БОЛЬШИЕ ВЫЗОВЫ

ВСЕРОССИЙСКИЙ КОНКУРС
НАУЧНО-ТЕХНОЛОГИЧЕСКИХ ПРОЕКТОВ

МГК 2021

**Региональный трек**
**Всероссийского конкурса**
**научно-технологических проектов**

**«БОЛЬШИЕ ВЫЗОВЫ»**

**направление**

## Большие данные, искусственный интеллект, финансовые технологии и машинное обучение

**название работы**

# Система для автоматической верификациии русскоязычного текста

**участник(и)**


## Бабанский Виталий Витальевич


#большиевызовы
#мгк

mgk.olimpiada.ru

г. Москва
2021

# Команда проекта

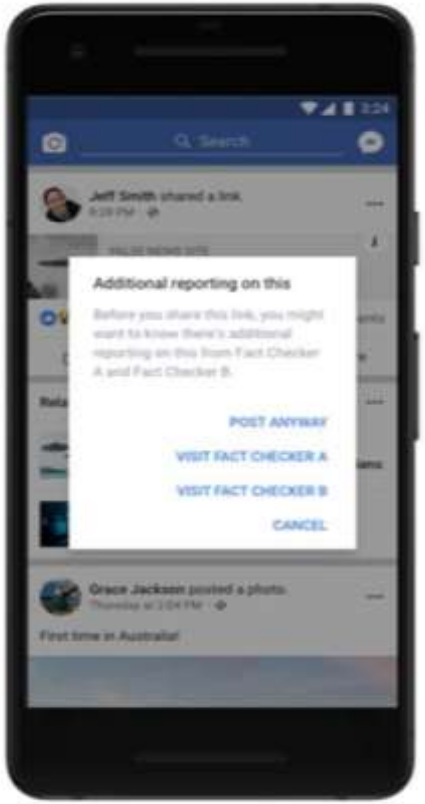
| ФИО | Должность | Функция | Задачи в проекте |
|---|---|---|---|
| Жемчужников Дмитрий Григорьевич | Преподаватель информатики | Научный руководитель | Организация работы проектной команды |
| Бабанский Виталий Витальевич | Учащийся 10 В класса | Программист, тестировщик | 1. Проектирование системы.<br>2. Сбор данных для машинного обучения.<br>3. Реализация программной части проекта.<br>4. Тестирование системы. |

# Проблема

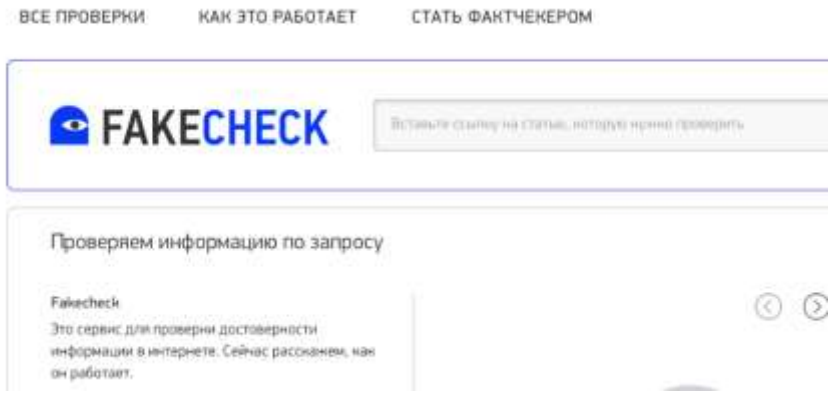
По статистике в январе 2021 года интернетом регулярно пользовалось 4,66 миллиарда человек во всём мире, что составляет примерно 59% мирового населения. В интернете огромное количество недостоверных и ненаучных источников информации, которым нередко, к сожалению, слепо верят многие люди.

**Решаемая проблема:** потребность в автоматическом анализе русскоязычных электронных текстов на достоверность и отсутствие инструментов для такого анализа.

# Анализ аналогов




Facebook




Fakecheck

**Недостатки:**

- Отсутствие бесплатной гарантированной проверки
- Отсутствие полной автоматизированности
- Большое количество рекламы

# Цель проекта

Создать рабочий прототип открытой системы, позволяющей загружать и верифицировать русскоязычный текст в автоматическом режиме.

