**Техническая документация**

(beta – версия)

Дата последнего изменения: 06.07.2023

**Чат-бот в Вконтакте**

Оглавление

[Слова автора документации 3](#_Toc139529130)

[Глава 1. Общее 3](#_Toc139529131)

[О программе 3](#_Toc139529132)

[Сценарии 6](#_Toc139529133)

[Глава 2. Техническое 8](#_Toc139529134)

[Как запустить программу 8](#_Toc139529135)

[Используемые библиотека, которые надо скачивать 8](#_Toc139529136)

[Главный файл 8](#_Toc139529137)

[Функция checking\_working\_time 9](#_Toc139529138)

[Функция handle\_events 9](#_Toc139529139)

[Функция classify\_mera\_and\_tema 10](#_Toc139529140)

[Функция kakoi\_filifl 11](#_Toc139529141)

[Функция kakai\_mera 11](#_Toc139529142)

[Работа сценариев 12](#_Toc139529143)

[Функция run\_keyboard 12](#_Toc139529144)

[Функция write\_msg 12](#_Toc139529145)

[Функция close\_scenario 12](#_Toc139529146)

[Функция check\_script\_odobrenie 12](#_Toc139529147)

[Особенность сценария «Что мне положено» 13](#_Toc139529148)

[Глава 3. Часто задаваемые вопросы 14](#_Toc139529149)

[Глава 4. Рекомендации/планы по улучшению 15](#_Toc139529150)

[Список литературы 16](#_Toc139529151)

# Слова автора документации

Это моя первая в жизни документация. Примеров я не нашла. Если она покажется вам плохой (а скорее вам не покажется) – то лучше я просто не могла.

# Глава 1. Общее

## О программе

Данная программа является чат-ботом и работает для группы <https://vk.com/cspmsrorb>.

Технические характеристики:

* Язык программирования – Python
* Модель ИИ – Catboost

В современном мире, когда у каждого телефон под рукой, коммуникации между людьми становиться все больше. Это касается не только личного общения, то и остальных форм, такие как бизнес-коммуникации. Чаще всего именно когда у нас что-то перестает работать мы идет и пишем компании-представителю услуг. Будь то жалоба или просто просьба о помощи. А может человек хочет только начать пользоваться услугами данной компании и ему нужна консультация.

Каким бы не был запрос человека, он такой не один. И вопрос то же не один. Ранее компании нанимали людей для кол-центров, куда люди звонили и задавали свои вопросы. Плавно голосового оператора сместил, но не вытеснил, оператор в чате. Теперь же на помощь всем операторам приходят боты.

Бот – это программа-собеседник, которая общается с клиентом на сайте, в приложении или мессенджере, а также на телефонной линии. Цель чат-бота – ответить на вопросы клиента. Он общается в форме диалога на понятном клиенту языке. В зависимости от способностей бота их относят к разным сферам. Мы же будем рассматривать ботов, обрабатывающие исключительно текстовые сообщения – чат-ботов.

В настоящее время чат-боты, как современные инструменты коммуникаций, стали широко использоваться во многих сферах жизнедеятельности человека. Наибольшую популярность чат-боты получили, когда началось их использование в мессенджерах и социальных сетях (к примеру, в Telegram, Viber, Facebook, ВКонтакте).

Понятие «чат-бот» произошло от двух английских слов: to chat – непринужденный разговор в сети Интернет, bot (robot) – сокращенно робот, из чего следует, что это роботы, предназначенные для осуществления коммуникаций с пользователями в сети Интернет, выполняющие действия согласно заложенному сценарию.

Такая программа интернет-коммуникации как чат-бот основана на современных технологиях, в ее основе лежат такие информационно-коммуникационные технологии как искусственный интеллект, нейронные сети, что позволяет приближать коммуникации чат-ботов к человеческому виртуальному общению и решать с их помощью коммерческие и маркетинговые задачи.

В силу того, что контакт с чат-ботами является не только интересным занятие, но и полезным для человека, с точки зрения получения новой информации и решения потребительских задач, стоит отметить, что интернет-аудитория приняла их появление доброжелательно. Многие организации в свою очередь углядели в интересе аудитории к чат-ботам возможности для собственного продвижения и стали разрабатывать собственные чат-боты, используя их для привлечения внимания потребителей к компании за счет использования современного инструмента коммуникации.

Сейчас даже государственные организации обратили свое внимание на такую технологию, как чат-бот. Одна из таких организаций, о которой в дальнейшем пойдет речь, Государственное казенное учреждение Оренбургской области «Центр социальной поддержки населения» (ГКУ «ЦСПН»).

В данную организацию мною разрабатывается чат-бот для социальной сети Вконтакте. Данная необходимость возникла с большим потоком однотипных вопросов и отсутствием возможности найма дополнительных людей на роль оператора. Да и дополнительная сила в виде человеческой тут не поможет. Нужно более современное автоматизированное решение.

Таким решением является создание чат-бота. На рисунках 3 и 4 изображены модели общения в стартовой точке и желаемой конечной точке.

