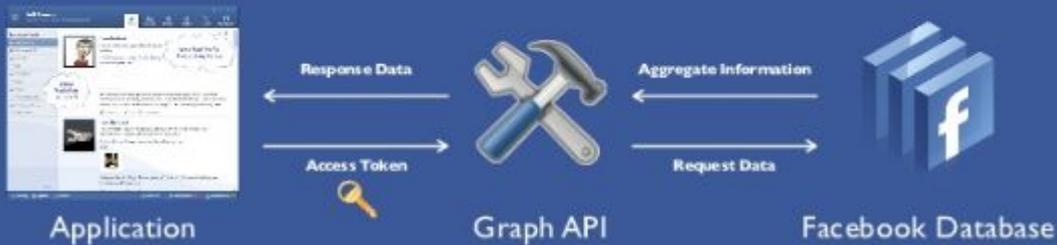
API ВКонтакте

- **API ВКонтакте** – это интерфейс, позволяющий получать информацию из базы данных сайта vk.com при помощи HTTP-запросов к серверу
- Для работы бота достаточно запросов HTTP **GET** (**не нужно получать специальный ключ API**, чтобы совершать запросы, направленные на получение общедоступной информации)
- **Запросы** к API ВКонтакте выглядят следующим образом:
`https://api.vk.com/method/METHOD_NAME?SOME_PARAMETERS`,
где `METHOD_NAME` – это название метода в API ВКонтакте, а `SOME_PARAMETERS` – это некоторые параметры запроса, например, версия API ВКонтакте или `id` пользователя.
- API ВКонтакте возвращает в **ответ JSON-объект**, либо с запрошенными данными, либо с сообщением об ошибке.
- Для облегчения работы с API ВКонтакте будет использован **Python-модуль vk**.

```
session = vk.Session() // сначала необходимо создать новую сессию,  
api = vk.API(session, v='5.63', timeout=10) // затем надо подключиться к API ВКонтакте, где v – это версия  
API ВКонтакте, а timeout – это время ожидания ответа от сервера.  
api.method.name(parameter=value) // обращение к методам API ВКонтакте, где method.name – это метод, к  
которому идет обращение, parameter – это название параметра этого метода (может быть несколько), а value – это  
значение параметра
```

API Facebook

Graph API



- RESTful HTTP request & response
- Response data-type: JSON
- Access URL: graph.facebook.com

```
import facebook
```

```
graph = facebook.GraphAPI()  
user = graph.get_object("littleq0903")
```

```
my_profile = graph.get_object("me")
```

```
graph = GraphAPI(ACCESS_TOKEN)
```

