ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ

«ШАГ В БУДУЩЕЕ, МОСКВА»

1111

регистрационный номер

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

КОМПЬЮТЕРНАЯ СИСТЕМА ОБРАБОТКИ ТЕКСТА, СОДЕРЖАЩЕГО НЕЦЕНЗУРНУЮ ЛЕКСИКУ

Автор: Шалаев Алексей Дмитриевич

ГБОУ г. Москвы «Школа № 1537 «Информационные технологии» 10 класс

Научный руководитель: Минченко Михаил Михайлович

ГБОУ г. Москвы «Школа № 1537 «Информационные технологии»

к.э.н., куратор Инновационно-технологического центра

_____

Москва – 2021

Компьютерная система обработки текста,
содержащего нецензурную лексику

Аннотация

Цель работы – создание компьютерной системы (КС), способной анализировать текст с целью исключения нецензурной лексики.

Методологическую основу разработанных и программно-реализованных алгоритмов составляет метод нормализации слов.

Средство программной реализации – язык объектно-ориентированного программирования C++ с использованием среды разработки Visual Studio 2019. Данная программная платформа обеспечивает расширенный функционал обработки, анализа строковых выражений. Исходными данными являются текстовые файлы различного содержания, загружаемые пользователем или введенные наборы символов с помощью клавиатуры.

Программную структуру компьютерной системы можно представить в виде последовательности основных этапов:

1. Токенизация — разбиение строк текста на слова (от пробела до пробела).
2. Удаление всех символов, не являющимися буквами русского и английского алфавитов внутри слова.
3. Приведение всех символов к нижнему регистру.
4. Замена латинских схожих букв на русские («n» — «п», «u» — «и»).
5. Стемминг — процесс нахождения основы слова с помощью алгоритма Стеммера Портера.
6. Сравнение полученной “нормальной” формы слова с базой нецензурных слов и его замена при совпадении.

В результате программной разработки создана компьютерная система, выявляющая нецензурную лексику в исходном тексте и заменяющая ее специальным набором символов. Разработанная компьютерная система может найти практическое применение в СМИ и различных интернет-ресурсах.

Содержание

ВВЕДЕНИЕ

Проблема употребления нецензурной лексики относится к проблемам экологии языка. Современная молодёжь является носителем и хранителем будущего современного русского языка. Экологическую катастрофу легче предупредить, чем ликвидировать. Борьба за чистоту языка – долг каждого.

Бороздя просторы интернета, особенно различного рода форумы, конференции, блоги, становится неловко от применения пользователями бранных выражений, ругани, мата. Ненормативная лексика прочно вошла в нашу жизнь. Большинство граждан, особенно молодых, не представляют свою речь без нецензурного выражения.

В наше время мат широко используется:

- для повышения эмоциональности речи
- эмоциональной разрядки
- оскорбления
- унижения адресата речи
- демонстрации агрессии
- демонстрации отсутствия страха

Но, так или иначе, а у российского законодательства свои взгляды на употребление мата в общественных пространствах. И оно квалифицирует применение в общении нецензурных выражений как административное нарушение.

3 февраля - Всемирный день борьбы с ненормативной лексикой. Свой вклад в общее и, безусловно, благородное дело внесли и российские законодатели: с 1 февраля 2021 года на территории Российской Федерации вступила в силу новая редакция Федерального закона от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и защите информации", которая предписывает обязательное удаление матерных постов из соцсетей.

Для достижения цели снижения употребления нецензурной лексики в речи молодежи нужно искать различные методы. Одним из таких методов является принятие закона, запрещающего использовать нецензурную лексику, при нарушении закона придется заплатить административный штраф. Однако этот метод, как и любой другой отдельно взятый, не поможет полностью избавиться от проблемы. Меры должны приниматься комплексно. Компьютерная система, описанная в данной работе, является также одним из методов борьбы с нецензурной лексикой. Предполагается, что перед публикацией статьи, посты или любые другие текстовые данные будут проходить проверку на наличие нецензурной лексики. Как следствие, в интернете значительно снизится наличие ненормативных выражений, пользователи будут меньше видеть, а значит и использовать в своем лексиконе матерные слова.

Цель работы – создание Компьютерной системы (КС), функционалом которой является анализ текста на предмет выявления и последующего исключения нецензурной лексики. Анализу и корректировке можно подвергнуть текстовый контент из различных источников: социальных сетей, мессенджеров и так далее.

Основная ***задача работы*** – компьютерная реализация алгоритма, выявляющего и убирающего нецензурную лексику.

Актуальность разработки обусловлена принятием нового закона, в соответствии с которым социальные сети теперь обязаны самостоятельно выявлять и блокировать незаконный контент, к которому относятся и публикации, содержащие «нецензурную брань» — употребление ненормативной лексики в Российской Федерации является «действием, образующим состав административного правонарушения».

МЕТОДЫ АЛГОРИТМИЧЕСКОЙ И ПРОГРАММНОЙ РЕАЛИЗАЦИИ

Методологическую основу разработанных и программно реализованных алгоритмов составляет метод нормализации слов.

