**Цифровая технология ФИЛЬТРАЦИИ**

**потоков сообщений В СОЦИАЛЬНЫХ СЕТЯХ**

**ШИФР: соцфильтр**

Исмакова Даяна

Содержание

[ВВЕДЕНИЕ 1](#_Toc56764083)

[1. Сценарий решения задачи А-м «как есть» 2](#_Toc56764084)

[2. Анализ сценария “как есть” и формирование сценария “как будет” 4](#_Toc56764085)

[3. Проектирование СОЦФИЛЬТРа 5](#_Toc56764086)

[3.1. Требования «клиента» (А) к ЦТ 5](#_Toc56764087)

[3.2. Требования к функционалу ЦТ 5](#_Toc56764088)

[3.3. Концепция СОЦФИЛЬТРа 5](#_Toc56764089)

[4. Архитектура СОЦФИЛЬТРа 6](#_Toc56764090)

[4.1. Инструментарий разработки ЦТ 6](#_Toc56764091)

[4.2. Физическая структура 8](#_Toc56764092)

[4.3. Логическая структура 9](#_Toc56764093)

[4.4. Функциональная структура 10](#_Toc56764094)

[ЗАКЛЮЧЕНИЕ 12](#_Toc56764095)

[ИСТОЧНИКИ 13](#_Toc56764096)

# ВВЕДЕНИЕ

Ручная фильтрация негативных комментариев (***НК***) занимает много времени и вредит здоровью [1, 2], а существующая автоматическая часто работает некорректно. Факторы, замедляющие обработку ***НК***:

* использование замещающих символов, которые не способен определить компьютер;
* слова, которые вне контекста являются корректными;
* оскорбления, затрагивающие определенную группу людей без использования некультурной лексики.

**Проблема**: распространение токсичной лексики в социальных сетях и отсутствие ее быстрой и качественной модерации.

**Цель**: создать ЦТ, повышающую скорость и качество обработки ***НК***. Стратегическая Ц: ИС = ***СОЦФИЛЬТР*** + ***А***. (миссия ИС - нормальные условия общения в соцсети).

# 1. Сценарий решения задачи А-м «как есть»

**Задача**: исключить из потока сообщений негатив в течение 2 часов.

* Исполнитель: администратор социальной сети (***А***).
* Предусловие: ***А*** авторизован, открыта социальная сеть.

Основной процесс представлен в табл. 1 в виде последовательности операций, доступных ***А*** для манипулирования комментариями (***К***) клиентов соцсети (***П***) и их групп.

