**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА ИУ7. Программное обеспечение ЭВМ и информационные технологии

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***НА ТЕМУ:***

***Метод определения коэффициента смысловой***

***схожести двух текстов***

Подготовил: Попов Ю.А.

2023 г.

**Содержание**

[Введение](#_qxjh5fn790f4)

[1 Аналитическая часть](#_fwrqrwfj07xj)

[1.1 Существующие аналоги](#_q18qbf43ykdf)

[1.2 Особенности работы с текстами](#_ij7hrp8888eb)

[1.3 Требования к разрабатываемому методу](#_ognqtwaho641)

[2 Конструкторская часть](#_lgitrvo9edtz)

[2.1 Нейросетевые языковые модели](#_z1f4eim373m3)

[2.2 Векторное представление слова](#_brtxvyd2yskd)

[2.3 Алгоритм работы программы](#_fdqcd53d2evb)

[2.4 Выбор модели](#_w3rkgrgy1xho)

[2.5 Word2vec](#_863sqfvm9lew)

[3. Технологическая часть](#_4m8dc0o1b4i2)

[3.1 Выбор средств программной реализации](#_ewwwbk8wnklr)

[3.2 Применение технологии API](#_kfjxp0cqugun)

[3.3 Взаимодействие с word2vec](#_uh4fny4jb2dh)

[3.4 Сайт с моделями rusvectores](#_780o9skvctrm)

[3.5 Взаимодействие с пользователем](#_jtwc74kg8a7k)

[3.6 Демонстрация функциональности](#_xjo464c6ijze)

[4. Исследовательская часть](#_351htz1u9r41)

[Заключение](#_8m1jzif5xy7r)

[Среди направлений дальнейшего развития выделены:](#_k8ly5ricjuoo)

[Список источников](#_3suje8f54hnm)

# **Введение**

В наше время люди сталкиваются с огромным количеством текстовой информации. Довольно часто одни и те же события описываются с использованием различных формулировок, что делает актуальной проблему распознавания плагиата. Ручная обработка занимает много времени, поэтому является проблематичнойю Данная работа может быть оптимизирована с использованием программного обеспечения. В связи с этим было принято решение разработать соответствующее программное средство.

Цель работы – разработать и программно реализовать метод определения коэффициента смысловой схожести двух текстов. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести сравнительный анализ предметной области и существующих методов сравнения текстов.
2. Разработать метод определения коэффициента смысловой схожести двух текстов.
3. Программно реализовать предложенный метод.
4. Исследовать характеристики разработанного программного обеспечения.

# **1 Аналитическая часть**

В современном мире происходят ситуации, в которых необходимо сравнить два текста между собой. Для оптимального анализа текстовой информации можно использовать программное обеспечение. Для того, чтобы определить особенности, которыми должно обладать программное обеспечение сравнения текстов, необходимо рассмотреть существующие аналоги и выявить их достоинства и недостатки.

## **1.1 Существующие аналоги**

Среди программных средств для сравнения текстов есть сервисы «Панда-Копирайтинг»[1] , «Prostudio»[2], а также «Text.ru»[3].

Сервис «Панда-Копирайтинг» предлагает сравнить тексты, узнать количество слов, символов и различающихся символов. Среди дополнительных функций пользователь может воспользоваться инструментами для очистки, создания и преобразования текста, такими как генератор заголовков, число прописью, смена регистра, зачеркнутый текст и шаблоны заголовков. Сервис сравнивает тексты по наличию одинаковых символов и слов. В случае замены слов на их синонимы данный подход некорректно оценивает уникальность, что является функциональным недостатком. Интерфейс сервиса “Панда-Копирайтинг” представлен на рисунке 1.