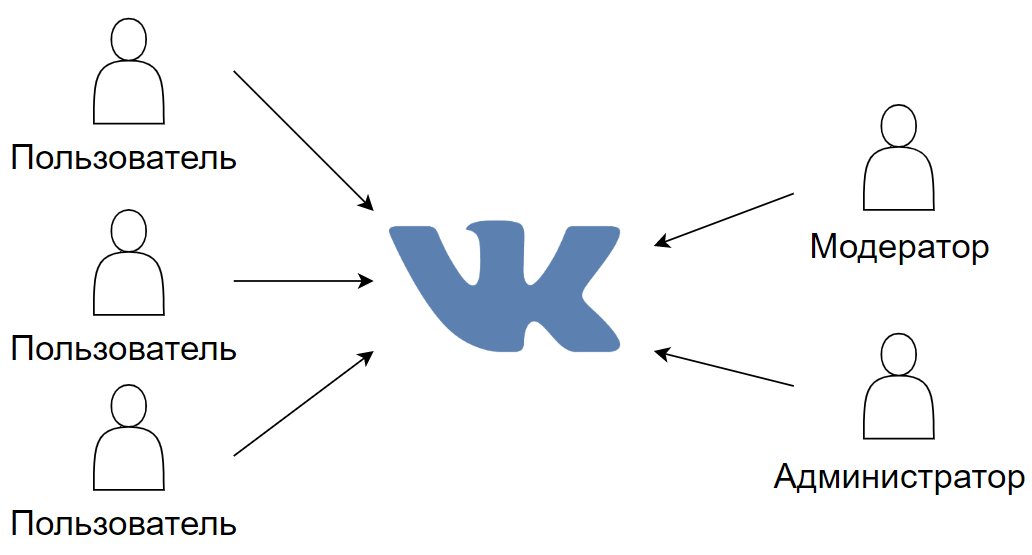


Рисунок 3 – Модель общения: стартовая точка

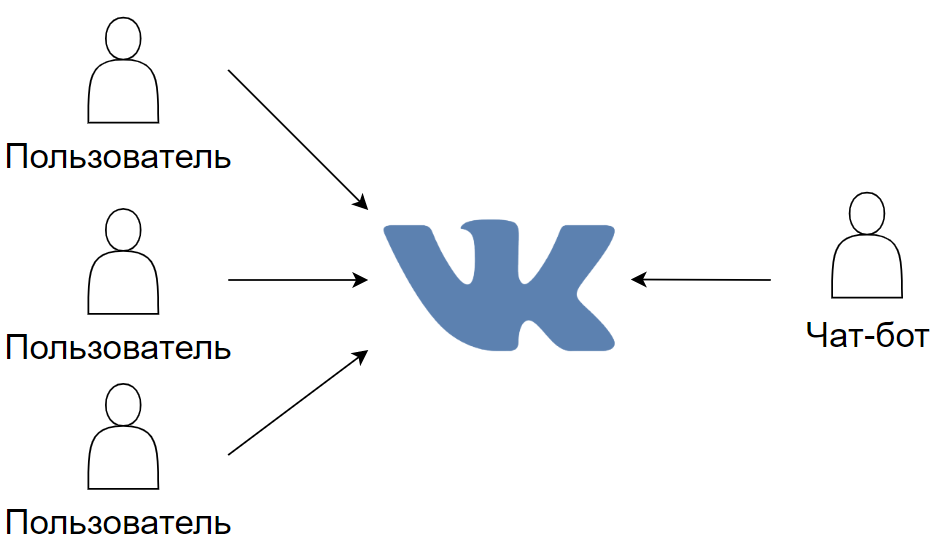


Рисунок 4 – Модель общения: конечная точка

Как это будет работать?

Боту необходимо будет обрабатывать так называемые события, которые он будет получать от Вконтакте через специальную технологию – API– открытое взаимодействие с сервером Вконтакте. Для доступа мы получаем специальный ключ – токен. Именно по нему мы подключаемся к Вконтакте и получаем события.

Событием может быть любое действие пользователя:

* личное сообщение;
* комментарий под постом на стене;
* отписка от сообщества и т.д.

Наш бот будет обрабатывать одно конкретное событие – личное сообщение пользователя в сообщение группы. Т.е. наш чат-бот будет работать как сотрудник в сообществе (группы). На рисунке 5 приведены «обязанности», этапы работы чат-бота.

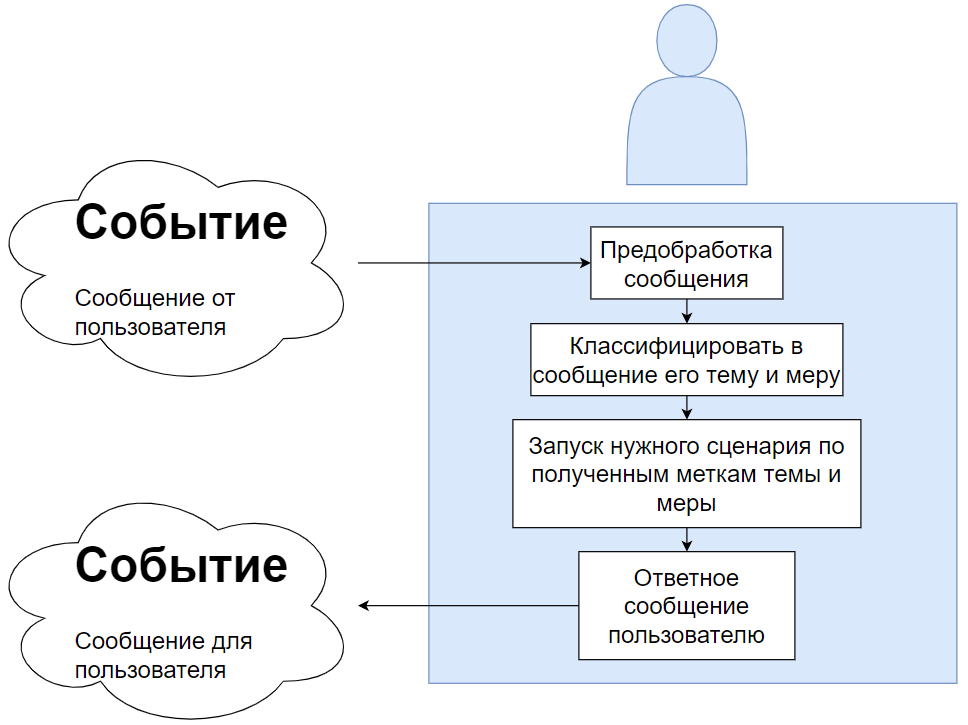


Рисунок 5 – Обязанности чат-бота

Данная работа будет рассматривать только 1 этап – классификацию сообщения по теме и мере.

Тема – то, о чем идет речь в сообщении.

Пример. Сообщение: «Добрый день. Подскажите, какие документы необходимо предоставить для получения субсидии?»

Тема: какие документы.

Больше примеров в таблице 1.

Мера – то, о какой мере идет речь в сообщении.

Пример. Сообщение: «Добрый день. Подскажите, какие документы необходимо предоставить для получения субсидии?»

Мера: Субсидия (на самом деле она называется по-иному, но всем привычно называть её именно так).