Алгоритмическую и программную структуру компьютерной системы можно представить в виде следующей последовательности основных этапов:

1. Токенизация – это самый первый шаг при обработке текста. Заключается в разбиении (разделении) длинных строк текста на слова (от пробела до пробела).
2. Приведение всех символов к нижнему регистру.
3. Замена букв на аналогично выглядящие латинские («п» – «n», «и» – «u»).
4. Удаление внутри каждого слова всех символов, не являющихся буквами русского и английского алфавитов.
5. Стемминг – процесс нахождения основы слова для заданного исходного слова (на основе реализации алгоритма Стемминга Портера).
6. Сравнение полученной “нормальной” формы слова с базой нецензурных слов и его замена при совпадении.

Подробнее рассмотрим и продемонстрируем перечисленные пункты, принципы их программной реализации и выполняемые функции. Для наглядного примера возьмем строковую последовательность: «D.\u!p_@ki вы» и будем рассматривать ее изменения на каждом этапе.

1 этап. Токенизация

Токенизация, или, другими словами, сегментация текста, производится во время получения текста пользователя. Программа считывает текст построчно и разбивает каждую строку на отдельные слова. За счет встроенных в стандартные библиотеки C++ функций алгоритмом токенизации является создание подстроки от начала слова до ближайшего вхождения пробела. Данный алгоритм несет в

себе цель рассмотрения каждого слова по отдельности. В этапах, описанных ниже, используется копия вектора слов, полученных на данном шаге.

Действие: разделение (токенизация)

Результат: [‘D.\u!p_@кі’ , ‘вы’] – 2 слова (токена).

2 этап. Приведение символов к нижнему регистру

На данном этапе программа заменяет все символы, являющиеся заглавными буквами русского и английского алфавитов, на строчные по кодировке ASCII. Это необходимо для упрощения реализации последующих этапов программы и сокращения времени ее работы. По-другому – это оптимизация процесса, которая позволяет КС сократить алфавит символов, используемых в программе.

Действие: замена (‘D’ = ’d’).

Результат: [‘d.\u!p_@кі’ , ‘вы’].

3 этап. Замена русских букв на аналогично выглядящие латинские

Этот этап программной реализации является очень важным в структуре КС. Со временем пользователи интернет-ресурсов научились “маскировать” нецензурную лексику, для этого они заменяли русские буквы на символы или набор символов, после чего смысл слова оставался прежним, а его написание менялось. Наиболее популярные способы “маскировки” в интернете представлены ниже:

- Смешение кириллицы и транслита;
- Замена букв на аналогично выглядящие латинские («п» — «n», «и» — «u»).
- Замена согласных на парные созвучные им (например «х» на «к»);
- Употребление псевдографики.

Например, русскую букву “а” заменяли на латинскую букву “a” или специальным символом “@”. Список замен, использованных в программе, представлен ниже.

'а' < = > ['а', 'a', '@']	'б' < = > ['б', '6', 'b']
'в' < = > ['в', 'b', 'v', 'w']	'г' < = > ['г', 'g']
'д' < = > ['д', 'd']	'е' < = > ['е', 'ё', 'e']
'ж' < = > ['ж', 'zh', '*']	'з' < = > ['з', '3', 'z']
'и' < = > ['и', 'й', 'i']	'к' < = > ['к', 'k', 'i{', '|{']
'л' < = > ['л', 'l', 'ji']	'м' < = > ['м', 'm']
'н' < = > ['н', 'n']	'о' < = > ['о', 'o', '0']
'п' < = > ['п', 'p']	'р' < = > ['р', 'r', 'p']
'с' < = > ['с', 'c', 's', '$']	'т' < = > ['т', 't']
'у' < = > ['у', 'y', 'u']	'ф' < = > ['ф', 'f']
'х' < = > ['х', 'x', '}{', 'h']	'ч' < = > ['ч', 'ch']
'ш' < = > ['ш', 'sh']	'щ' < = > ['щ', 'sch']
'ю' < = > ['ю', 'io']	'я' < = > ['я', 'ya', 'уа']

Действие: замена (‘d’ = ’д’, ‘u’ = ’у’, ‘@’ = ’а’, ‘i’ = ‘и’).
Результат: [‘д.\у!р_аки’ , ‘вы’] .

4 этап. Удаление ложных символов

Следующий немаловажный этап программной обработки текста – удаление внутри слова символов, не являющихся буквами русского и английского алфавитов. Помимо “маскировки” символов, реализуемой на 3 этапе, в интернете появилась тенденция ”ложных” символов, появление которых не учитывается. КС определяет значение символа, и, если он не является буквой, то КС удаляет его, то есть символы – такие, как знаки препинания, подчеркивания, дефисы, косые черты и так далее.

Действие: удаление (‘.’, ‘\’, ‘!’, ‘_’).

Результат: [‘дураки’ , ‘вы’] .