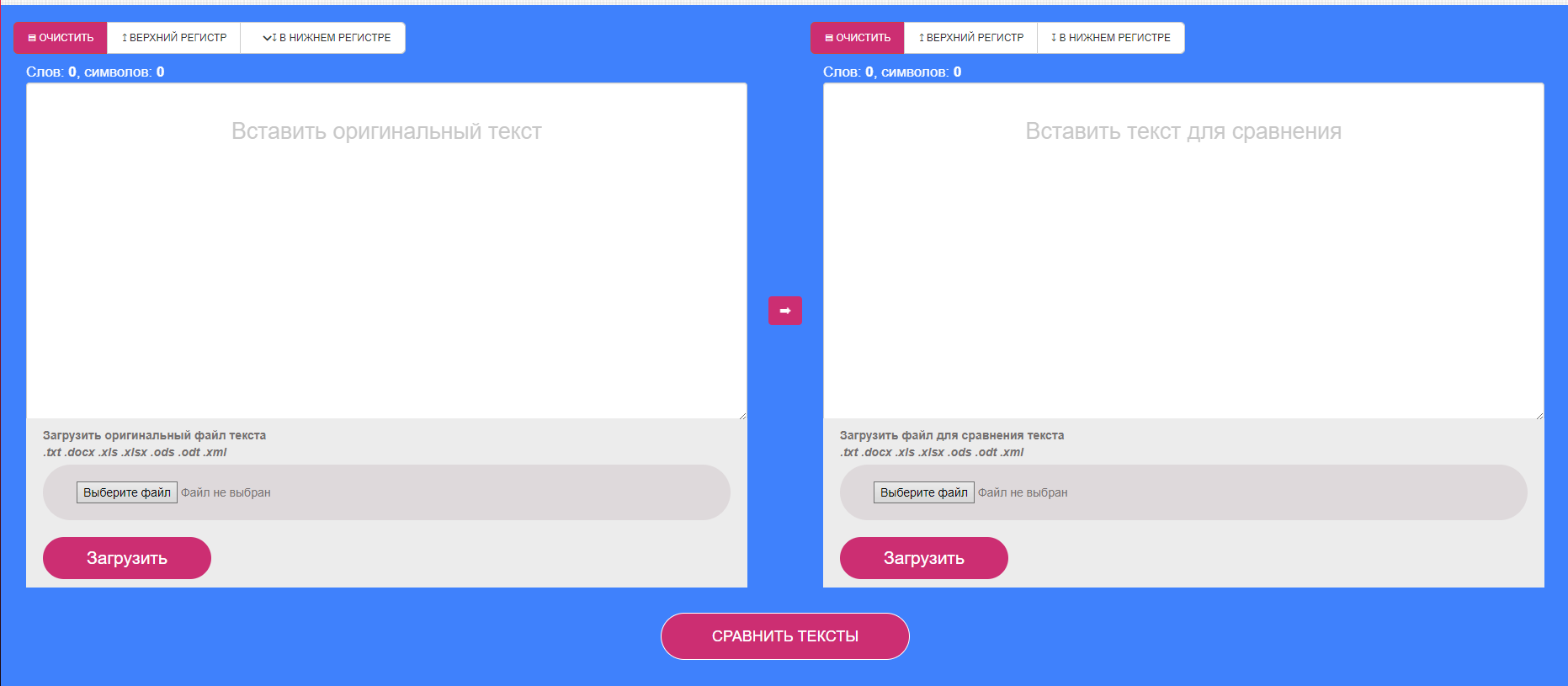


Рис. 1. Страница сравнения текстов сервиса «Панда-Копирайтинг».

Сервис «Prostudio» представляет собой вспомогательную утилиту в рамках веб-студии, с помощью которой можно определить сходство в двух текстах и узнать их “пересечения”. Но в то же время можно выделить невозможность узнать процент схожести текста. Сервис определяет совпадающие подстроки между двумя текстами, но обойти его можно заменой слов на синонимы. Интерфейс сервиса “Prostudio” представлен на рисунке 2.

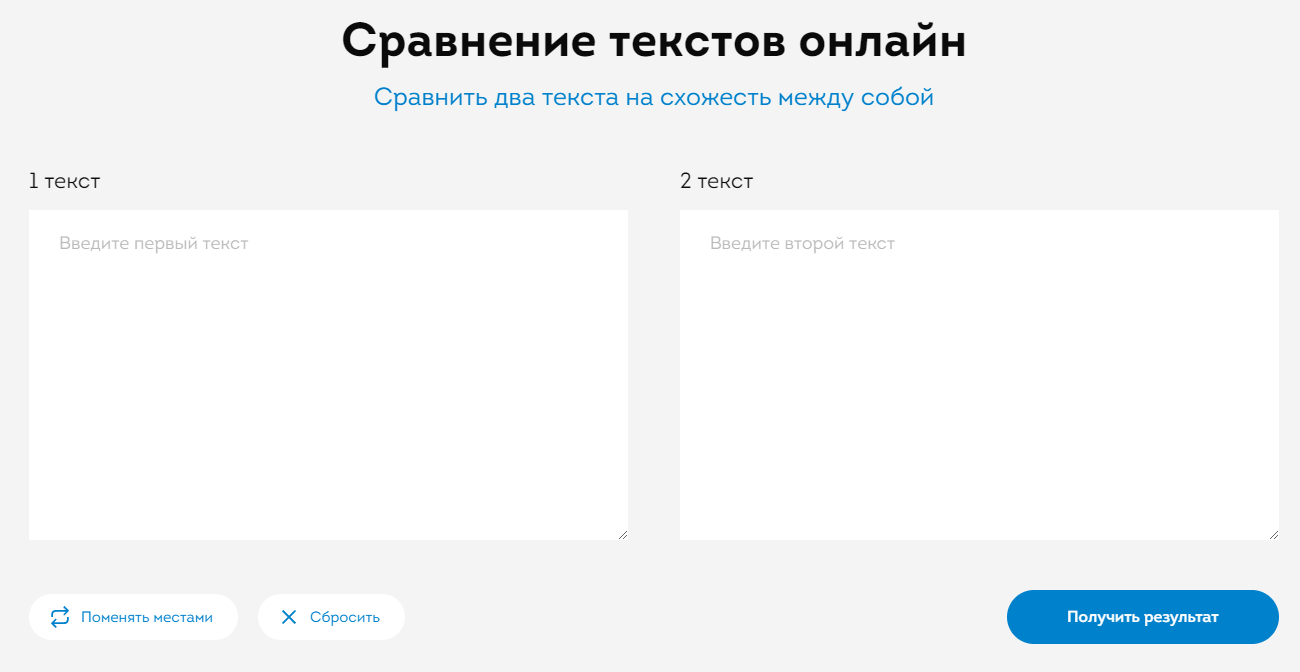


Рис. 2. Страница сравнения текстов сервиса «Prostudio».

Сервис «Text.ru» предоставляет пользователю возможность проверить текст на плагиат среди Интернет-ресурсов, найти его аналоги, проверить и дать краткую характеристику:

1. количество символов и слов;
2. уникальность;
3. заспамленность - это слишком частое повторение ключевого слова в тексте[4].

Сервис заточен на выявление одинаковых символов и подстрок, что снижает его точность. Интерфейс сервиса “Text.ru” представлен на рисунке 3.

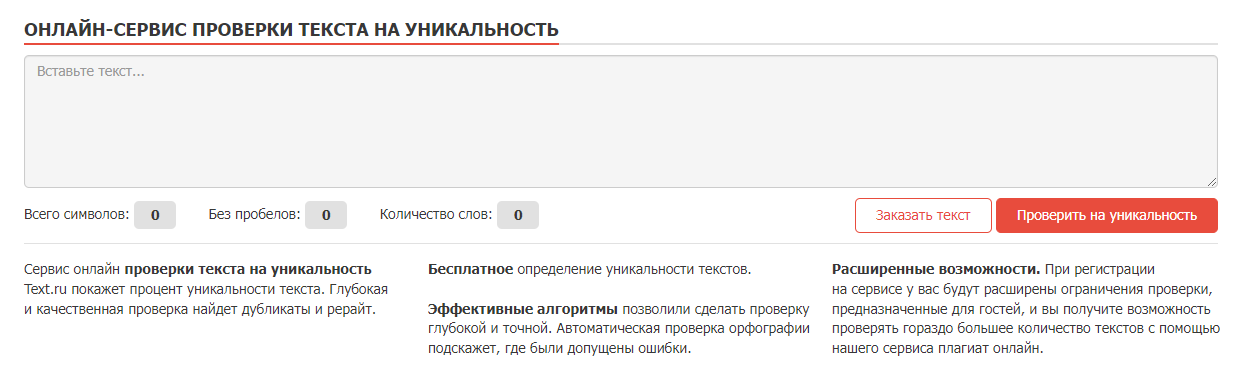


Рис. 3. Страница сравнения текстов сервиса «Text.ru».