Важно учесть недостатки аналогов и обеспечить доступность и точность системы.

Потенциальные потребители: пользователи интернета, СМИ

# Задачи проекта

*Цель определила задачи проекта:*

1. Проанализировать существующие решения, определить способы реализации проекта
2. Разработать алгоритм, проверяющий орфографию текста
3. Разработать алгоритм, оценивающий истинность текста на основе машинного обучения
4. Собрать обучающие данные для машинного алгоритма
5. Разработать интерфейс сервиса
6. Протестировать проект
7. Документировать проект и определить дальнейшие перспективы развития проекта

# План проекта

| № п/п | Название этапа | Сроки реализации |
|---|---|---|
| 1 | **Подготовка, планирование** | |
| | Постановка цели, обсуждение концепции и особенностей проекта | 01.10 – 25.10 |
| 2 | **Разработка** | |
| | Реализация алгоритма проверки орфографии текста | 02.11 – 12.11 |
| | Сбор данных для машинного обучения | 12.11 – 22.11 |
| | Разработка алгоритма оценки истинности текста на основе машинного обучения | 22.11 – 02.12 |
| | Разработка интерфейса системы, визуализация проекта | 02.12 – 15.12 |
| | Проведение тестов | 15.12 – 20.12 |
| 3 | **Подведение итогов** | |
| | Доработка программного комплекса, обсуждение итогов и документирование проекта | 20.12-31.01 |

# Машинное обучение

- Сбор текстов верных и ложные статей
- Разделение данных на обучающие и контрольные
- Обработка текста
- Частотное кодирование

```
компания    закупила    гидрологическое оборудование    заявил  интерсакс
0.35432190412191394 0   0   0.8692317197309762  0   -1.5
```

Пример кодирования текста

```
matrix_vec, target = shuffle(matrix_vec, target)
border = int(train_validate_div * len(target))
X, Y = shuffle(matrix_vec[:border, :], target[:border])
X_control, Y_control = shuffle(matrix_vec[border:, :], target[border:])

print('training')
self.model = GaussianNB()
self.model.fit(X, Y)
predicted = self.model.predict(X_control)
```

Тренировка классификатора

# Комплексная проверка текста

- Итог проверки орфографии – отношение количества верно написанных слов к их общему количеству

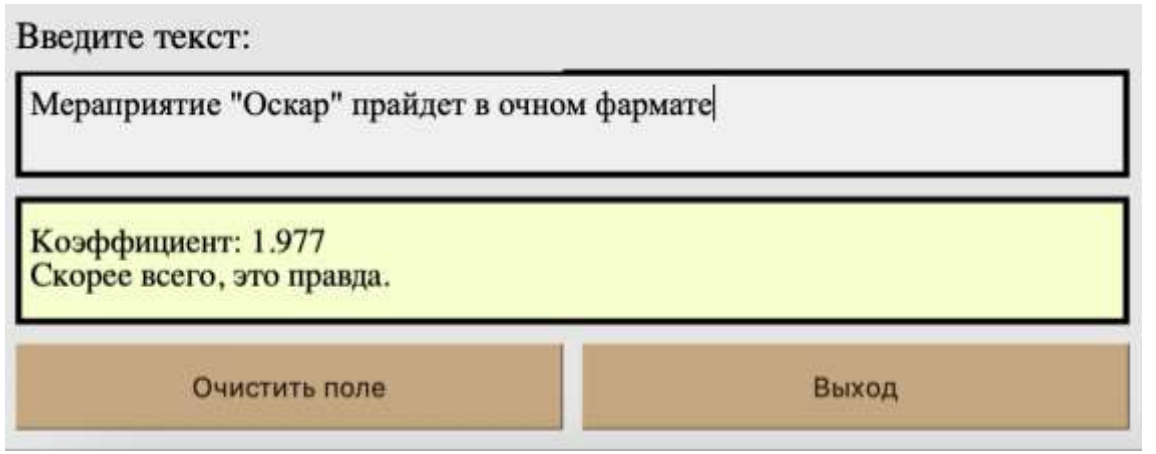
```
t0 = time.time()
text = 'Добрый вечир, дамы и господа, рад приветствовать' \
       ' ваз на нашем благотварительном вечери'
text_tokens = proccess_text(text, normal_form=True)
print('grammar rate is  ', get_grammar_score(text_tokens))
# grammar rate is   0.8
```