5 этап. Стемминг

Пятый этап программной обработки текста является основной функцией нормализации слов. Стемминг – это нахождение основы слова (стеммы). Термин стемминг образован от слова «stem» – ствол, стебель, основа. Алгоритм не использует баз основ слов, а лишь, применяя последовательно ряд правил, отсекает окончания и суффиксы, основываясь на особенностях языка, в связи с чем работает быстро, но не всегда безошибочно. Для реализации алгоритма Стемминга Портера была взята базовая библиотека C++11 “regex” (библиотека регулярных выражений).

Действие: удаление окончания ‘и’.

Результат: [‘дурак’, ‘вы’].

6 этап. Цензурирование

Завершающий этап программы – цензурирование. После получения “нормальной” формы слова остается понять, принадлежит ли оно к нецензурной лексике. Для этого автором был создан текстовый файл-тезаурус, содержащий большое количество матерных и оскорбляющих слов, прошедших алгоритм Стемминга Портера. Все исходные данные после обработки сверяются с массивом “плохих” слов. А также происходит особая проверка слова на случайное срабатывание цензурирования. Слово сверяется со списком исключений, которые не являются плохими словами, но содержат в своей основе матерное слово, например глагол “оскорблять”. При совпадени исходное слово заменяется на нейтральный набор символов: “[censored]”.

Действие: замена (‘D.\u!p_@ki’ = ‘[censored]’).

Результат: [‘[censored]’ , ‘вы’].

Средство программной реализации – язык объектно-ориентированного программирования C++ с использованием среды разработки Visual Studio. Эта инструментальная платформа обеспечивает расширенный функционал обработки, анализа и генерации строковых выражений, а также предоставляет более надежный доступ к файловым данным операционной системы.

Исходными данными для программы являются:

1. введенные наборы символов в текстовое поле
2. текстовые файлы различного содержания, загружаемые пользователем
3. фотография с русским текстом
4. снимок части экрана с русским текстом

РЕЗУЛЬТАТЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

Выполнение этапов обработки текста в структуре *Компьютерной системы* реализовано строго последовательно. Выполнение каждого из выделенных этапов возможно и без данных, получаемых в результате работы предыдущего. Однако тогда можно получить ошибочный результат.

В результате программной разработки создана *Компьютерная система*, обеспечивающая программную реализацию 4 основных команд в графическом окне (см. рис. 1):

Рис. 1 Главное меню

1. Ввод слов с клавиатуры осуществляется в текстовое поле. По нажатию на кнопку “ПУСК” программа выдает результат в текстовом поле, заменяя исходные данные (см. рис. 2).