Больше примеров в таблице 2.

В общем виде задача классификация по теме и мере делиться на 2 независимые классификации соответственно. В обоих случаях, это будет – многоклассовая классификация. Т.к. чат-боту необходимо соотнести полученное сообщение с 1 классом из всех представленных ему классов.

Таблица 1 – Варианты классов «Тема»

|  |  |
| --- | --- |
| Благодарность | Когда там спасибо и т.д. |
| Какие документы | Если нужно назвать перечень документов |
| Когда выплата | Из разряда «сегодня будет выплата?» |
| Номер телефона | Надо будет прислать номер телефона |
| Продлить | Спрашивают о продлении меры |

Таблица 2 – Варианты классов «Мера»

|  |  |
| --- | --- |
| ЕДВ 3-7 | Ежемесячная денежная выплата на детей в возрасте с 3 до 7 |
| Уточнить | В сообщении не понятно, о какой мере идет речь. Стоит уточнить |
| ЕВ при рождении | Единовременная выплата при рождении/усыновлении |
| Не наша МСП | В сообщение идет речь о мере, которая выплачивается не ГКУ «ЦСПН» |

## Сценарии

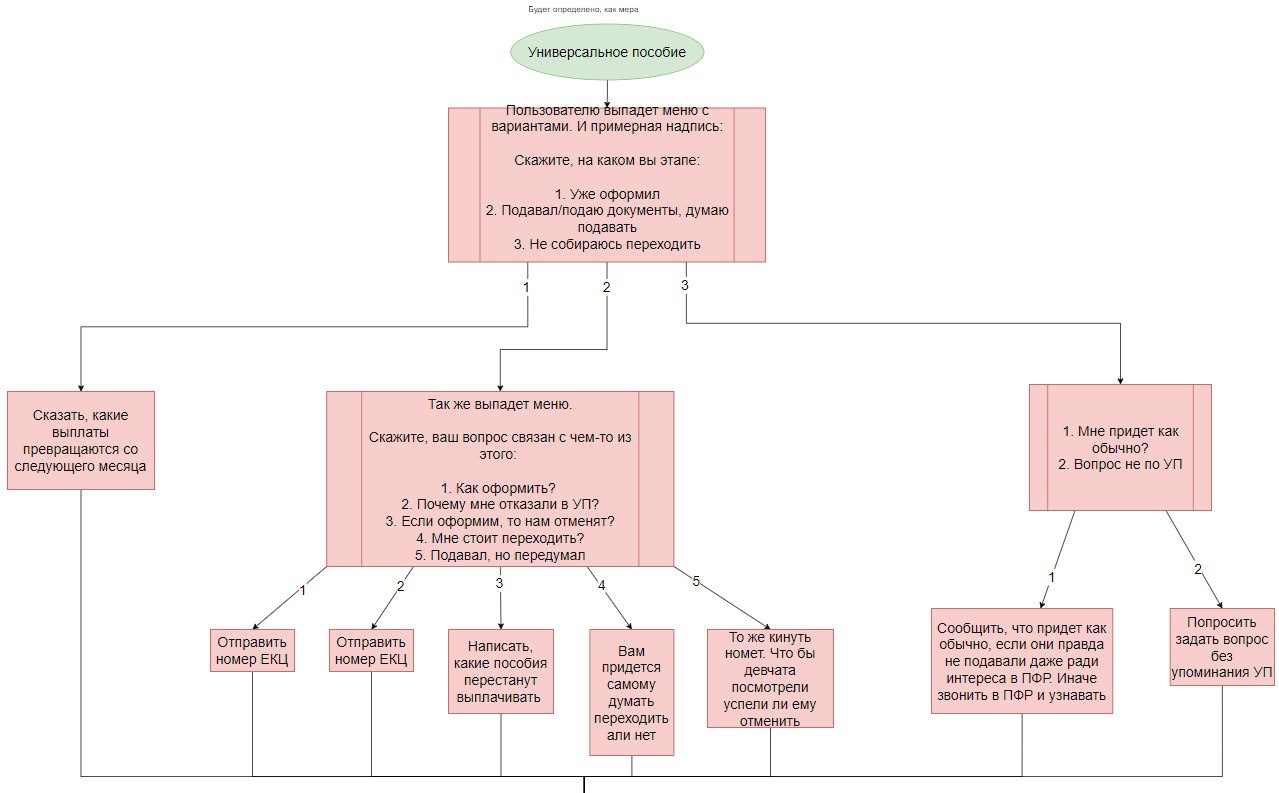
Для работы бота одной обученной модели недостаточно – должен быть какой-то сценарий его работы.

Наш бот будет работать в зависимости от пары «Тема» и «Мера», которые ему выдадут модели. На данный момент с помощью моделей определяется тема, а мера простым перебором по нахождению ключевых слов с помощью словаря. Но в конечном итоге мы также получаем пару «Тема» и «Мера».

Всего будет 4 вида сценариев:

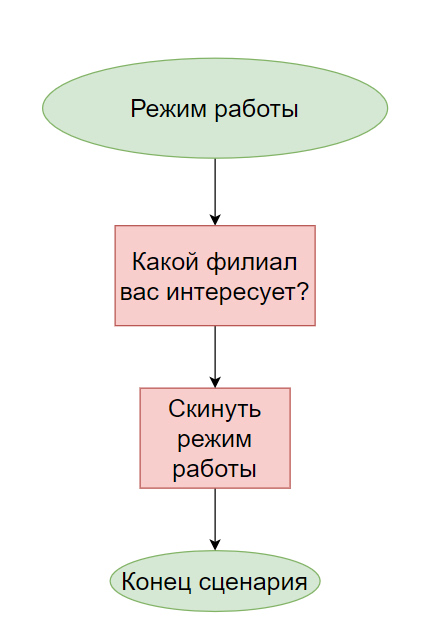
* + зависит и от темы, и от меры;
  + зависит только от темы;
  + зависит только от меры;
  + вспомогательный сценарий.