Сравнение сервисов представлено в таблице номер 1.

Способ сравнения — насколько корректно работает алгоритм сравнения текстов.

Функционал — насколько возможности сервисам соответствуют потребностям.

Таблица 1. Сравнение сервисов определения идентичности двух текстов.

| **Название сервиса** | **Способ сравнения** | **Функционал** |
| --- | --- | --- |
| **Панда-Копирайтинг** | Сервис сравнивает тексты по наличию одинаковых символов и слов. | * Определение процента сходства; * подсчет количества символов в совпавших подстроках; * подсчет количества символов в несовпавших подстроках. |
| **Prostudio** | Сервис сравнивает тексты по наличию одинаковых словам и их частей. | * Результат сравнения; * число различий. |
| **Text.ru** | Сервис проверяет текст на плагиат. Поиск и сравнение ведется по сочетаниям символов. | * Уникальность; * проверка орфографии; * подсчет количества символов; * подсчет количества слов; * заспамленность. |

## **1.2 Особенности работы с текстами**

Два одинаковых по смыслу текста могут быть записаны с использованием различных слов или формулировок. Заменяя слова на контекстные синонимы и фразеологизмы, изменяя порядок предложений и перефразируя фрагменты, можно получить уникальный текст, хотя смысл останется тот же. Большинство сервисов конкурентов объединяет существенный недостаток: они не способны отличить документ, прошедший через “синонимайзер”. Если переписать предложения, используя сложную лексику и обороты, то эффективность существующих алгоритмов стремится к нулю. Например, если заменить слова на синонимы и запустить сервис-аналог «Панда-Копирайтинг», то процент сходства будет около 30%.

Таблица 2. Входные данные.

| **Текст 1** | **Текст 2** |
| --- | --- |
| Дядя Семён ехал из города домой. С ним была собака Жучка, Вдруг из леса выскочили волки. Жучка испугалась и прыгнула в сани. У дяди Семёна была хорошая лошадь. Она тоже испугалась и быстро помчалась по дороге. Деревня была близко. Показались огни в окнах. Волки отстали. | Человек Семён двигался изо мегаполиса домой. С ним водилась дворняжка Жучка, Недуманно-негаданно из леса вывернулись волки. Воровка перепугалась и прыгнула в сани. У дяди Семёна водилась превосходная лошадь. Она тоже перепугалась и проворно помчалась по дороге. Деревушка водилась близко. Проклюнулись огни в окнах. Волки отстали. |

Сравнение текстов, приведенных в таблице 2, сервисом «Панда-Копирайтинг»:

* процент сходства - 32.31;
* количество символов в совпавших подстроках - 147;
* количество символов в несовпавших подстроках - 308.

Сравнение текстов, приведенных в таблице 2, сервисом «Prostudio»:

* результат сравнения - “Тексты разные”;
* число различий - 28.

Сравнение 1 текста, приведенного в таблице 2, сервисом «Text.ru»:

* уникальность - 0%;
* проверка орфографии - “В данном тексте ошибки не найдены”;
* подсчет количества символов - 270;
* подсчет количества слов - 45;
* заспамленность - 34%.

Сравнение текста 2, приведенных в таблице 2, сервисом «Text.ru»:

* уникальность - 59.06%;
* проверка орфографии - “В тексте найдены 2 ошибки”;
* подсчет количества символов - 330;
* подсчет количества слов - 45;
* заспамленность - 38%.

Если рассмотреть вышеприведенный пример и изучить результат работы существующих аналогов, то можно прийти к выводу, что они не могут определить текст, прошедший через замену слов на синонимы. Из-за данного недостатка возникает необходимость в разработке нового программного обеспечения.

## **1.3 Требования к разрабатываемому методу**

В результате проведенного анализа можно сформулировать требования, предъявляемые к методу определения коэффициента смысловой схожести двух текстов. Разрабатываемый метод должен:

* определять коэффициент схожести двух текстов;
* не зависеть от порядка слов в предложении;
* не зависеть от словоформ;
* иметь слабое влияние замены слов в текстах на синонимы, при вычислении коэффициента.

# 

# **2 Конструкторская часть**

Если немного изменить два текста путем перифраза либо заменой слов на синонимы, то для большинства алгоритмов они будут выглядеть совершенно по-разному, потому что количество “пересечений” будет невелико. Для более точного анализа текста необходимо использовать нейросети, которые сейчас находятся на пике своей популярности.[5]

## **2.1 Нейросетевые языковые модели**

Нейронная сеть — это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса машинного обучения, называемый глубоким обучением, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг. Он создает адаптивную систему, с помощью которой компьютеры учатся на своих ошибках и постоянно совершенствуются. Таким образом, искусственные нейронные сети пытаются решать сложные задачи, такие как резюмирование документов или распознавание лиц, с более высокой точностью. [6]

Нейросетевые языковые модели — это большие нейронные сети, которые обучаются предсказывать следующее слово в тексте с учетом предыдущего контекста. Несмотря на кажущуюся простоту задачи, оказалось, что такая постановка задачи приводит к появлению весьма многофункциональной нейронной сети. [7]

Нейросетевые модели испытывают сложности при необходимости работы с разреженными категориальными признаками. Эмбеддинги — это возможность уменьшения размерности таких признаков ради повышения производительности модели. В сфере обработки естественных языков (Natural Language Processing, NLP) обычно работают со словарями, состоящими из тысяч слов. Эти словари обычно вводят в модель с использованием методики быстрого кодирования (One-Hot Encoding), что, в математическом смысле, равносильно наличию отдельного столбца для каждого из слов. Когда слово передается в модель, в соответствующем столбце оказывается единица, в то время как во всех остальных выводятся нули. Это ведет к появлению очень сильно разреженных наборов данных. Решение этой проблемы заключается в создании эмбеддинга. Эмбеддинг, на основе обучающего текста, группирует слова со сходным значением и возвращает их местоположение в векторной системе координат. Например, значение эмбеддинга слова «fun» может быть подобно значениям эмбеддингов слов «humor» и «dancing», или словосочетания «machine learning». На практике нейронные сети демонстрируют значительное улучшение производительности при применении подобных репрезентативных свойств. [8]