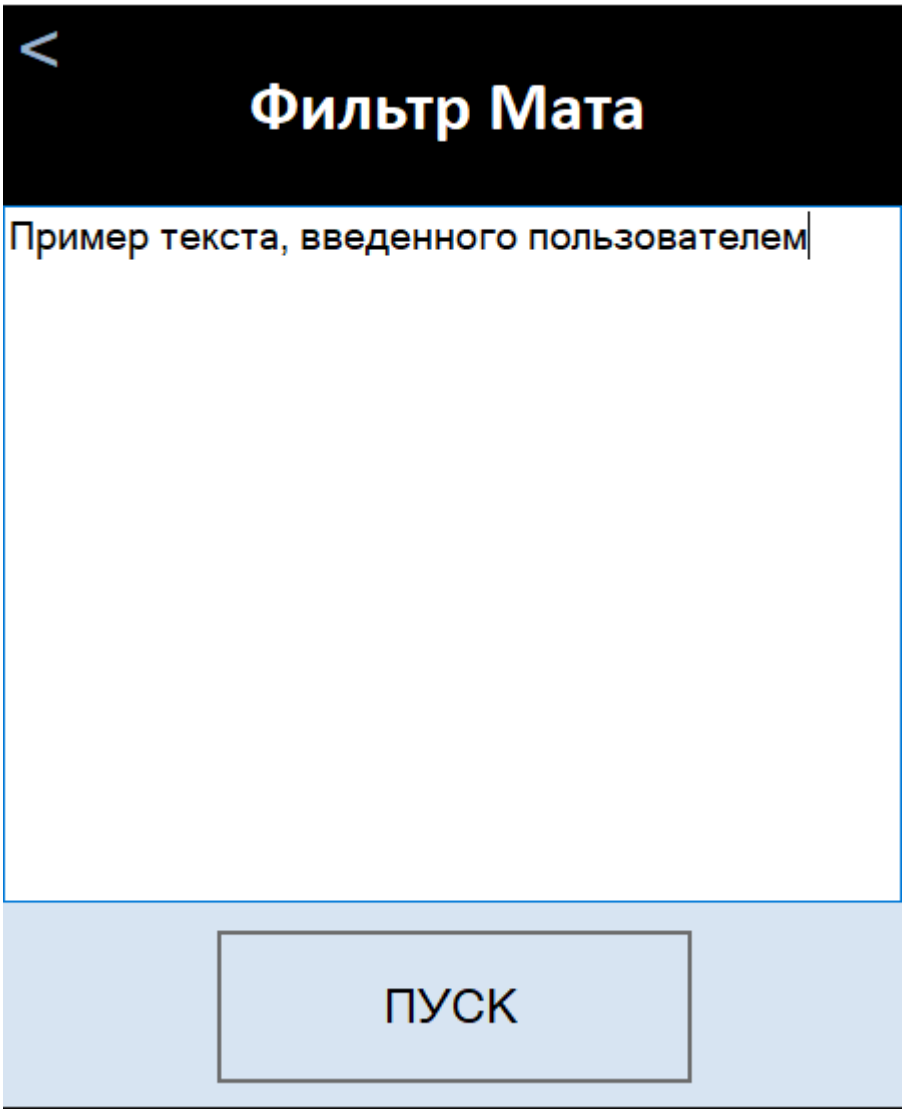

Рис. 2. Клавиатурный режим ввода текста в КС

2. Предоставление программе текста с помощью файла, содержащего данные, требующие обработки (см. рис. 3).

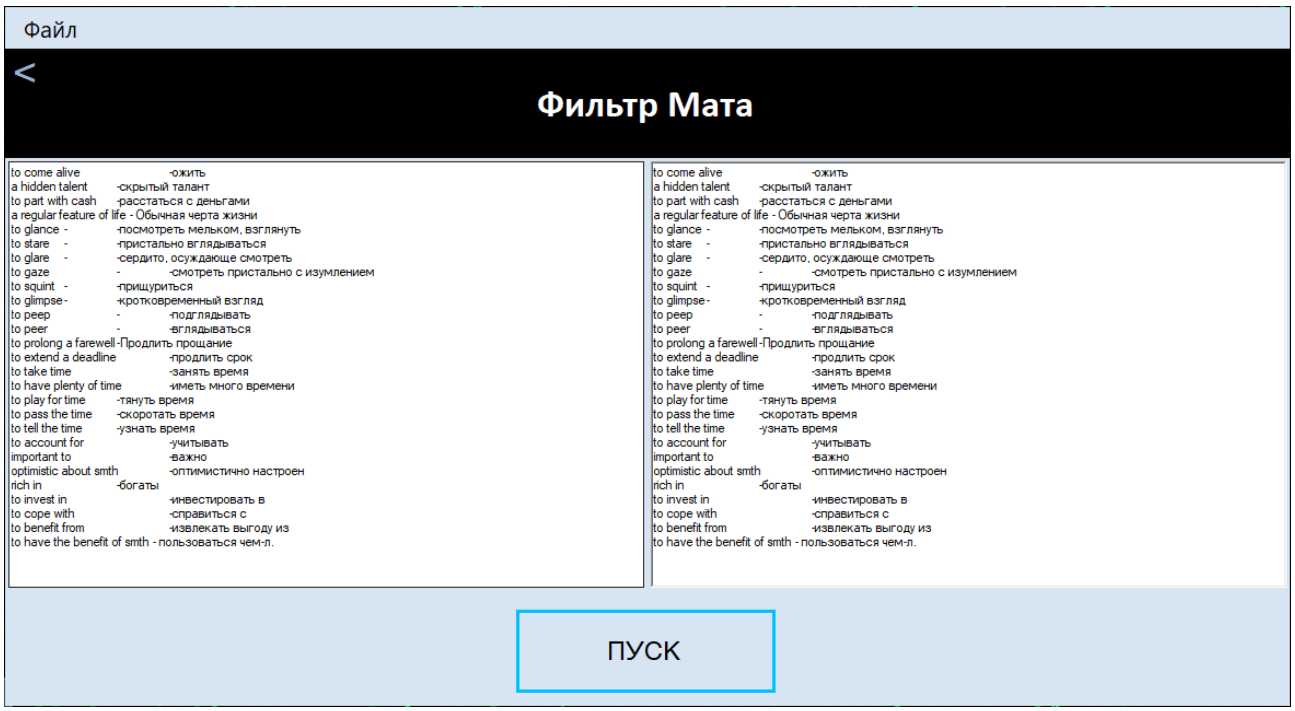

Рис. 3. Обработка файла

В данном режиме необходимо выбрать файл с расширением ”.txt” (см. рис. 4):