Таблица 1. Решение задачи администратором “Обнаружение и исправление контента в течение 2ч.”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Операция** | **Сценарии** | **Действие *А*** | **Отклик соцсети** | **Возможная программная реализация** |
| 1 | Открытие окна ... | 1.1) ... ***К*** в группе | 1.1.1) Клик «группа»  1.1.2) Клик «пост»  1.1.3) Секция с ***К*** | Показ ***К*** | Одно окно для показа всех К |
|  |  | 1.2) … жалоб на ***К*** | 1.2.1) Клик «окно жалоб» | Показ ***К*** |  |
| 2 | Выделение ***А***-м | 2.1) Распознавание ***НК*** | 2.1.1) Прочитывает К  2.1.2) Находит НК  2.1.3) Определяет степень токсичности | - | Автоматическая обработка текста с распознаванием токсичности ***К*** |
| 3 | Принятие решения | 3.1) Низкая степень токсичности ***К*** | 3.1.1) {2.1.1 – 2.1.3}  3.1.2) Определил ***К*** как «Низкая степень токсичности»  3.1.3) Смотрит, есть ли данный ***П*** в списке нарушителей  3.1.3.1) Есть, но не превысил макс. количество раз, отправляет предупреждение ***П***  3.1.3.2) Есть, превысил норму, блокировка ***П*** на неделю | Сохраняет ***П*** в список нарушителей | Построение BERT модели или наивного Байесовского классификатора для распознавания ***НК***,  для конфликтных ситуаций предложить замену НК |
|  |  | 3.2) Средняя степень токсичности К | 3.2.1) {2.1.1 – 2.1.3}  3.2.2) Определил ***К*** как «Средняя степень токсичности»  3.2.3) Смотрит, есть ли данный ***П*** в списке нарушителей  3.2.3.1) Есть, но не превысил макс. количество раз, отправляет предупреждение ***П***  3.2.3.2) Есть, превысил норму, ***П*** блокировка на 2 недели  2.3.г) Удаляет ***К*** | Сохраняет ***П*** в список нарушителей,  удаляет ***НК*** | Построение BERT модели или наивного Байесовского классификатора для распознавания ***НК***,  для конфликтных ситуаций предложить решение |
|  |  | 3.3) Высокая степень токсичности | 3.3.1) {2.1.1 – 2.1.3}  3.3.2) Определил ***К*** как «Высокая степень токсичности»  3.3.3) Смотрит, был ли данный ***П*** в списке нарушителей  3.3.3.1) Был уж много раз, пожизненная блокировка  3.3.3.2) Был, блокировка на месяц  3.3.3.3) Не был, блокировка на 2 недели  3.4) Удаляет ***К*** | Сохраняет ***П*** в список нарушителей,  удаляет ***НК*** | Построение BERT модели или наивного Байесовского классификатора для распознавания ***НК***,  для конфликтных ситуаций предложить решение |
| 4 | Составление статистики за конкретный период | 4.1) По группам | 4.1.1) Открывает окно статистики  4.1.2) Отмечает группы с самым большим количеством ***НК***  4.2.3) Сохраняет список групп | Сохраняет списки групп | Автоматическое сохранение список групп в БД с  обновлением статистики каждые 2 часа.  По нажатию на кнопку - вывод отчета по группам |
|  |  | 4.2) По пользователям | 4.2.1) Открывает окно статистики  4.2.2) Отмечает пользователей с самым большим количеством ***НК***  4.2.3) Сохраняет список пользователей | Сохраняет списки пользователей | Автоматическое сохранение список пользователей в БД с  обновлением статистики каждые 2 часа.  По нажатию на кнопку - вывод отчета по группам  м |
| 5 | Блокировка | 5.1) Групп | 5.1.1) Получение статистики п.4.1  5.1.2.1) Если группа не исправила контент и до этого не было блокировок, блокировка на неделю  5.1.2.2) Если группа не исправила контент и до этого было от 1 до 4х блокировок, блокировка на месяц  5.1.2.3) Если группа не исправила контент и до этого было более 4х блокировок, пожизненная блокировка | - | Автоматическая блокировка |
|  |  | 5.2) Пользователей | 5.2.1) Получение статистики п.4.2  5.2.2.1) Если ***П*** не исправил контент и до этого не было блокировок, блокировка на неделю  5.2.2.2) Если ***П*** не исправил контент и до этого было от 1 до 4х блокировок, блокировка на месяц  5.2.2.3) Если ***П*** не исправил контент и до этого было более 4х блокировок, пожизненная блокировка | - | Автоматическая блокировка |
| 6 | Полуавтоматическая фильтрация | 6.1) Разрешение конфликтных ситуаций | 6.1.1) Определение степени токсичности  6.1.1.1) ***К*** не относится к ***НК,*** ***А*** убирает ***К*** из списка негативных  6.1.1.2) ***К*** относится к ***НК***, ***А*** определяет токсичность ***К***, переход к п. {3.1-3.3} | Запоминает выбранные результаты | Сохранение в БД и переобучение по новой выборке |

# 2. Анализ сценария “как есть” и формирование сценария “как будет”

Проанализировав сценарий “как есть”, были выявлены следующие недостатки при работе ***А*** с соцсетью:

1. многократное повторение одинаковых действий,
2. постоянное переключение между окнами системы,
3. ручное оформление отчетов и статистики за определенный период,
4. ручная блокировка групп и пользователей,
5. длительная обработка комментариев.

**Р**ешение этих проблем приведено в колонке “возможная программная реализация” табл.1.

Автоматизировав процесс работы А, получаем сценарий “как будет” табл.2.

**Задача**: исключить из потока сообщений негатив в течение 2 часов.

* Исполнитель: администратор социальной сети (***А***).
* Предусловие: ***А*** авторизован, открыто приложение СОЦФИЛЬТРа.

Основной процесс представлен в табл. 2 в виде последовательности операций, доступных ***А*** для манипулирования комментариями (***К***) клиентов соцсети (***П***) и их групп.