## **2.2 Векторное представление слова**

Векторное представление слов — это один из самых популярных сегодня методов автоматической обработки языка. Основано оно на том, что все слова языка можно представить в виде векторов в многомерном пространстве определенной общности текстов этого же языка. Слова, которые в таком пространстве находятся ближе друг к другу, в языке будут схожи по своей семантике: например, слово «король» будет ближе к слову «мужчина», чем к слову «женщина».

При этом векторное представление достаточно контекстно-зависимо, и то, как будут располагаться слова в пространстве, сильно зависит от величины выборки используемых текстов с одной стороны, а с другой — от определенной их тематики. Иногда это ограничение модели используют, например, для того, чтобы выделить из большого корпуса специфических текстов какую-то важную информацию.[9]

## **2.3 Описание функционирования системы**

Cosine similarity[10] — это значение, которое равно косинусу между векторами, полученных в результате обработки двух текстов.

Процесс обработки начинается с очистки текста от частиц, междометий, предлогов и знаков препинания. Это необходимо для более точного распознавания слов. После очистки, начинается процесс обработки текста. С помощью нейросети, каждое слово представляется как вектор, а после этого добавляется к общему вектору. После анализа всех слов, общий вектор делится на число распознанных слов, и получается итоговый результат обработки текста. Два полученных результата сравниваются между собой методом Cosine similarity. Результат работы метода через API передается в клиентскую часть и выводится на экран (рис. 4).



Рис. 4. Функциональная схема работы метода в нотации IDEF0.

Взаимодействие пользователя с методом представлено на диаграмме USECASE (рис. 5).

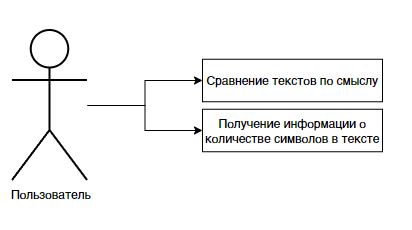


Рис. 5. Диаграмма USECASE.

Порядок работы программного обеспечения с данными продемонстрирован на рисунке 6

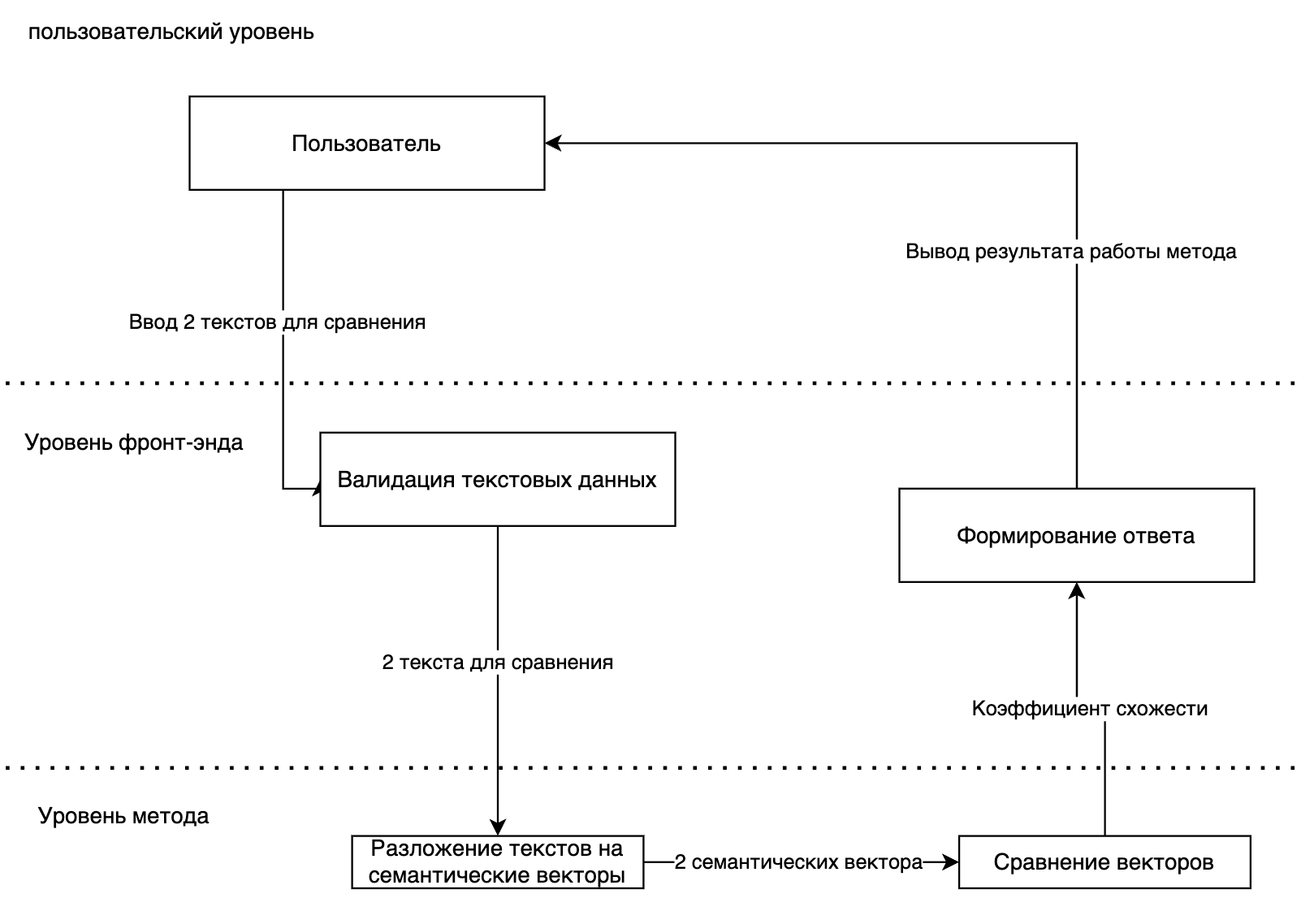


Рис. 6. Диаграмма потоков данных.