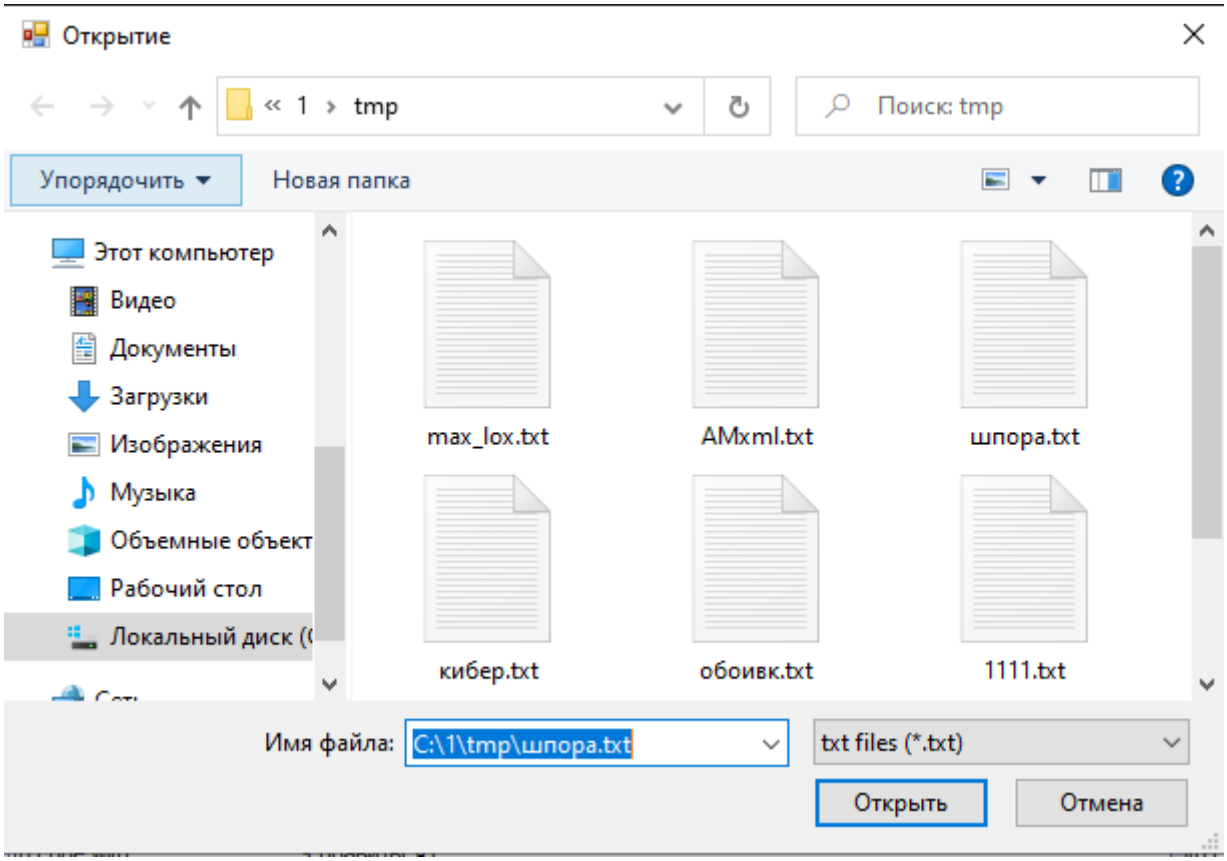

Рис. 4. Диалоговое окно

Если пользователь не выбрал файл, то ему выводится сообщение об ошибке (см. рис. 5).

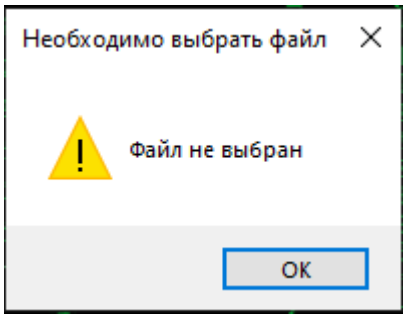

Рис. 5. Ошибка «Файл не выбран»

3. Предоставление программе текста с помощью фотографии, содержащего данные, требующие обработки (см. рис. 6).