Те, что зависят только от меры, всего 1 – «Универсальное пособие» (рисунок 19) (сценарии названы по тем темам, по которым они запускаются). Данная мера не является представляемой от данной организации, но все же есть вопросы, саванные с ГКУ «ЦСПН». Поэтому просто отнести данную меру к «чужим» мы не можем.



Блок-схема сценария «Универсальное пособие»

Самый простой вариант сценария – не зависящий от меры. Примерам такого является «Режим работы». Данный сценарий запускается, если человек спрашивает, как работает организация или её филиал. Блок-схема сценария представлена на рисунке 20.



Блок-схема сценария «Режим работы»

Сложными сценарии считаются, которые зависят от меры.

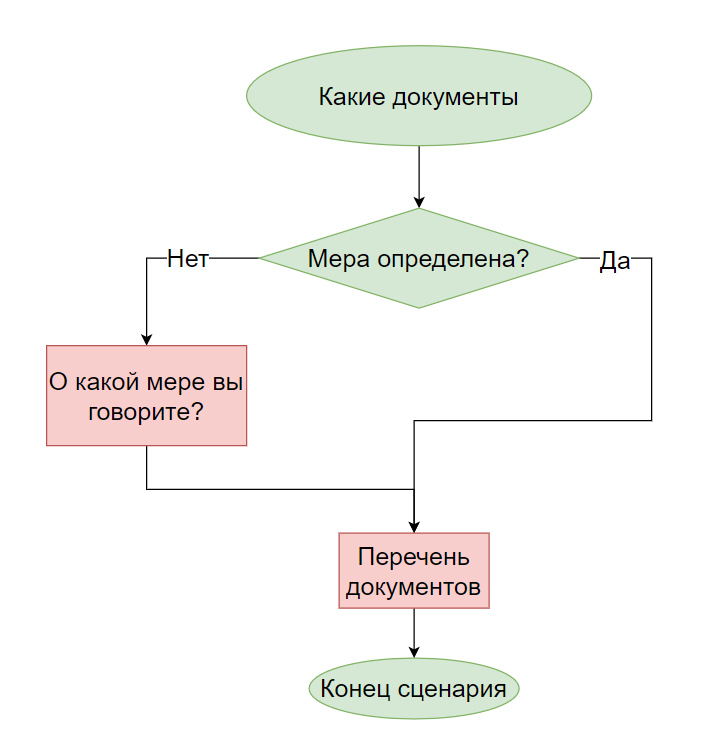
Есть два развития события перед их запуском:

* Нам удалось определить меру. Можем спокойно запустить сценарий;
* Мы не сумели определить меру.

В первом случае нет никаких сложностей. Как определилось – так и запускаем сценарий.

Во втором случае нам необходимо как-то узнать, о какой мере нас спрашивает пользователь. Именно для этих целей и есть вспомогательный сценарий «Нет меры». При его запуске мы выясняем, о какой мере идет речь. И когда нам это удается понять, запускаем нужный сценарий уже со знанием меры.

На рисунке 21 продемонстрирована блок-схема сценария «Какие документы», который зависит от меры.



Блок-схема сценария «Режим работы»

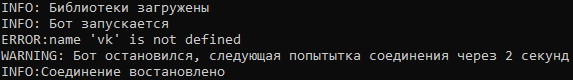
# Глава 2. Техническое

## Как запустить программу

В файлах программы есть файл start.bat. Достаточно кликнуть по нему 2 раза, и он запустит бота, открыв командную строку.

При запуске будет предупреждение о загрузке библиотек (это просто подключение всех файлов для работы, а не буквальная загрузка). После чего сообщение о запуске бота и моментальная ошибка. Все так и надо! В следующих пунктах будет подробнее описан алгоритм работы.

После обработки первой ошибки он восстановит сообщение и начнет работу в обычном режиме.



INFO: Библиотеки загружены

INFO: Бот запускается

ERROR:name 'vk' is not defined

WARNING: Бот остановился, следующая попытытка соединения через 2 секунд

INFO:Соединение востановлено

## Используемые библиотека, которые надо скачивать

python 3.10.x (на 05.07.2023 с версией выше не работает pymorphy2. Т.е. версия 3.11 не подойдет!)

П.с. на версии 3.10.11 точно все работало.

Для работы использовались следующие библиотеки:

|  |  |  |
| --- | --- | --- |
| Название библиотеки | Установить (простая установка и вариант конкретной версии) | Версия не ниже |
| vk-api | pip install vk-api  pip install vk-api==11.9.9 | 11.9.9 |
| catboost | pip install catboost  pip install catboost==1.1.1 | 1.1.1 |
| pymorphy2 | pip install pymorphy2  pip install pymorphy2==0.9.1 | 0.9.1 |
| scikit-learn | pip install scikit-learn  pip install scikit-learn==1.1.3 | 1.1.3 |

## Главный файл

Файл start.bat запускает файл bot\_catboost.py, который является главным файлом программы. Вся программа работает в цикле (для наглядности мною были убраны строки логирования, т.к. на функциональность это не влияет)

while True:

try:

now\_time\_1 = vk.method('utils.getServerTime', {})

working\_time = checking\_working\_time(now\_time\_1)

if now\_time\_1 - now\_time > 600:

now\_time = now\_time\_1

user\_dict, id\_del = chek\_time\_mess(user\_dict, now\_time, vk)

for id\_d in id\_del:

pass

events = longpoll.check()

if len(events)>0:

for event in events:

handle\_events(vk, working\_time, event, user\_dict )

time.sleep(0.01)

except KeyboardInterrupt as err:

break

except Exception as err:

t = True

while t:

time.sleep(2)

try:

vk, longpoll, now\_time = start\_bot()

except Exception as err:

else:

t = False

now\_time = vk.method('utils.getServerTime', {})

Переменные now\_time\_1 и now\_time хранят серверное время. Разница в том, что что now\_time обновляется раз в 10 минут, а now\_time\_1 каждый раз, когда к нам приходить какое-то событие. Данное время (10 минут) является оптимальным временем хранения информации о пользователях в памяти и так же проверке работы боты.

Самое важное происходит в этой части:

events = longpoll.check()

if len(events)>0:

for event in events:

handle\_events(vk, working\_time, event, user\_dict )

Получаем ответ от ВК. И если там что-то есть, то обрабатываем полученные события.