Таблица 2. Сценарий “как будет”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Операция | Действия *А* | Действия СОЦФИЛЬТРа |
| 1 | Открытие окна | Клик “приложение СОЦФИЛЬТР” | Открывается окно приложения, запускается мониторинг потока *К* |
| 2 | Выявление *НК* | - | Находит НК |
| 3 | Фильтрация | 3.1.2. Решает *КФ* | 3.1. Определяет степень токсичности *К*  3.1.1. Степень токсичности не определена, отправляет конфликтную ситуацию (*КФ*) *А*  3.1.3. Сохраняет решение  3.2. В соответствии с уровнем токсичности определяет шаги для устранения *К* |
| 4 | Составление статистики | 4.1. Клик “составить статистику”  4.2. Клик “за период” | Составляет статистику |
| 5 | Блокировка | - | Мониторинг комментаторов и групп, блокировка при нарушении правил |

# 3. Проектирование СОЦФИЛЬТРа

## 3.1. Требования «клиента» (А) к ЦТ

ЦТ должна выполнять следующее:

1. время формирования и оформления статистики за 2ч. < 30 сек.;
2. < 5% конфликтных ситуаций;
3. обрабатывать ≥ 400 комментариев в течение 1 часа (вместо одного дня);
4. количество кликов ≤ количества конфл. ситуаций;
5. графо-тексто-речевой, объектный, сценарный диалог.

## 3.2. Требования к функционалу ЦТ

ЦТ должна выполнять следующие требования:

1. обрабатывать входные К;
2. сохранять К в БД;
3. определять токсичность К;
4. вести статистику К по группам и пользователям;
5. блокировать пользователей и группы.

## 3.3. Концепция СОЦФИЛЬТРа

СОЦФИЛЬТР - это ЦТ для автоматической фильтрации токсичных комментариев, использующая нейронную сеть при распознавании НК.

# 4. Архитектура СОЦФИЛЬТРа

## 4.1. **Инструментарий разработки ЦТ**

В качестве языка для работы с серверной частью был выбран Python [10]. Он позволяет использовать различные библиотеки без интеграции со сторонними приложениями.

Для работы с клиентской частью будет использоваться библиотека Vue.js [14], основанная на JavaScript и совмещающая в себе огромное количество компонентов для быстрой и функциональной работы приложения.

Мостом между клиентской и серверной частью приложения будет Flask [11] на Python [10] и Axios [12] для Vue.js [14].

Таблица 3. Инструментарий СОЦФИЛЬТРа

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Модули (М) СОЦФИЛЬТРа** | **Программные инструменты** | **Обоснование выбора** |
| 1 | Интерфейс пользователя | Vue.js [14] | Vue.js - удобный фреймоворк для работы с клиентской частью приложения. Позволяет создать интерактивную ЦТ, с уменьшением количества кликов. |
| 2 | М оповещения | Vue.js [14] | Использование одного фреймворка позволяет сделать работу оповещений быстрой без использования стороннего функционала. |
| 3 | М составления отчетов | Pandas [17] | Pandas – Python библиотека для работы с документами и созданием отчетности. Возможность работать с основным языком программирования. |
| 4 | М архивирования | DB-API [15], psycopg2 [16] | Psycopg2 - библиотека для работы с PostgreSQL, DB-API - позволяет уменьшить количество кода за счет простых операций. |
| 5 | М поддержки принятия решений | Python [10] + обученная нейронная сеть | Позволяет использовать программные решения без стороннего функционала. |
| 6 | М обработки текста | nltk [13] | Nltk – Python библиотека, в которой собраны различные компоненты для обработки текст. |
| 7 | М работы с социальной сетью | VK Api [4] | VK Api предоставляет методы для работы с социальной сетью Вконтакте. |
| 8 | М сохранения решений конфликтных ситуаций | DB-API [15], psycopg2 [16] | Psycopg2 - библиотека для работы с PostgreSQL, DB-API - позволяет уменьшить количество кода за счет простых операций. |
| 9 | Обучающая модель | BERT [8] | BERT – современная модель для машинного обучения. |