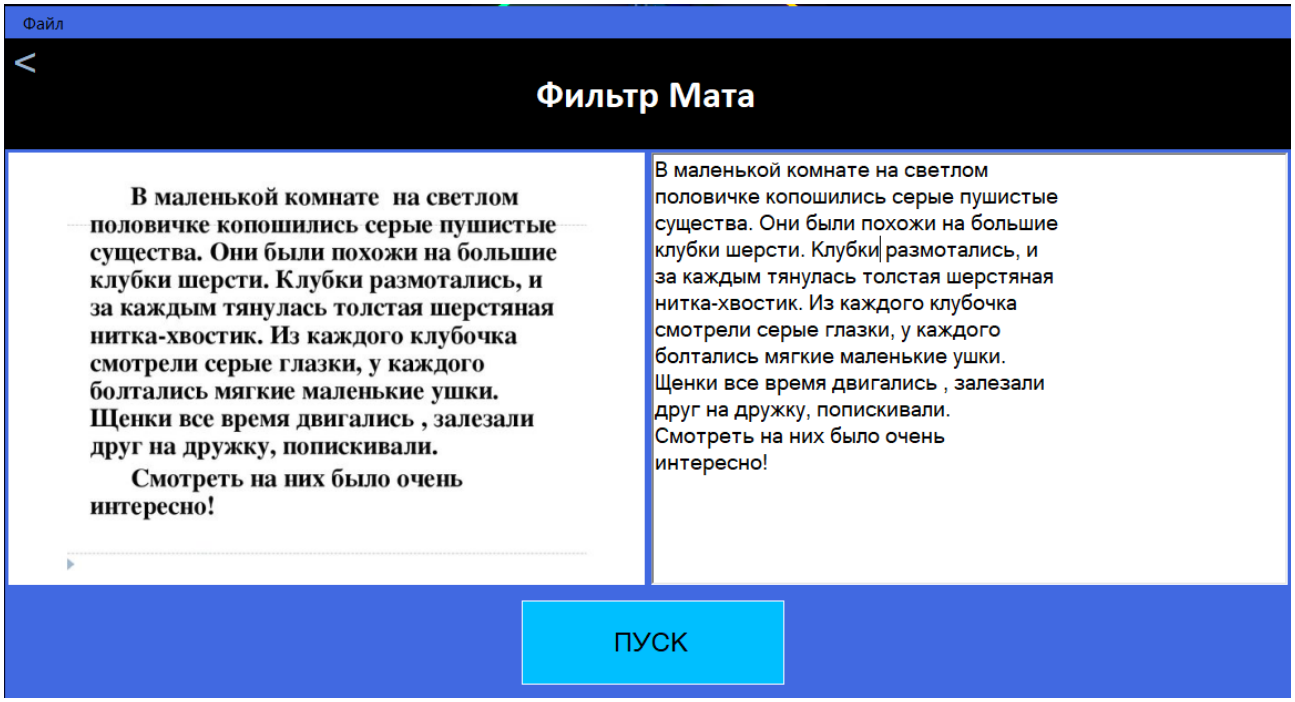

Рис. 6. Обработка фотографии

В данном способе программе распознает по картинке русский текст и фильтрует его, аналогично предыдущим пунктам. Если пользователь не выбрал фотографию, то ему выводится сообщение об ошибке (см. рис. 7)

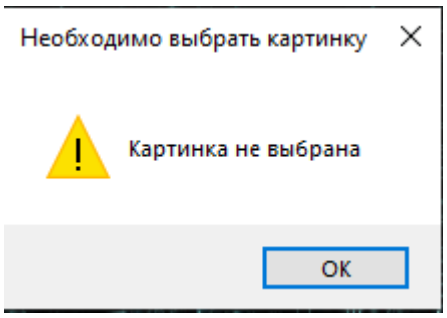

Рис. 7. Ошибка «Картинка не выбрана»

4. Предоставление программе текста с помощью снимка экрана, содержащего данные, требующие обработки. Работает данная функция, аналогично предыдущей команде. (см. рис. 8).