## Функция checking\_working\_time

Что делает: проверяет, рабочее ли сейчас время

Получает:

* now\_time: int – время в формате timestamp

Возвращает:

* значение bool – рабочее время или нет

Рабочее время установлено в соответствии с режимом работы организации.

## Функция handle\_events

Что делает: раскидывает сообщения по сценариям

Получает:

* vk – экземпляр вк
* working\_time: bool – рабочее ли сейчас время
* event – событие
* user\_dict: dict – словарь пользователей

Возвращает:

* ничего

Функция осуществляет связь всех способностей бота. Отвечаем на событие только если:

* оно пришло не в беседе if event.peer\_id != 2000000002
* это сообщение if event.type == VkEventType.MESSAGE\_NEW

При этом у нас есть 2 режима обработки сообщений:

* Когда боту нельзя отвечать на сообщения
* Когда боту можно отвечать на сообщения

**Боту нельзя отвечать на сообщения**

В данном режиме бот может обработать только 2 сообщения. Одно от пользователя, одно от него самого. Оба связаны и приводят к завершению работы в данном чате (т.е. работы с данным пользователем).

[Смотреть рекомендации по улучшению 1]

**Боту можно отвечать на сообщения**

В данном режиме бот обрабатывает практически все сообщения, написанные для него, кроме тех, что попадают в режим не ответа. Мы не обрабатываем сообщение, отправленные нами, подразумевая, что все сообщения, отправленные нами – это сообщения от бота.

В первую очередь обрабатываем 4 сообщения:

1. Пустое сообщение

Пользователь скорее всего переслал сообщение и ничего не написал. Т.е. просто переслал сообщение. Просим его написать, а не пересылать.

1. Сообщение спасибо
2. Сообщение приветствия
3. Сообщение просто знак вопроса

Это простые сообщения, не нуждающиеся в классификации. Когда мы выяснили, что это ничего из списка выше, то начинаем классифицировать сообщение для запуска сценария. Делаем это с помощью функции, которую рассмотрим позже.

user\_dict[event.user\_id] = classify\_mera\_and\_tema(event.text, user\_dict, event.user\_id)

Оставшаяся часть функции обрабатывает полученный результат классификации сообщения.

[Смотреть рекомендации по улучшению 2]

## Функция classify\_mera\_and\_tema

Что делает: классифицирует сообщение

Получает:

* mess: str – экземпляр вк
* user\_dict: dict – словарь пользователей
* user\_id: int– ID пользователя, для которого классифицируем

Возвращает:

* user\_dict[user\_id]: dict – возвращает словарь только данного пользователя

**Выяснение меры**

Если до этого мы не получали сообщения от пользователя и не определяли о какой мере идет речь в сообщении, или до этого мера не определилась, то производим выяснение о какой мере идет речь.

if user\_dict[user\_id]['mera']==None or user\_dict[user\_id]['mera']=="\*\*\*\*":

user\_dict[user\_id]['mera']=kakai\_mera(request\_no\_morph\_mess)

**Выяснение темы**

Проходимся по словарю моделей, в котором

* ключ – название сценария
* значение – список с моделью, где:
  + [0] – сам классификатор (модель)
  + [1] – надо ли ему давать морфологическую форму предложения
  + [2] – важно сценарию знать, какая мера
  + [3] – важно сценарию знать, какой филиал

Соответственно, если классификатору нужен текст в морфологической форме – мы даем его, если нет – то нет.

Для векторизации используется в основном 2 векторизатора, но есть два сценария, для которых нужны именно их специальные векторизаторы. Поэтому именно их мы векторизуем непосредственно в цикле.

В этом же цикле мы сразу делаем предсказание. И если классификатор вернул, что это данная тема, тогда мы его добавляем в список определенных тем.

tems = []

for name\_scenario, model in models.items():

if model[1]:

if model[0].predict(mess\_morph): tems.append((name\_scenario, model[2], model[3]))

else:

if name\_scenario == 'Филиал' or name\_scenario == 'Благодарность':

mess = vectorizer[name\_scenario].transform([request\_no\_morph\_mess])

if model[0].predict(mess): tems.append((name\_scenario, model[2], model[3]))

elif model[0].predict(mess\_no\_morph): tems.append((name\_scenario, model[2], model[3]))

## Функция kakoi\_filifl

Что делает: определяет филиал в сообщении

Получает:

* event – событие
* user\_dict: dict – словарь пользователей

Возвращает:

* user\_dict[user\_id]: dict – возвращает словарь только данного пользователя

Проходимся по словарю филиалов. Если есть совпадение – мы нашли.

Далее делаем проверки, т.к. некоторые филиалы определяются вместе с другими по схожим словам. Теперь нам надо исключить не нужный, если мы это можем сделать. Так:

* есть просто 'Октябрьском районе', а есть 'Октяб г. Орска'. Если обнаружился 'Орск', то тогда это 'Октяб г. Орска' и нам надо удалить все признаки первого и удалить 'Орск'. По такой логике и вся остальная проверка.
* со всеми филиалами г. Оренбурга так же определяется и сам 'Оренбург'. Если какой-то из филиалов нашелся, то удаляем город.

Если мера 1 – то так и оставляем этот филиал. Если 0 или 2+ – то оставляем звезды, как неопределенное.

## Функция kakai\_mera

Что делает: определяет меру в сообщении

Получает:

* mess: str – сообщение, которое классифицируется
* user\_dict: dict – словарь пользователей

Возвращает:

* user\_dict[user\_id]: dict – возвращает словарь только данного пользователя

Проходимся по словарю мер. Если есть совпадение – мы нашли меру. Даже если мы нашли 1 меру, все равно по всему словарю, т.к. в сообщении может быть несколько мер.

Если мера 1 – то так и оставляем эту меру. Если 0 или 2+ – то оставляем звезды, как неопределенное.

Универсальное пособие узнаем отдельно, т.к. если как мера оно есть – оно важнее. Т.е. на него отдельный сценарий.

## Работа сценариев

Работа происходит +- одинаково у всех сценариев, поэтому их работу опишу на одном сценарии.