## 4.2. Физическая структура

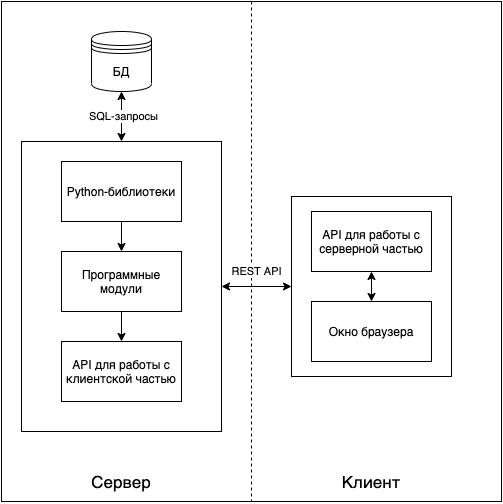


Рисунок 1. Физическая структура СОЦФИЛЬТРа

## 4.3. Логическая структура

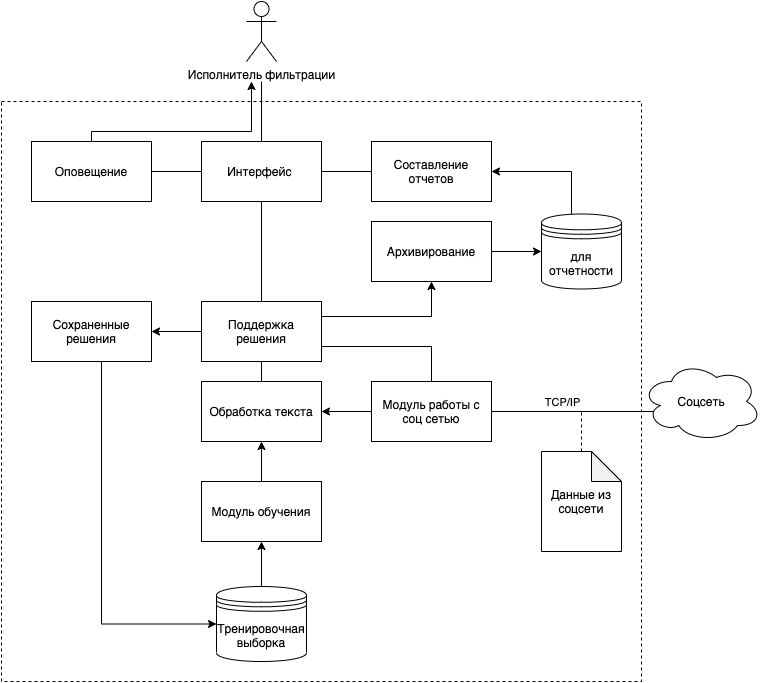


Рисунок 2. Логическая структура СОЦФИЛЬТРа

## 4.4. Функциональная структура

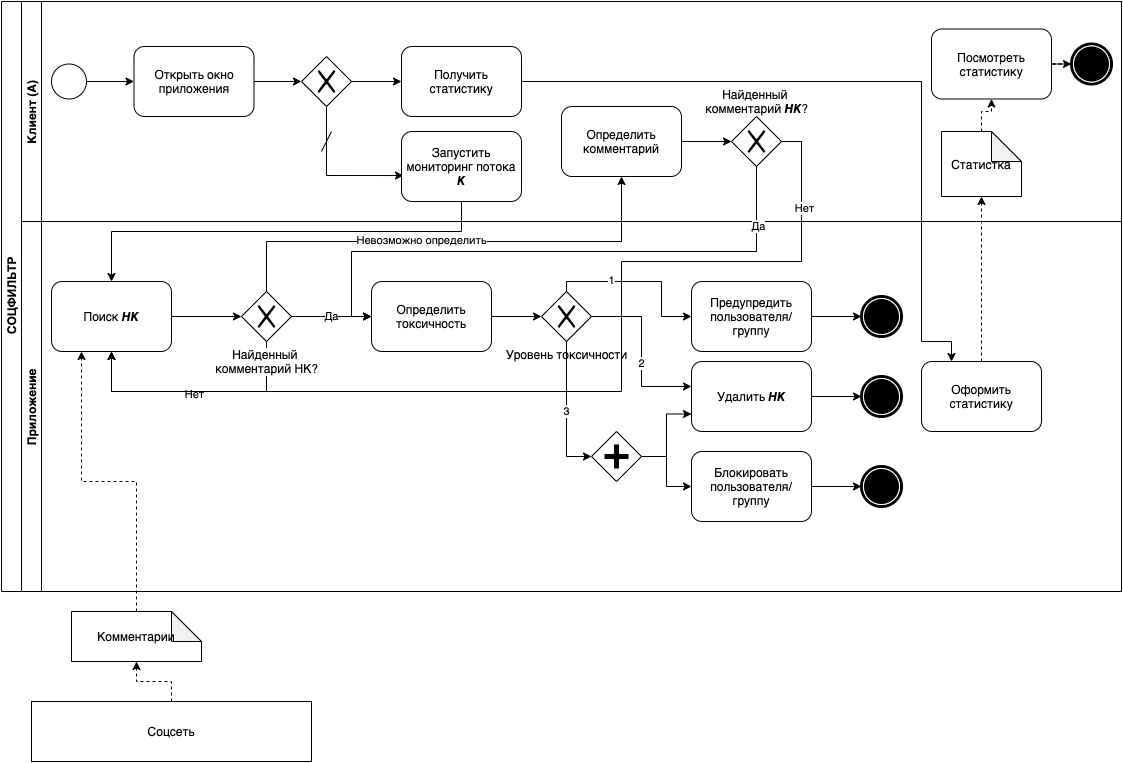


Рисунок 3. Функциональная диаграмма СОЦФИЛЬТРа в нотации bpmn

**5. Реализация?**

# ЗАКЛЮЧЕНИЕ

Результаты проделанной работы?

Планы на будущее?

# ИСТОЧНИКИ

1. Flint V. Темная сторона Google: проблемы модерации. URL: <https://dw.expert/2020/03/30/temnaya-storona-google-problemy-moderacii-eksklyuzivnoe-intervyu-s-dejzi-soderberg-rivkin/> (Дата обращения: 04.11.2020).
2. Чумаков Н. Как работают модераторы «Вконтакте»: полная анонимность, стресс и тысячи постов в сутки. URL: <https://tjournal.ru/internet/88068-kak-rabotayut-moderatory-vkontakte-polnaya-anonimnost-stress-i-tysyachi-postov-v-sutki> (Дата обращения: 04.11.2020).
3. Coding by pixc, Пишем простой парсер ВК на python. URL: <https://medium.com/@pixc/простой-парсер-вк-на-python-5f202814eac3> (Дата обращения: 04.11.2020).
4. Вконтакте, Методы API. URL: <https://vk.com/dev/methods> (Дата обращения: 04.11.2020).
5. Wikipedia, Предварительная обработка данных. URL: [https://ru.wikipedia.org/wiki/Предварительная\_обработка\_данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%B2%D0%B0%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (Дата обращения: 10.11.2020).