## **2.4 Выбор модели**

На данный момент для свободного использования доступны несколько языковых моделей, таких как: RNNLM[11], Word2vec[9], GloVe[11] и fastText[12]. Сравнение моделей представлено в таблице номер 3.

Мешок слов – это модель представления текста в виде вектора. Каждому слову сопоставляется число его вхождений в текст. [11]

Skip-Gram – Формируется «контекстное окно» – последовательность из k слов в тексте. Одно из этих слов пропускается, и нейросеть пытается его предсказать. Таким образом, слова, которые часто встречаются в похожем контексте, будут иметь похожие векторы. [11]

Негативное семплирование (Negative Sampling) – Многие слова в текстах не встречаются вместе, поэтому модель выполняет много лишних вычислений. Подход Negative Sampling позволяет максимизировать вероятность встречи нужного слова в контексте, который является для него типичным, и минимизировать – в нетипичном контексте. [11]

n-грамм – каждое слово представляется композицией нескольких последовательностей символов определённой длины.

Таблица 3. Сравнение языковых моделей. [11]

| **Название модели** | **Преимущества** | **Недостатки** | **Используемые технологии** |
| --- | --- | --- | --- |
| Рекуррентная нейросетевая языковая модель (RNNLM) | * Модель быстро обучается и генерирует эмбеддинги * Предварительно обученные версии доступны на многих языках. | * Не учитывает долгосрочные зависимости; * Недостаточная скорость по сравнению с новыми embeddings моделями; | * Одна из первых эмбеддинг моделей; * Модель принимает на вход векторные представления предыдущих слов и может анализировать семантику предложения. Обучение модели базируется на алгоритме непрерывного мешка слов. |
| Word2vec | * Модель быстро обучается и генерирует эмбеддинги. | * Обучение на уровне слов: нет информации о предложении или контексте, в котором используется слово; * Совместная встречаемость игнорируется. Модель не учитывает то, что слово может иметь различное значение в зависимости от контекста использования; | * Вместо алгоритма непрерывного мешка слов модель Word2Vec использует Skip-gram; * Чтобы сделать обучение эффективнее, используется негативное семплирование; |
| GloVe (Global Vectors) | * Простая архитектура без нейронной сети; * Высокая скорость работы; * Осмысленные эмбеддинги. | * GloVe остаётся обученной на уровне слов и дает немного данных о предложении и контексте, в котором слово используется; * Плохо обрабатывает неизвестные и редкие слова; | * GloVe учитывает совместную встречаемость, а не полагается только на контекстную статистику. Векторы слов группируются вместе на основе их глобальной схожести. |
| fastText | * Благодаря n-граммам работает с редкими и устаревшими словами. | * Обучение на уровне слов; * Модель игнорирует совместную встречаемость. | * Для векторизации слов используются одновременно и skip-gram, и негативное семплирование, и алгоритм непрерывного мешка.; * использование n-граммов. |

## **2.5 Word2vec**

В 2013 году Томас Миколов из Google предложил более эффективную модель обучения векторных представлений слов – Word2vec. Метод основывался на предположении, что слова, которые часто находятся в одинаковых контекстах, имеют схожие значения. Изменения были просты – устранение скрытого слоя и аппроксимация (упрощение) цели – но стали поворотной точкой в развитии языковых моделей NLP.

Вместо алгоритма непрерывного мешка слов модель Word2Vec использует Skip-gram (словосочетание с пропуском). Цель этой модели прямо противоположная предыдущей модели – предсказать окружающие слова на основе центрального[11]. Использование модели word2vec дает несколько новых возможностей по сравнению со стандартными методами:

* возможность разложить слово на вектор, для последующего сравнения (рисунок 7);
* высокая скорость работы, по сравнению с обычными методами;
* возможность получить «семантическое расстояние» между словами;
* модель быстро обучается и генерирует эмбеддинги.

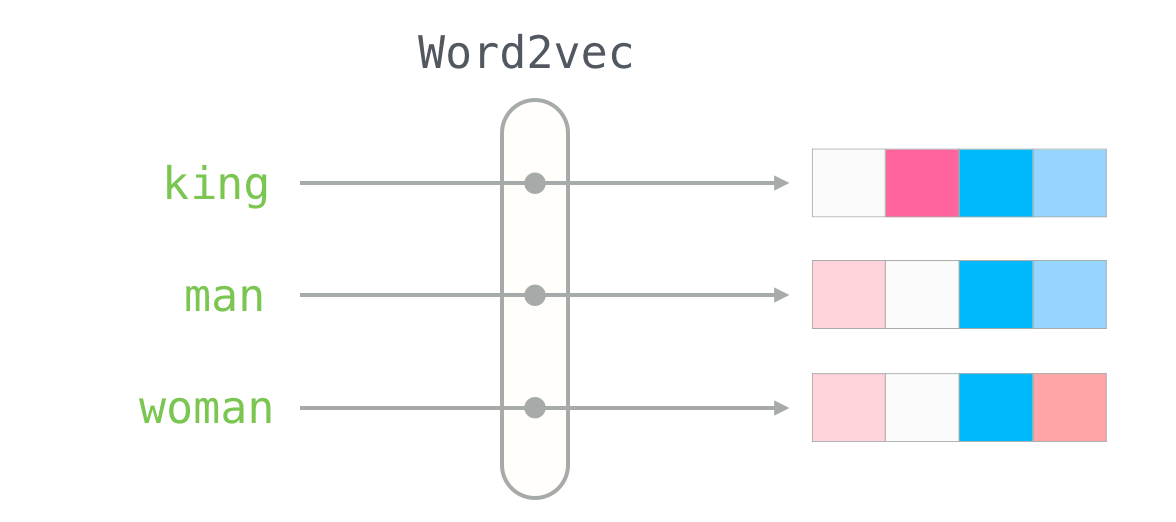


Рис. 7. Векторное представление слова[9]

# **3. Технологическая часть**

## **3.1 Выбор средств программной реализации**

Средством реализации алгоритма данного программного обеспечения был выбран язык python 3.10[13] в связи с обширным количеством библиотек для работы с предварительной обработкой данных и машинным обучением, такими как TensorFlow[14], Keras[15], Pandas[16], wordnet[17], pymorphy[18], nltk[19]. Клиентская часть была написана на HTML[20] и CSS[21]. Для написания скриптов использовался Brython[22] - это реализация Python, написанная на JavaScript, которая позволяет писать код на Python для веб разработки.