Сценарий работает в группе из:

* главного файла сценария;
* файл клавиатуры (меню);
* файл текста и сообщения сценария.

Файл меню и сообщений находиться в соответствующих папках.

Пример работы будет произведен на сценарии «Телефон/госуслуги» ( в файлах он называется odobrenie). Соответственно:

* главный файл: odobrenie
* файл меню: menu\_odobrenie
* файл сообщений: mess\_odobrenie

### Функция run\_keyboard

Что делает: отправляет сообщение и прикрепляет клавиатуру пользователю

Получает:

* vk – экземпляр вк
* user\_id: int – id пользователя, которому отправляем клавиатуру
* message: str – сообщение, которое отправляем вместе с клавиатурой
* keyboard – клавиатура, которую отправляем пользователю

Возвращает:

* ничего

### Функция write\_msg

Что делает: отправляет сообщение и открепляет клавиатуру пользователю

Получает:

* vk – экземпляр вк
* user\_id: int – id пользователя, которому отправляем клавиатуру
* message: str – сообщение, которое отправляем вместе с клавиатурой
* keyboard – клавиатура, которую отправляем пользователю (пустая клавиатура)

Возвращает:

* ничего

### Функция close\_scenario

Что делает: закрывает данный сценарий

Получает:

* user\_dict: dict – словарь пользователей
* event – событие

Возвращает:

* user\_dict: dict – словарь пользователей

### Функция check\_script\_odobrenie

Что делает: обрабатывает сообщения, направленные в данный сценарий

Получает:

* vk – экземпляр вк
* event – событие
* user\_dict: dict – словарь пользователей

Возвращает:

* user\_dict[user\_id]: dict – возвращает словарь только данного пользователя

Первым делом делаем пометку, что для данного пользователя запустили такой сценарий, что бы в последствии мы обрабатывали новые сообщения в этом сценарии.

Если это первый запуск (['check']=True), то подключаем меню. И отправляем первое сообщение по этому сценарию и клавиатуру пользователю.

Если же это не первый запуск, то начинаем обрабатывать сообщение. Если оно входит в наше меню, то сценарий работает. Если же пришло сообщение, не относящееся к данному сценарию, то сценарий закрывается и сообщение возвращается распределителю (функции handle\_events).

### Особенность сценария «Что мне положено»

Это сценарий – опрос. Он работает, как и все сценарии. Но у него есть главное отличие – пути порядка вопросов. Так вот один из путей:

next\_question\_N = {

1: {0: 2, 1: [2, 'Почетный донор']},

2: {0: 8, 1: [3]},

3: {0: 10, 1: [4]},

4: {0: 10, 1: [10, 'ЕДВ', 'Услуги связи']},

5: {0: 10, 1: [3, 'МСП на оплату жилья и коммунальных услуг']},

6: {0: 5, 1: [7, 'Пособие до 16(18) лет', 'Универсальное пособие']},

7: {0: 9, 1: [9, 'ЕДВ на 3', 'РМК', 'Статус многодетной семьи', 'Школьная форма', 'Проезд многодетным']},

8: {0: 6, 1: [6, 'Социальная стипендия']},

9: {0: 10, 1: [10, 'областная ежеквартальная надбавка детям-инвалидам']},

10: {0: None, 1: [None, 'Субсидия']}}

Ключи – это номер вопроса. Это работает таким образом:

1. Задается вопрос. В зависимости от ответа он получает 0 или 1.
2. В зависимости от ответа, запускаем следующий вопрос.

Раз у нас был первый вопрос, то получив ответ с индексом 0, мы запустим вопрос 2. Если получим ответ, с индексом 1, то так же запустим вопрос 2, но добавим в ответ 'Почетный донор'.

Теперь мы на втором вопросе. И если получим индекс 0 – запустим 8 вопрос, если индекс 1 – запустим 3 вопрос.

1. Если в переходимом вопросе стоит None, то опрос завершается.
2. При получении ответа с индексом 1, мы добавляем к ответу то, что находиться в данном вопросе, если там не только ссылка на следующий вопрос (т.е. в списке 2 значения, а не одно).
3. В любом случае путь доходит до состояния None.
4. При достижении завершения опроса, пользователю выводиться результат этого самого опроса.

# Глава 3. Часто задаваемые вопросы

Зачем бот каждые 10 минут пишет «Запланированное переподключение», хотя сам не делает его?

У ВК есть свои приколы в виде не отправки ответа на запрос. Т.е. мы можем ждать от него ответа и просто не получить. Для этого было ограничено время ожидания ответа и добавлена такая «пауза» в виде записи в логи для удобства отслеживания работы. В большинстве случаев ограничение по времени ожидания ответа спасает, но все же надо следить, ибо бывает стопориться.

Почему бот включен, но не отвечает?

Это значит, что мы не получили ответ от ВК и продолжаем находиться в режиме ожидания. Для выхода их него 1 раз зажмите Ctrl+C.

# Глава 4. Рекомендации/планы по улучшению

1. Изменить режим, в котором бот не может отвечать на сообщения.

Надо как-то понять, что, например, конкретное сообщение мы можем обработать. Т.е. человек написал, бот не смог ответить, тогда ответил оператор. После пользователь пишет ещё сообщение в течение 10 минут и уже на него бот может ответить, но он стоит в режиме не ответа для данного чата. Это надо как-то поправить.

2. Реализовать ответ на несколько тем

Мы уж получаем все темы, что определились в сообщении. Но сейчас мы обрабатываем сообщение, только если тема определилась 1. Если их будет две и более, то бот не будет отвечать. Тут есть два варианта изменения/улучшения:

* проходиться по всему списку определенных тем по очереди.   
  Отработать 1 сценарий, потом второй, потом третий и тд. Проблема: не факт, что все определенные темы — это действительно то, о чем спрашивал пользователь.
* среди определенных тем делать предсказание, какая из данных тем более вероятна.   
  Из того, что мы уже определили, можно теперь выяснить, что наиболее вероятно. Тут то же 2 пути: или придумать какой-то способ выяснения приоритета; или делать повторное предсказание этими моделями, только выбрать иной выход результат – не 0 или 1, а в % соотношении. И потом сравнивать % вероятностей. Т.е. например сценарий А выдал (30 и 60)%, а сценарий В (10 и 90)%. Оба определяться, как тема в данном сообщении, но В имеет больший процент, чем А. Поэтому мы скажем, что в данном сообщении говориться про В.