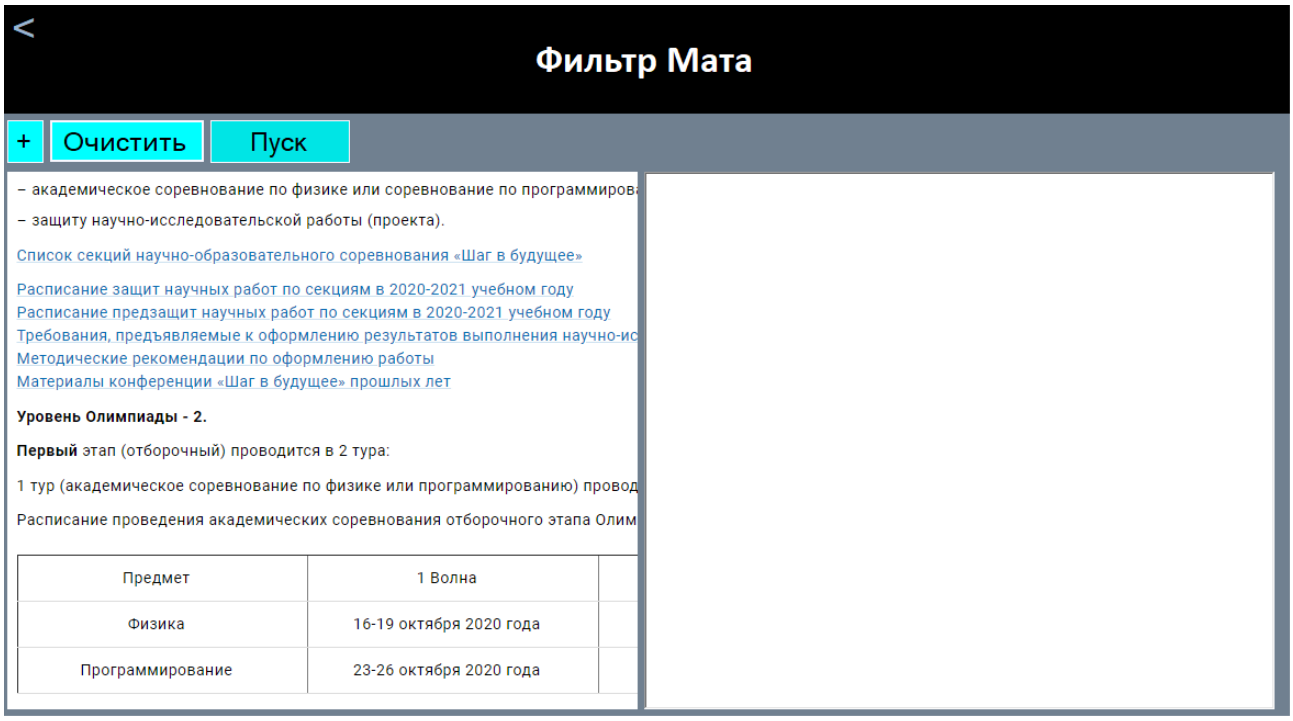

Рис. 8. Вид 4 функции приложения

При нажатии кнопки “+” создается снимок экрана. Кнопка “Очистить” удаляет созданное изображение. “ПУСК” сканирует текст и выдает результат текстовое поле (см. рис. 9).

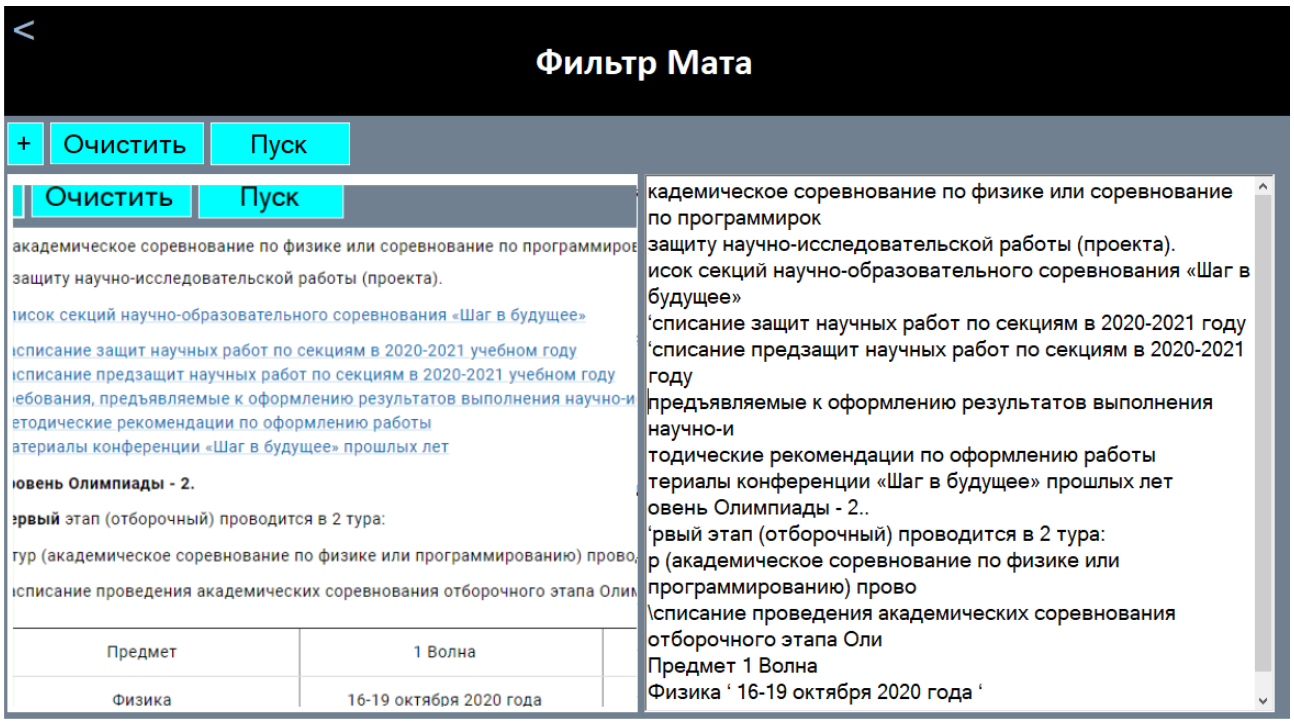

Рис. 9. Обработка снимка экрана

По нажатию на знак вопроса в левом верхнем углу главного меню происходит открытие с помощью вызова диалогового окна справки программы, представляющей собой краткое описание функций Компьютерной системы, а также информацию о разработчике. После чего пользователь может продолжить работу с приложением. (см. рис. 10)