## **3.2 Применение технологии API**

API (Application Programming Interface) — это совокупность инструментов и функций в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой. Это позволяет разработчикам расширять функциональность своего продукта и связывать его с другими. Большинство крупных компаний разрабатывают API для сторонних разработчиков.

В проекте Api был использован для связи серверной и клиентской частей. При нажатии пользователем кнопки отправляется запрос к методу /api/result, который обрабатывает входные данные, получает ответ и передает его. В результате работы этого метода программа выводит на экран схожесть текстов в процентах.

## **3.3 Взаимодействие с word2vec**

Тематическая модель — модель коллекции текстовых документов, которая определяет, к каким темам относится каждый документ коллекции. [23]

Тематическое моделирование — построение тематической модели.[23]

Для корректной работы Программное обеспечение было решено использовать библиотеку Gensim. Gensim[24] — это библиотека обработки естественного языка предназначения для «Тематического моделирования». С его помощью можно обрабатывать тексты, работать с векторными моделями слов и создавать тематические модели текстов. Для работы необходимо было найти обученную модель или натренировать ее самому. Была выбрана модель “news\_upos\_skipgram\_300\_5\_2019”[25]. Данная модель была обучена на новостях и содержит 249000 записей.

## **3.4 Сайт с моделями rusvectores**

Сервис RusVectōrēs вычисляет семантические отношения между словами русского языка и позволяет скачать предобученные модели, в том числе контекстуализированные. Он назван по аналогии с RusCorpora, веб-сайтом Национального Корпуса Русского Языка (НКРЯ). На RusCorpora можно работать с корпусами, а на RusVectōrēs — с лексическими векторами. Они представляют значение слова, автоматически извлеченное из статистики совместной встречаемости слов в корпусах (больших коллекциях текстовых данных). [25]

## **3.5 Взаимодействие с пользователем**

Для начала работы с программой пользователь заходит на сайт и вводит два текста для анализа. Для запуска обработки необходимо нажать на кнопку. После этого программа выведет сходство текстов, а также количество символов и слов в них. (рис. 6)



Рис. 6. Страница сравнения текстов.

Если программа получает неожиданный вход, например, пользователь не ввел текст в одно из текстовых полей, или ни одно из слов, введенных пользователем, не было распознано, то программа выведет ошибку. (рис. 7)