# Список литературы

1. XGBoost, LightGBM или CatBoost – какой алгоритм бустинга следует использовать? [Электронный ресурс]. – Режим доступа: <https://ai-news.ru/2021/10/xgboost_lightgbm_ili_catboost_kakoj_algoritm_bustinga_sleduet_ispolzovat.html>
2. CatBoost, XGBoost и выразительная способность решающих деревьев [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/ods/blog/645887/#1>
3. Частотный анализ русского текста и облако слов на Python [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/517410/
4. Generating WordClouds in Python Tutorial [Электронный ресурс]. – Режим доступа: https://www.datacamp.com/tutorial/wordcloud-python
5. Data validation for NLP machine learning applications [Электронный ресурс]. – Режим доступа: https://www.depends-on-the-definition.com/data-validation-for-nlp/
6. Предобработка текста в NLP [Электронный ресурс]. – Режим доступа: https://python-school.ru/blog/nlp-text-preprocessing/
7. Основы Natural Language Processing для текста [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/Voximplant/blog/446738/
8. Плавное введение в Natural Language Processing (NLP) [Электронный ресурс]. – Режим доступа: https://datastart.ru/blog/read/plavnoe-vvedenie-v-natural-language-processing-nlp/
9. Обработка текстов на естественных языках [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/vk/blog/358736/
10. pymorphy2 [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/176575/
11. Основные термины в Natural Language Processing [Электронный ресурс]. – Режим доступа: https://singularika.com/ru/nlp/natural-language-processing-terms/
12. What Is Data Cleaning and Why Does It Matter? [Электронный ресурс]. – Режим доступа: https://careerfoundry.com/en/blog/data-analytics/what-is-data-cleaning/
13. Нетрадиционный анализ тональности текста: BERT vs CatBoost [Электронный ресурс]. – Режим доступа: https://temofeev.ru/info/articles/netraditsionnyy-analiz-tonalnosti-teksta-bert-vs-catboost/
14. Data pre-processing [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Data\_pre-processing/
15. Как легко понять логистическую регрессию [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/io/blog/265007/
16. Логистическая регрессия на Python [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/skillfactory/blog/701530/
17. Мультиномиальная логистическая регрессия [Электронный ресурс]. – Режим доступа: https://translated.turbopages.org/proxy\_u/en-ru.ru.90c72965-63a87f4f-11a83352-74722d776562/https/en.wikipedia.org/wiki/Multinomial\_logistic\_regression
18. Логистическая регрессия для решения задач классификации (пример) [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Логистическая\_регрессия\_для\_решения\_задач\_классификации\_%28пример%29
19. Открытый курс машинного обучения. Тема 4. Линейные модели классификации и регрессии [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/ods/blog/323890/#2-logisticheskaya-regressiya
20. Логистическая регрессия [Электронный ресурс]. – Режим доступа: https://studfile.net/preview/10044442/page:4/
21. Классификация как задача машинного обучения [Электронный ресурс]. – Режим доступа: https://koroteev.site/text/ml2/
22. Логистическая регрессия в Python [Электронный ресурс]. – Режим доступа: https://chel-center.ru/python-yfc/2020/12/20/logisticheskaya-regressiya-v-python/#what-is-classification
23. Логистическая регрессия и ROC-анализ — математический аппарат [Электронный ресурс]. – Режим доступа: https://loginom.ru/blog/logistic-regression-roc-auc
24. Введение в логистическую регрессию [Электронный ресурс]. – Режим доступа: https://machinelearningmastery.ru/introduction-to-logistic-regression-66248243c148/
25. Text processing [Электронный ресурс]. – Режим доступа: https://catboost.ai/en/docs/concepts/python-text-processing
26. Быстрый градиентный бустинг с CatBoost [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/otus/blog/527554/
27. Catboost для самых маленьких [Электронный ресурс]. – Режим доступа: https://habr.com/ru/sandbox/122369/
28. CatBoost Enables Fast Gradient Boosting on Decision Trees Using GPUs [Электронный ресурс]. – Режим доступа: https://catboost.ai/news/catboost-enables-fast-gradient-boosting-on-decision-trees-using-gpus
29. CatBoost [Электронный ресурс]. – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost
30. Ваш первый BERT: иллюстрированное руководство [Электронный ресурс]. – Режим доступа: https://temofeev.ru/info/articles/vash-pervyy-bert-illyustrirovannoe-rukovodstvo/
31. Объяснение BERT: полное руководство с теорией и учебным пособием [Электронный ресурс]. – Режим доступа: https://evogeek.ru/articles/300167/
32. BERT в двух словах: Инновационная языковая модель для NLP [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/otus/blog/702838/
33. BERT Explained: State of the art language model for NLP [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
34. Что из себя представляет алгоритм Google BERT и как он работает [Электронный ресурс]. – Режим доступа: https://www.seonews.ru/events/chto-iz-sebya-predstavlyaet-algoritm-google-bert-i-kak-on-rabotaet/
35. BERT — state-of-the-art языковая модель для 104 языков. Туториал по запуску BERT локально и на Google Colab [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/436878/