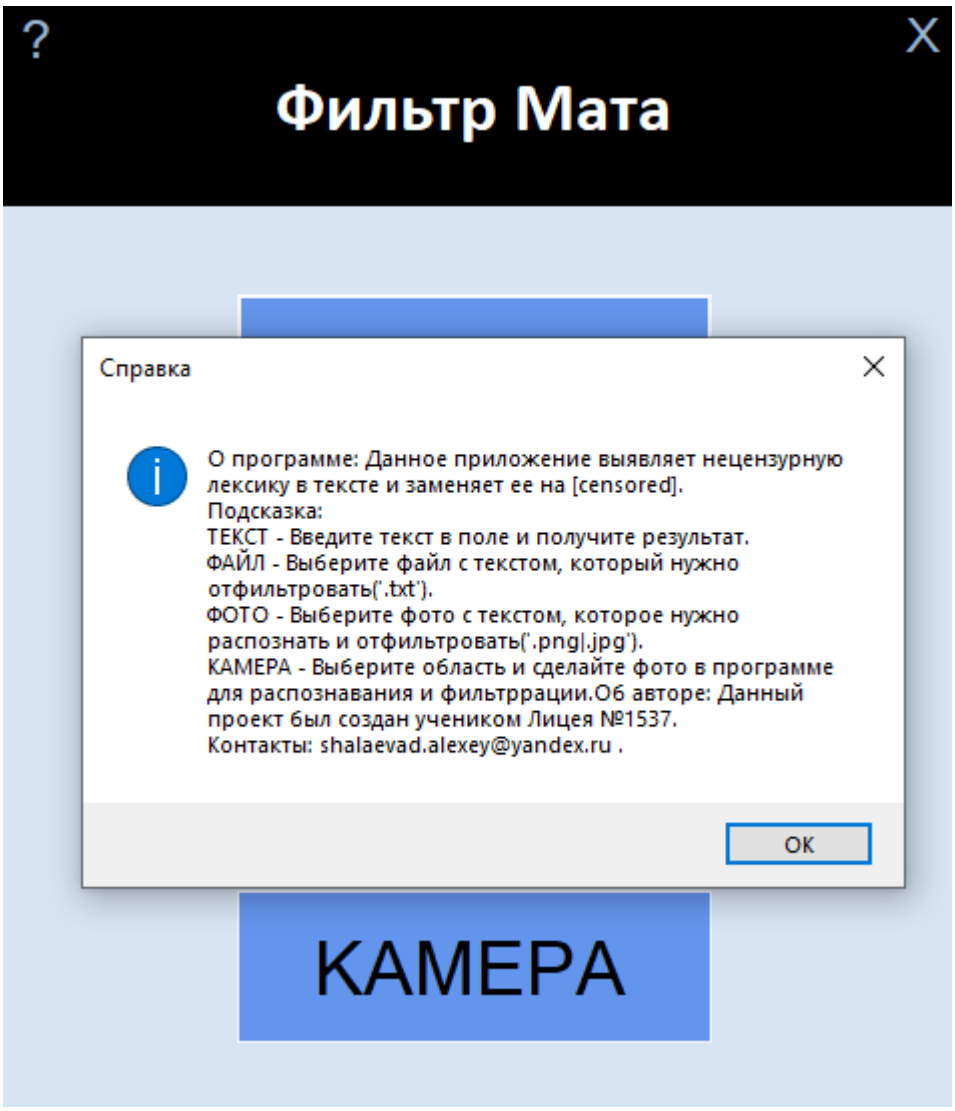

Рис. 10. Справка о программе

ЗАКЛЮЧЕНИЕ

Реализованный программный продукт выявляет и убирает из текста нецензурную лексику.

При дальнейшей доработке компьютерной системы предполагается добавление следующих дополнительных функций:

- реализация новой команды, позволяющей добавлять и редактировать словарь нецензурной лексики, а также кастомизировать (изменять размеры шрифтов и тому подобное) приложение под себя;
- внедрение Лемматизатора для улучшения качества фильрации текста и увеличение библиотеки нецензурной лексики;
- программная реализация фильтрации текста, содержащего не только нецензурную лексику, но и содержащего пропаганду употребления алкоголя, наркотиков, призыва к суицидам, порнографию и так далее;
- создание бота для фильтрации чатов участников в наиболее популярных мессенджерах (например, платформа Discord).

Разработанная компьютерная система может найти практическое применение в СМИ и различных интернет-ресурсах. Кроме того, данная программа призвана способствовать уменьшению «загрязнения» русского языка и снижению использования нецензурной лексики в речи молодежи. Разработки компьютерной системы могут быть использованы в соцсетях для решения данной проблемы.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон от 30.12.2020 № 530-ФЗ "О внесении изменений в Федеральный закон "Об информации, информационных технологиях и о защите информации"
   URL: https://away.vk.com/away.php
2. Ilya Segalovich A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine, 2003
   URL: https://cache-mskmar02.cdn.yandex.net/download.yandex.ru/company/iseg-las-vegas.pdf
3. Jurafsky, D. Speech and Language Processing / D. Jurafsky, J. H. Martin. – 2nd – New Jersey: Prentice Hall, 2008. – 1024 p.
4. Шалак В.И. Современный контент-анализ. – М.: Омега-Л, 2009
5. Автоматическая обработка текстов на естественном языке компьютерная лингвистика: учеб. пособие /Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В.— М.: МИЭМ, 2011
6. Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017.
7. Russian stemming algorithm [Электронный ресурс]
   URL: http://snowball.tartarus.org/algorithms/russian/stemmer.html
8. Regular expressions library [Электронный ресурс]
   URL: https://en.cppreference.com/w/cpp/regex
9. Список нецензурной лексики.
   URL: https://yadi.sk/d/McrzPrrcj3hd7