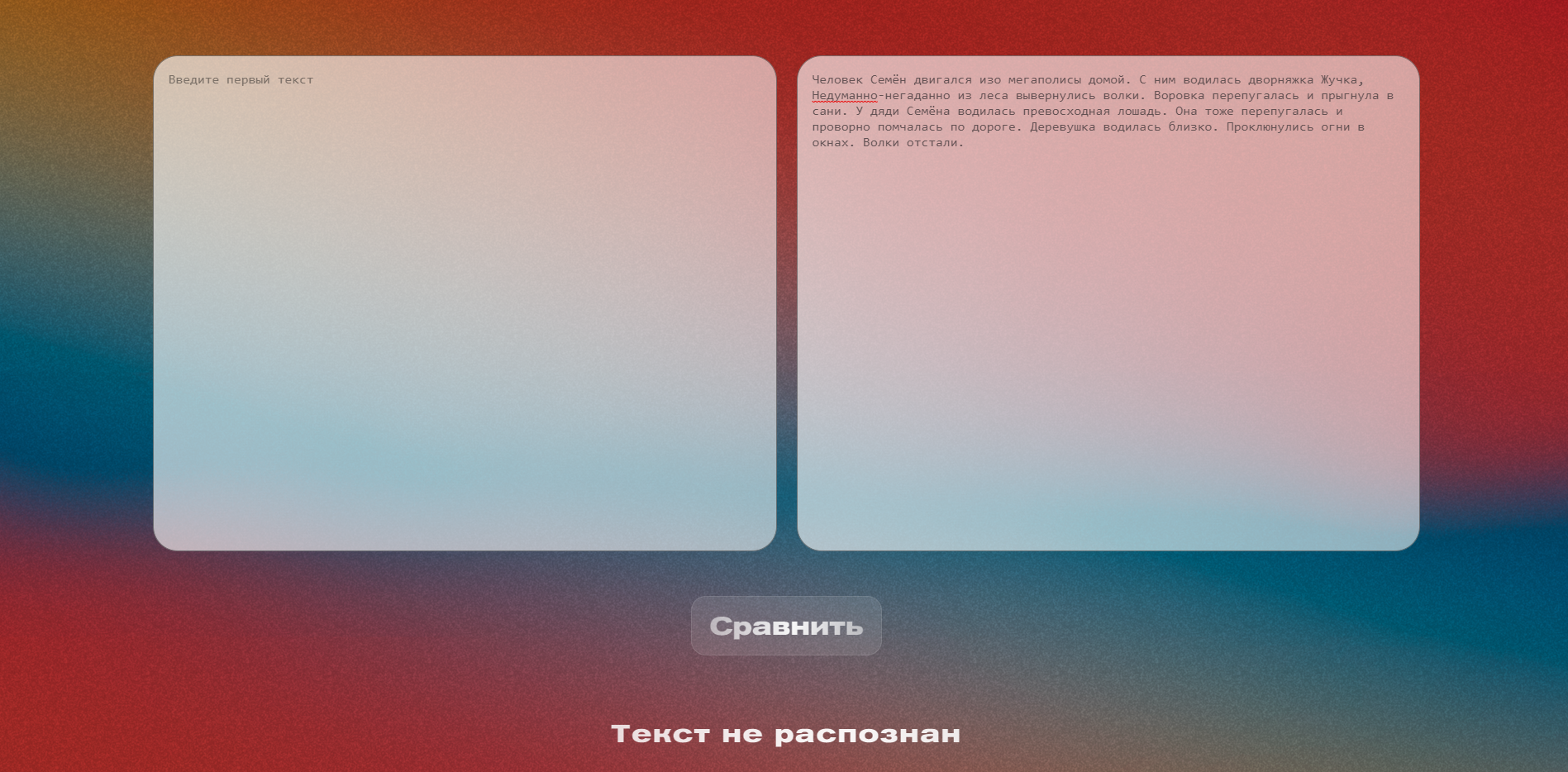


Рис. 7 Ошибка при некорректном вводе.

## **3.6 Демонстрация функциональности**

Для визуализации полученных результатов работы метода, была сделана сравнительная таблица 4.

Таблица 4. Демонстрация работоспособности предложенного метода.

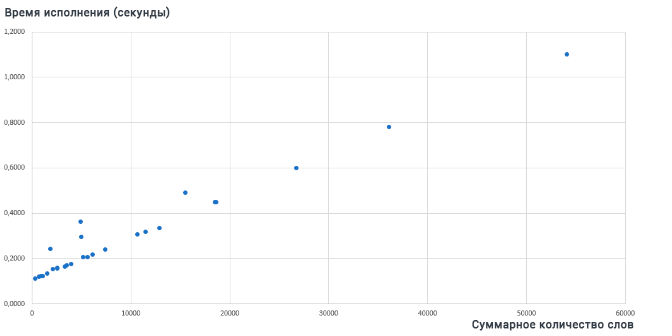
| **Текст 1** | **Текст 2** | **Результат** |
| --- | --- | --- |
| Дядя Семён ехал из города домой. С ним была собака Жучка, Вдруг из леса выскочили волки. Жучка испугалась и прыгнула в сани. У дяди Семёна была хорошая лошадь. Она тоже испугалась и быстро помчалась по дороге. Деревня была близко. Показались огни в окнах. Волки отстали. | Человек Семён двигался изо мегаполиса домой. С ним водилась дворняжка Жучка, Недуманно-негаданно из леса вывернулись волки. Воровка перепугалась и прыгнула в сани. У дяди Семёна водилась превосходная лошадь. Она тоже перепугалась и проворно помчалась по дороге. Деревушка водилась близко. Проклюнулись огни в окнах. Волки отстали. | **94.9%** |
| Отличный чат, спасибо, за время потраченное на вопросы и ответы, такие мамочки как я, очень Вам благодарны, в учёбу не лезу, но понимание что происходит имею очень хорошее | Замечательный разговор. Потраченного времени не жаль. Желаю вам всего хорошего. Учиться не буду. | **76.29%** |
| Замечательный разговор. Потраченного времени не жаль. Желаю вам всего хорошего. Учиться не буду. | Ужасно. Потраченного времени жаль. Спасибо всем за этот кирдык. | **45.95%** |
| Лондон столица Великобритании | Лондон столица Англии | **100.0%** |
| Лондон столица Великобритании | Москва столица России | **37.37%** |
| Дядя Семён ехал из города домой | С ним была собака Жучка | **38.92%** |
| В столице прошел дождь | В Москве будет дождь | **61.28%** |
| В Москве прошел дождь | В Москве будет дождь | **85.75%** |
| В Москве прошел ливень | В Москве будет дождь | **70.96%** |

Исходя из полученных результатов можно сделать вывод, что при результате более 75% тексты обладают общим смыслом. От 50% до 75% - тексты похожи, но могут содержать несогласованности или ошибки. При результате менее 50% нельзя однозначно сказать об их схожести. Однако стоит понимать, что автоматический анализ текста может быть ограниченным в своей точности и не всегда учитывать контекст и специфику использования текста. Например, он может не распознать сарказм или иронию, что приведет к снижению точности семантического анализа.[26]. Кроме того, автоматический анализ может допускать ошибки при распознавании грамматических и лексических конструкций, особенно при работе с редко встречающиеся словами, потому что их может не быть в модели.

Во время модульного тестирования выяснилось, что при анализе больших объемов текстов происходит снижение эффективности метода, связанное с тем, что большое количество одинаковых слов присутствует одновременно и в исходном тексте, и в тексте, с которым он сравнивается. Из-за этого общие вектора текстов становятся похожими между собой, хоть сами текста отличаются.

# **4. Исследовательская часть**

При увеличении размера входных данных не должно возникать экспоненциального роста времени обработки. С целью анализа производительности метода было произведено исследование зависимости времени работы реализованного метода от размера входных данных. (График 1)

График 1. Демонстрация скорости работы от объема входных данных

# 

# 

# **Заключение**

В результате проведённой работы было проведено аналитическое исследование существующих решений для определения схожести двух текстов. Было выявлено, что существующие решения не способны определить схожесть текстов на основе контекста. Большинство аналогов ищет совпадения символов, поэтому перестают работать при замене слов на синонимы. В результате анализа, был предложен собственный метод определения коэффициента схожести двух текстов. На основе предложенного метода было разработано программное обеспечение. При этом были решены следующие задачи:

1. Проведен сравнительный анализ предметной области и существующих методов сравнения текстов.
2. Разработан метод определения коэффициента смысловой схожести двух текстов.
3. Программно реализован предложенный метод.
4. Исследованы характеристики разработанного программного обеспечения.

Реализованное программное средство является прототипом, в связи с чем можно составить план дальнейших нововведений в программное обеспечение. Среди возможных направлений дальнейшего развития выделены:

* интеграция TF-IDF;
* снижение влияния объема текста на точность работы;
* разработка полноценного сервиса-антиплагиата, с возможностью индексирования интернет ресурсов;
* разработка расширенного API;
* разработка портативной версии в виде бота в Телеграмме.