36. Как использовать BERT для мультиклассовой классификации текста [Электронный ресурс]. – Режим доступа: https://neurohive.io/ru/tutorial/bert-klassifikacya-teksta/
37. Глубокое понимание LightGBM [Электронный ресурс]. – Режим доступа: https://russianblogs.com/article/6403728054/
38. С.Николенко, А.Кадурин, Е.Архангельская. Глубокое обучение. Погружение в мир нейронных сетей. – СПб.: Питер, 2018. – 480 с.
39. Джоэл Грас. Data Science. Наука о данных с нуля. – СПб.: БХВ-Петербург, 2018. – 336 с.
40. Хобсон, Л. Обработка естественного языка в действии. / Л. Хобсон, Х. Хапке, К. Ховард. – Санкт-Петербург : Питер, 2020. – 576 с.
41. Надежда Владиславова. Продвинутые техники NLP. – М.: София, 2015. – 320 с.
42. М.М. Пелехатый, Ю.А. Чекчурин. Сертификационный курс НЛП-Практик. – М.: Твои книги, 2015. – 272 с.
43. Ричард Риз. Обработка естественного языка на Java. – М.: ДМК Пресс, 2016. – 264 с.
44. Классификация и методы создания чат-бот приложений: сайт. – Режим доступа: <https://cyberleninka.ru/article/n/klassifikatsiya-i-metodysozdaniya-chat-bot-prilozheniy/viewer> – 10.03.2023.
45. Как использовать чат-боты в бизнесе: 5 идей и 5 кейсов // vc.ru : сайт. – Режим доступа: <https://vc.ru/services/93850-kak-ispolzovat-chat-boty-v-biznese-5-idey-i-5-keysov> – 25.02.2023.
46. Чат-боты: для чего нужны и как создать: сайт. – Режим доступа: <https://skillbox.ru/media/marketing/gayd-chatboty/> – 10.03.2023.
47. Смыслова, Л.В. Чат-бот как современное средство интернет-коммуникаций / Л.В. Смыслова. – М: Молодой ученый, 2018. – 150 с.
48. Что такое Чат-Бот: Определение и Руководство: сайт. – Режим доступа: <https://sendpulse.com/ru/support/glossary/chatbot> – 06.02.2023.
49. Акулич, М.В. Чат-боты и маркетинг / М.В. Акулич. – М: Ridero, 2022. – 180 с.
50. Кузнецов, В.В. Перспективы развития чат-ботов / В.В. Кузнецов // Успехи современной науки: электронный журнал. – Режим доступа: <https://www.elibrary.ru/item.asp?id=27712514> – ISSN 2412-6608. – 10.03.2023.
51. Срини, Д. Разработка чат-ботов и разговорных интерфейсов /  
    Д. Срини. – М: ДМК-Пресс, 2019. – 340 с.
52. Чат-боты – кто они и что умеют?: сайт. – Режим доступа: <https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html> –20.02.2023.
53. Гераськов, С. Доступный чат-бот / С. Гераськов. – М: Synergy Book, 2020. – 120 с.
54. Как создать чат-бота или бота для канала в Телеграм // iflife.ru: сайт. – Режим доступа: <https://iflife.ru/telegram/kak-sozdat-bota> – 20.02.2023.
55. Элиза, Пэри и Алиса: история и эволюция чат-ботов // REG.RU: сайт. – Режим доступа: <https://www.reg.ru/blog/istoriya-i-evolyutsiya-chat-botov/> – 12.02.2023.
56. Понкратов, В.А Анализ возможностей чат-ботов / В.А Понкратов // Актуальные проблемы экономики и управления. – Гжель, 2022. – С. 115-120.
57. Как создать и подключить Telegram чат-бота: сайт. – Режим доступа: <https://sendpulse.com/ru/knowledge-base/chatbot/telegram/create-telegram-chatbot> –10.03.2023.
58. Микаэл, Я. Чат-боты будут учить ваших детей: сайт. – Режим доступа: <https://rb.ru/opinion/bot-study/> – 10.03.2023.
59. От чат-ботов к разговорному ИИ: разработка умных ассистентов для бизнеса // vc.ru: сайт. – Режим доступа: <https://vc.ru/promo/42838-aimylogic> – 16.02.2023.
60. Работа-работа, перейди на чат-бота: как развивается российский рынок виртуальных операторов // Официальный сайт СберПро: сайт. – Режим доступа: <https://sber.pro> – 14.02.2023.
61. Чат-боты в Telegram на Python // Skillbox: сайт. – Режим доступа: <https://skillbox.ru/media/code/chatboty-v-telegram-na-python> – 04.03.2023.
62. Чат-боты: история, современность и перспективы: сайт. – Режим доступа: <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460/?ysclid=lb1ykhsj8s350242066> – 19.02.2023.
63. Как создать чат-бот в Telegram: от настроек до запуска // TexTerra: сайт. – Режим доступа: <https://texterra.ru/blog/kak-sozdat-chat-bot-v-telegram-ot-nastroek-do-zapuska.html> – 14.10.2023.
64. Рытикова, В.О. Разработка чат-бота помощника на языке Python для ускорения работы техподдержки компании СКБ Контур: специальность р: диссертация на соискание студент группы РИМ-281217 / Рытикова В.О. – Екатеринбург, 2020. – 109 с.
65. Чат-бот для Telegram // Jivo: сайт. – Режим доступа: <https://www.jivo.ru/blog/tutorials-jivo/chat-bot-history.html> – 04.03.2023.
66. Классификация и методы создания чат-бот приложений // КиберЛенинка: сайт. – Режим доступа: <https://cyberleninka.ru/article/n/klassifikatsiya-i-metody-sozdaniya-chat-bot-prilozheniy> – 25.02.2023.
67. Чат-бот vs живой чат – что более эффективно? // Webest: сайт. – Режим доступа: <https://webest.ru/blog/chat-bot-vs-zhivoy-chat-chto-bolee-effektivno>
68. Эффективность использования чат-ботов в образовательном процессе / Б.С. Горячкин, Д.А. Галичий, В.С. Цапий [и др.] // E-Scio. – 2021. – № 4 (55). – С. 529-551.
69. Куликова, О.М. Роль чат-ботов в построении эффективных коммуникаций / О.М. Куликова, С.Д. Суворова // Международный журнал гуманитарных и естественных наук. – 2021. – № 4-3 (55). – С. 33-37.
70. Как использовать чат-ботов: сценарии и примеры // uplab: сайт. – Режим доступа: <https://www.uplab.ru/blog/how-to-use-chat-bots/>
71. Сухас, У. Краткий обзор: Чат-боты с искусственным интеллектом: информационный документ-пособие / У. Сухас. – 2017. – 24 с.
72. Что такое чат-бот?: сайт. – Режим доступа: <https://www.oracle.com/cis/chatbots/what-is-a-chatbot/>