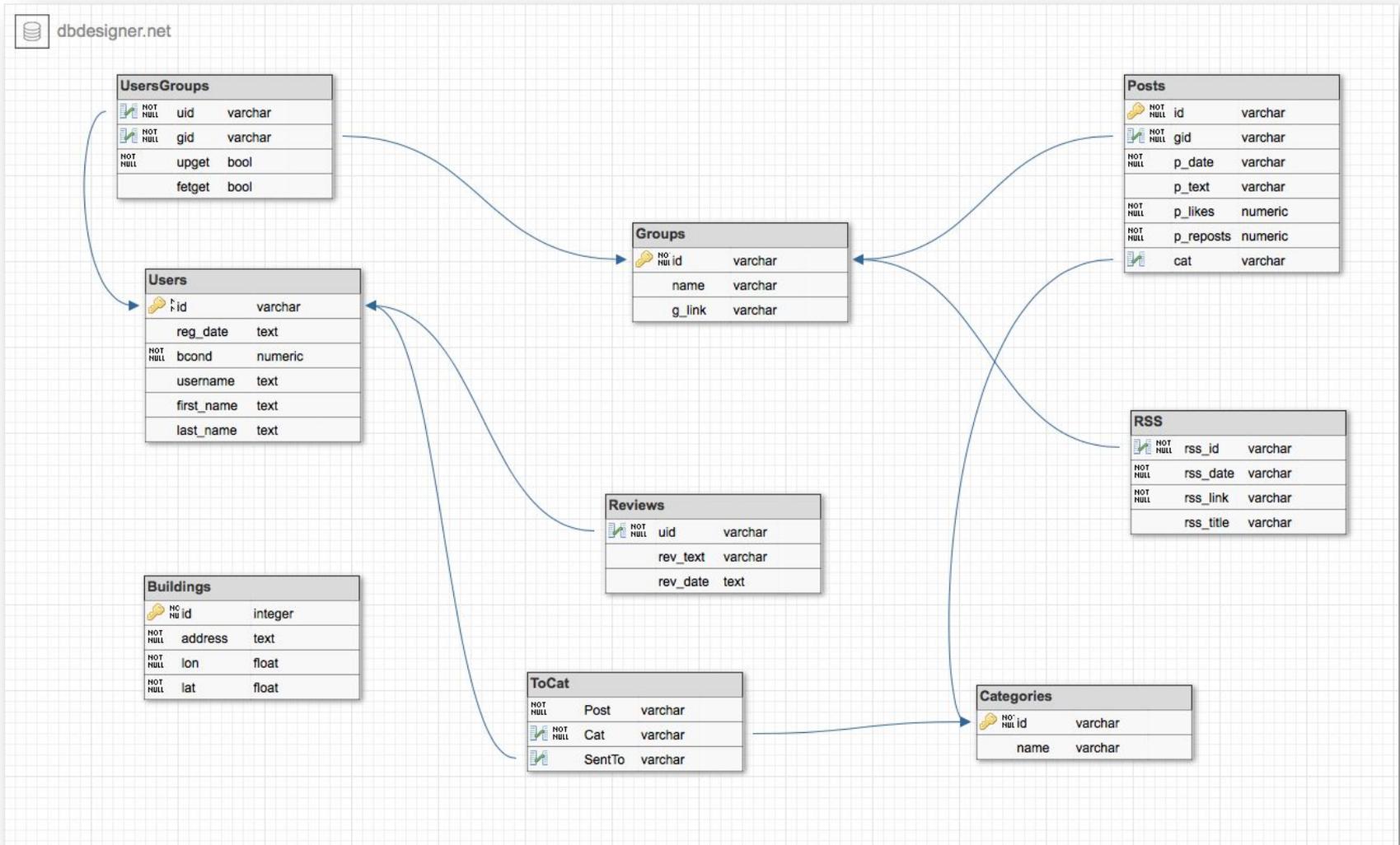
Python SDK

facebook  python™

Graph Methods

- get_object, get_objects
- get_connections
- put_object
- put_wall_post, put_comment, put_like
- delete_object

Структура базы данных



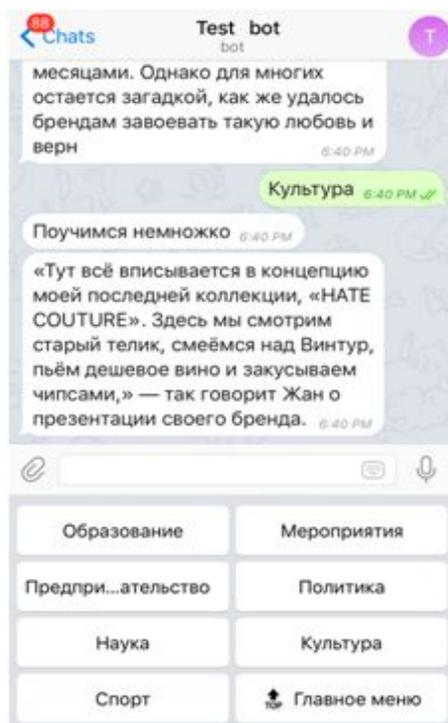
Категоризация сообщений



machine learning in Python

Категоризация текста получаемых сообщений реализована с использованием библиотеки scikit-learn для Python.

Библиотека использует классификаторы, регрессионные модели и кластеры, в том числе метод опорных векторов (Метод опорных векторов, n.d.), «случайный лес» (Random forest, б.д.), градиентный ускоритель (Gradient boosting, б.д.), метод k-средних (k-means, б.д.), DBSCAN (DBSCAN, б.д.) и множество других



Интерфейс категоризации текстов

Для ручной оценки сообщений был разработан дополнительный интерфейс для бота, вызываемый командой **/learn** и позволяющий выбрать категорию при помощи виртуальных клавиш, причем данная команда доступна только пользователям, включенным в список администраторов в файле конфигурации

Все тексты внутри системы переводятся на английский язык с помощью расширения Python под названием **translate**, которое осуществляет перевод текстов через сервис Google Translate