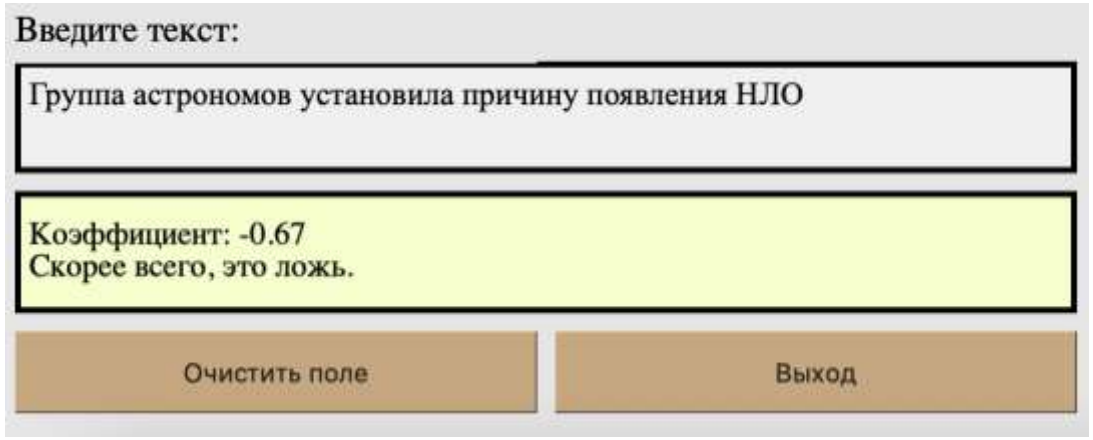
Тест алгоритма на тексте с ошибками

```
def complex_check(self, text):
    verdict, st_s, mn_s = self.check_text(text)
    predicted = 0.5 if verdict == 't' else -0.5
    tokenized_text = proccess_text(text, normal_form=True)
    grammar_rate = get_grammar_score(tokenized_text)
    check_score = predicted * 3 + (grammar_rate - 0.85) * 2 + st_s + mn_s
    return check_score, verdict
```

Вычисление итогового коэффициента

- Создана формула для получения вердикта комплексной проверки
- Реализован универсальный ООП интерфейс



- Наглядное изменение результата в реальном времени
- 4 степени верности информации

# Ключевые результаты

- Создан удобный интерфейс
- Достигнута точность в верификации русскоязычного текста
- Проект успешно протестирован на текстах из интернета
- Создан и задокументирован API

# Таблица сравнения с аналогами

| Проект / Характеристики | Fakecheck | Snopes | Facebook | Данный проект |
|---|---|---|---|---|
| Автоматизированность | 0 | 0 | 1 | 1 |
| Доступность | 0 | 1 | 1 | 1 |
| Работа с русскоязычными текстами | 1 | 0 | 1 | 1 |
| Отсутствие рекламы | 1 | 0 | 0 | 1 |
| Наличие гарантированной проверки текста | 1 | 0 | 0 | 1 |
| Универсальность | 1 | 1 | 0 | 1 |
| Проверка в реальном времени | 0 | 0 | 0 | 1 |
| Итоги | 4 | 2 | 3 | 6 |

# Дальнейшие шаги по работе над проектом

- Расширение функций, например, добавление дополнительной ручной проверки текста
- Повышение точности системы за счёт увеличения количества обучающих данных
- Получение более детального анализа в результате.

Итоговую версию прототипа можно скачать по ссылке: https://drive.google.com/drive/folders/1brOAFxDUTvryY8fCjNFTuEUW8Gp33I7R?usp=sharing

# Использованные ресурсы


1. https://www.web-canape.ru/business/internet-2020-globalnaya-statistika-i-trendy/
2. https://stackoverflow.com/
3. https://habr.com/ru/post/332078/
4. https://datastart.ru/blog/read/chto-takoe-scikit-learn-gayd-po-populyarnoy-biblioteke-python-dlya-nachinayuschih
5. https://habr.com/ru/post/332078/
6. https://pypi.org/project/pyenchant/
7. https://www.machinelearningmastery.ru/text-classification-in-python- dd95d264c802/

# Спасибо за внимание!

ПРОЕКТ
“Система для автоматической верификации русскоязычного текста”