6. ???, Предварительная обработка данных. URL: <http://pzs.dstu.dp.ua/DataMining/preprocessing/index.html> (Дата обращения: 10.11.2020).
7. Сметанин С. Определение токсичных комментариев на русском языке. URL: <https://habr.com/ru/company/mailru/blog/526268/> (Дата обращения: 10.11.2020)
8. Wikipedia, BERT (language model), URL: <https://en.wikipedia.org/wiki/BERT_(language_model>) (Дата обращения: 13.11.2020)
9. Шкарин С. Ваш первый BERT. URL: <https://habr.com/ru/post/498144/> (Дата обращения: 13.11.2020).
10. Python, Документация Python. URL: <https://www.python.org/doc/> (Дата обращения: 17.11.2020).
11. Flask, Документация Flask. URL: <https://flask.palletsprojects.com/en/1.1.x/> (Дата обращения: 19.11.2020).
12. Axios, Используем Axios для доступа к APIPI. URL: <https://ru.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html> (Дата обращения: 19.11.2020).
13. NLTK, Документация NLTK. URL: <https://www.nltk.org> (Дата обращения: 17.11.2020).
14. Vue.js, Документация Vue.js. URL: <https://vuejs.org> (Дата обращения: 17.11.2020).
15. DaneSoul А. Python: Работа с базой данных, часть ½: используем DB-API. URL: https://habr.com/ru/post/321510/ (Дата обращения: 17.11.2020).
16. Khashtamov. A. Работа с PostgreSQL в Python. URL: <https://khashtamov.com/ru/postgresql-python-psycopg2/> (Дата обращения: 19.11.2020).
17. Pandas, Документация Pandas. URL: <https://pandas.pydata.org> (Дата обращения: 17.11.2020).