# **Список источников**

1. Панда-Копирайтинг — Текст : электронный // : [сайт]. — URL: <https://petr-panda.ru/sravnvit-teksty/> (дата обращения: 18.01.2023).  
2. Prostudio — Текст : электронный // : [сайт]. — URL: <https://prostudio.ru/tools/compare-text/> (дата обращения: 18.01.2023).

3. Text.ru — Текст : электронный // : [сайт]. — URL: <https://text.ru/> (дата обращения: 18.01.2023).

4. Что такое заспамленность текста и как этого избежать? — Текст : электронный // : [сайт]. — URL: [https://www.rush-analytics.ru/blog/chto-takoe-zaspamlennost-v-tekste-i-kak-ee-izbezhat](https://www.rush-analytics.ru/blog/chto-takoe-zaspamlennost-v-tekste-i-kak-ee-izbezhat#:~:text=%D0%9C%D0%B5%D1%80%D1%8B%2C%20%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D1%8B%D0%B5%20%D0%B2%D1%80%D1%83%D1%87%D0%BD%D1%83%D1%8E%3A-,%D0%A7%D1%82%D0%BE%20%D1%82%D0%B0%D0%BA%D0%BE%D0%B5%20%D0%B7%D0%B0%D1%81%D0%BF%D0%B0%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C%2C%20%D0%B2%D0%BE%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%8C%2C%20%D1%82%D0%BE%D1%88%D0%BD%D0%BE%D1%82%D0%B0%20%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0%3F,%D0%97%D0%B0%D1%81%D0%BF%D0%B0%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C%20%D0%BD%D0%B5%20%D0%B4%D0%BE%D0%BB%D0%B6%D0%BD%D0%B0%20%D0%BF%D1%80%D0%B5%D0%B2%D1%8B%D1%88%D0%B0%D1%82%D1%8C%2060%25) (дата обращения: 16.01.2023).

5. Исследования особенностей развития нейронных сетей в современном мире — Текст : непосредственный // Технические науки: проблемы и перспективы : материалы IV Междунар. науч. конф. (г. Санкт-Петербург, июль 2016 г.). — Санкт-Петербург : Свое издательство, 2016. — С. 9-11. — URL: https://moluch.ru/conf/tech/archive/166/10748/ (дата обращения: 18.01.2023).

6. Нейронная сеть — Текст : электронный // : [сайт]. — URL: https://aws.amazon.com/ru/what-is/neural-network/ (дата обращения: 18.01.2023).

7. Нейросетевые языковые модели — Текст : электронный // : [сайт]. — URL: <https://habr.com/ru/post/506042/> (дата обращения: 18.01.2023).

8. Эмбеддинги признаков и повышение точности ML-моделей / Хабр — Текст : электронный // : [сайт]. — URL: <https://habr.com/ru/company/wunderfund/blog/590651/> (дата обращения: 02.03.2023).

8. Векторное представление слов научило компьютер материаловедению — Текст : электронный // : [сайт]. — URL: <https://nplus1.ru/news/2019/07/05/materials-embedding> (дата обращения 18.01.2023)

9. Word2vec в картинках — Текст : электронный // : [сайт]. — URL: https://habr.com/ru/post/446530/ (дата обращения 18.01.2023)

10. Как рассчитать косинусное сходство в Python — Текст : электронный // : [сайт]. — URL: <https://www.codecamp.ru/blog/cosine-similarity-python/> (дата обращения 18.01.2023)

11. Обзор четырёх популярных NLP-моделей — Текст : электронный // : [сайт]. — URL: <https://proglib.io/p/obzor-chetyreh-populyarnyh-nlp-modeley-2020-04-21> (дата обращения 18.01.2023)

12. fastText — Текст : электронный // : [сайт]. — URL: https://fasttext.cc/ (дата обращения 10.03.2023)

13. Welcome to Python.org — Текст : электронный // : [сайт]. — URL: https://www.python.org/ (дата обращения 10.03.2023)

14. TensorFlow — Текст : электронный // : [сайт]. — URL: https://www.tensorflow.org/?hl=ru (дата обращения 10.03.2023)

15. Keras: the Python deep learning API — Текст : электронный // : [сайт]. — URL: https://keras.io/ (дата обращения 10.03.2023)

16. pandas - Python Data Analysis Library — Текст : электронный // : [сайт]. — URL: https://pandas.pydata.org/ (дата обращения 10.03.2023)

17. WordNet — Текст : электронный // : [сайт]. — URL: <https://wordnet.princeton.edu/> (дата обращения 10.03.2023)

18 Морфологический анализатор pymorphy2 — Морфологический анализатор pymorphy2 — Текст : электронный // : [сайт]. — URL: https://pymorphy2.readthedocs.io/en/stable/index.html# (дата обращения 10.03.2023)

19. NLTK :: Natural Language Toolkit — Текст : электронный // : [сайт]. — URL: https://www.nltk.org/ (дата обращения 10.03.2023)

20. HTML Standard — Текст : электронный // : [сайт]. — URL: https://html.spec.whatwg.org/multipage/ (дата обращения 10.03.2023)

21. Cascading Style Sheets — Текст : электронный // : [сайт]. — URL: https://www.w3.org/Style/CSS/ (дата обращения 10.03.2023)

22. британский — Текст : электронный // : [сайт]. — URL: https://brython.info/ (дата обращения 10.03.2023)

23. Тематическое моделирование — Текст : электронный // : [сайт]. — URL: http://www.machinelearning.ru/wiki/index.php?title=Тематическое\_моделирование (дата обращения 08.03.2023)

24. Gensim — Руководство для начинающих — Текст : электронный // : [сайт]. — URL: <https://webdevblog.ru/gensim-rukovodstvo-dlya-nachinajushhih/> (дата обращения 18.01.2023)

25. RusVectōrēs: модели — Текст : электронный // : [сайт]. — URL: https://rusvectores.org/ru/ (дата обращения 08.03.2023)

26. Recognition of Sarcastic Sentences in the Task of Sentiment Analysis — Текст : электронный // : [сайт]. — URL: https://www.elibrary.ru/item.asp?id=43858432 (дата обращения 08.03.2023)